การพัฒนาโมเดลสำหรับวิเคราะห์และเปรียบเทียบคำอธิบายรายวิชาจากมหาวิทยาลัยต่างๆ : กรณีศึกษารายวิชาทางด้านวิทยาการคอมพิวเตอร์

Development of a model for analyzing and comparing course descriptions from universities: a case study of courses in computer science

พีระพล กำลังพืช

มหาวิทยาลัยบูรพา

2564

การพัฒนาโมเดลสำหรับวิเคราะห์และเปรียบเทียบคำอธิบายรายวิชาจากมหาวิทยาลัยต่างๆ : กรณีศึกษารายวิชาทางด้านวิทยาการคอมพิวเตอร์

พีระพล กำลังพืช

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาการสารสนเทศ
คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา
2564

Development of a model for analyzing and comparing course descriptions from universities: a case study of courses in computer science

PEERAPON KAMLANGPUECH

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE MASTER DEGREE OF SCIENCE
IN INFORMATICS
FACULTY OF INFORMATICS
BURAPHA UNIVERSITY
2021

คณะกรรมการควบคุมวิทยานิพนธ์และคณะกรรมการสอบวิทยานิพนธ์ได้พิจารณาวิทยานิพนธ์ของ พีระพล กำลังพืช ฉบับนี้แล้ว เห็นสมควรรับเป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการสารสนเทศ ของมหาวิทยาลัยบูรพาได้

คณะกรรมการควบคุมวิทยานิพนธ์

..... อาจารย์ที่ปรึกษาหลัก

(ผู้ช่วยศาสตราจารย์ ดร. โกเมศ อัมพวัน)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธาน

(ผู้ช่วยศาสตราจารย์ ดร. โกเมศ อัมพวัน)

..... กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร. อุรีรัฐ สุขสวัสดิ์ชน)

..... กรรมการภายนอกมหาวิทยาลัย

(รองศาสตราจารย์ ดร.อนุชิต จิตพัฒนกุล)

คณะวิทยาการสารสนเทศอนุมัติให้รับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการสารสนเทศ ของมหาวิทยาลัยบูรพา

..... คณบดีคณะวิทยาการสารสนเทศ

(ผู้ช่วยศาสตราจารย์ ดร. กฤษณะ ชินสาร)

วันที่.....เดือน.....พ.ศ.....

60910058: สาขาวิชา: วิทยาการสารสนเทศ; วท.ม. (วิทยาการสารสนเทศ)
คำสำคัญ: การวิเคราะห์คำอธิบายรายวิชา, คำอธิบายรายวิชา, คำอธิบายรายวิชาในหลักสูตรวิทยาการคอมพิวเตอร์

พีระพล กำลังพืช : การพัฒนาโมเดลสำหรับวิเคราะห์และเปรียบเทียบคำอธิบายรายวิชาจากมหาวิทยาลัยต่างๆ : กรณีศึกษารายวิชาทางด้านวิทยาการคอมพิวเตอร์. (Development of a model for analyzing and comparing course descriptions from universities: a case study of courses in computer science) คณะกรรมการควบคุมวิทยานิพนธ์: โกเมศ อัมพวัน ปี พ.ศ. 2564.

ปัจจุบันวิธีการวิเคราะห์คำอธิบายรายวิชาได้ถูกพัฒนาอย่างต่อเนื่อง อาทิเช่น การถึงวิเคราะห์วัตถุประสงค์ที่ได้รับจากการเรียนรู้ในชั้นเรียนร่วมกับข้อมูลเชิงลึกข้อผู้เรียนแต่ละคน เพื่อทำการวัดประสิทธิภาพองค์ความรู้ของผู้เรียนแต่ละคนที่ได้รับจากการเรียนในชั้นเรียน, เพื่อการแนะนำคอร์สเรียนที่เหมาะสมกับผู้เรียนแต่ละคน เป็นต้น แต่ก็ยังไม่มีวิธีใดเลยที่จะทำการวิเคราะห์และเปรียบเทียบระหว่างคำอธิบายรายวิชาในรายวิชาหนึ่ง ๆ ดังนั้นผู้วิจัยจึงได้ทำการพัฒนาระบบวิเคราะห์คำอธิบายรายวิชาในหลักสูตรวิทยาการคอมพิวเตอร์ (เรียกโดยย่อว่า “ระบบซีเอสซีดีเอ”) ซึ่งเป็นระบบสำหรับการวิเคราะห์และเปรียบเทียบระหว่างคำอธิบายรายวิชา เพื่อแสดงให้เห็นถึงส่วนของเนื้อหาในคำอธิบายรายวิชาที่เหมือนและแตกต่างกัน นอกจากนี้ผู้วิจัยยังได้ทำการพัฒนาระบบวิเคราะห์คำอธิบายรายวิชาในหลักสูตรวิทยาการคอมพิวเตอร์ที่มีประสิทธิภาพ (เรียกโดยย่อว่า “ระบบอีซีเอสซีดีเอ”) ซึ่งเป็นระบบที่ถูกพัฒนาต่อมาจากระบบอีซีเอสซีดีเอ เพื่อให้ได้มาซึ่งผลลัพธ์ที่มีความครอบคลุมและมีประสิทธิภาพมากยิ่งขึ้น ในการประเมินประสิทธิภาพของระบบจะเป็นการดำเนินการทดสอบกับคำอธิบายรายวิชาในหลักสูตรวิทยาการคอมพิวเตอร์จากทั้ง 10 มหาวิทยาลัย ซึ่งจะแบ่งการประเมินออกเป็น 2 ส่วน คือ 1) การประเมินประสิทธิภาพของการเปรียบเทียบจากคำสำคัญที่สกัดได้ และ 2) การประเมินประสิทธิภาพของวิธีการเปรียบเทียบข้อความ จากการประเมินประสิทธิภาพแสดงให้เห็นว่าระบบอีซีเอสซีดีเอมีประสิทธิภาพที่ดีกว่าระบบและขั้นตอนวิธีอื่น ๆ ที่นำมาเปรียบเทียบ

60910058: MAJOR: INFORMATICS; M.Sc. (INFORMATICS)
KEYWORDS: Content analysis Course description Computer Science course

PEERAPON KAMLANGPUECH : DEVELOPMENT OF A MODEL FOR ANALYZING AND COMPARING COURSE DESCRIPTIONS FROM UNIVERSITIES: A CASE STUDY OF COURSES IN COMPUTER SCIENCE. ADVISORY COMMITTEE: KOMATE AMPHAWAN, Ph.D. 2021.

Now a day, the method of analyzing the content of the course descriptions has been developed and has become more diverse. For example, analyze the objectives gained from classroom learning together with insights into individual learners. To measure the effectiveness of each student's knowledge gained through classroom learning, in order to recommend courses that are suitable for each student, and so on. From the methods of analyzing the course descriptions, there is no method to analyze and compare between two or more course descriptions. Therefore, the researcher has developed a new system called “Computer Science Course Description Analysis system (CSCDA system)”. It is a system for analysis and comparison between course descriptions. To represent the similar content and different content in the course description. Furthermore, a new improvement of the CSCDA system is called “An efficient system for analyzing contents of Computer Science Courses (eCSCDA system)”. To achieve more comprehensive and effective results. In experiments were conducted on CS course contents gathered from ten Thai Universities. The experiment consists of two aspects i.e. 1) Evaluation of keyword extraction and 2) Evaluation of similarity matching. From the results, it can be seen and concluded that the eCSCDA system is more efficient than other systems and algorithms.

กิตติกรรมประกาศ

งานวิจัยฉบับนี้สำเร็จลุล่วงไปด้วยดีเนื่องจากผู้วิจัยได้รับความช่วยเหลือและดูแลเอาใจใส่เป็นอย่างดีจาก หลาย ๆ ฝ่าย โดยเฉพาะท่านอาจารย์ที่ปรึกษา ผู้ช่วยศาสตราจารย์ ดร. โกเมศ อัมพวัน ในการให้คำแนะนำ ตรวจแก้ไข ให้ข้อเสนอแนะ ติดตามความก้าวหน้าในการดำเนินงานวิจัย ผู้วิจัยรู้สึกซาบซึ้งในความกรุณาของอาจารย์ท่านนี้เป็นอย่างยิ่งและขอขอบพระคุณเป็นอย่างสูงไว้ ณ โอกาสนี้

ความสำเร็จในการทางานวิจัยฉบับนี้ผู้วิจัยขอน้อมลำลึกถึงพระคุณบิดามารดาที่ได้ส่งเสริมสนับสนุน และได้รับกำลังใจเป็นอย่างดีจากครอบครัว ตลอดจนเพื่อน ๆ พี่ ๆ และ น้อง ๆ ทุกคนในห้องปฏิบัติการวิจัย Computational Innovation Laboratory (CIL) มาโดยตลอด และขอรำลึกถึงครูอาจารย์ทุกท่านที่ประสิทธิ์ประสาทวิชาความรู้ให้แก่ผู้วิจัยตั้งแต่อดีตถึงปัจจุบัน

เนื่องจากงานวิจัยฉบับนี้ได้รับการสนับสนุนทุนวิจัยจากคณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา จึงขอขอบคุณมา ณ ทีนี้

สุดท้ายนี้ ขอบคุณตนเองที่มีจิตใจเข้มแข็ง พยายาม อดทน และไม่ย่อท้อต่ออุปสรรคต่าง ๆ จนสามารถดำเนินงานวิจัยฉบับนี้ได้สำเร็จลุล่วง

พีระพล กำลังพืช

สารบัญ

หน้า

สารบัญตาราง

หน้า

สารบัญภาพ

หน้า

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของงานวิจัย

ในปัจจุบันเทคโนโลยีทางด้านวิทยาการคอมพิวเตอร์ได้มีการเติบโตและมีความหลากหลายมากยิ่งขึ้น ศาสตร์ทางด้านวิทยาการคอมพิวเตอร์เองก็มีการพัฒนาองค์ความรู้และ/หรือเกิดองค์ความรู้ใหม่ ๆ อย่างต่อเนื่อง อาทิเช่น เทคโนโลยีวิทยาศาสตร์ข้อมูล (Data science technology) เทคโนโลยีข้อมูลขนาดใหญ่ (Big data technology) เทคโนโลยีอินเทอร์เน็ตของสรรพสิ่ง (IoT technology) และอื่น ๆ จากความแพร่หลายของเทคโนโลยีข้างต้น ส่งผลให้มหาวิทยาลัยต่าง ๆ ที่เปิดสอนหลักสูตรวิทยาการคอมพิวเตอร์จำเป็นต้องปรับปรุง/เปลี่ยนแปลงโครงสร้างหลักสูตร รวมถึงกำหนดให้มีรายวิชาดังกล่าวข้างต้น พร้อมกับมีการปรับปรุงเนื้อหาของรายวิชาต่าง ๆ ให้มีความสอดคล้อง/รองรับกับเทคโนโลยีใหม่ ๆ และเพื่อให้เนื้อหามีความเหมาะสมกับแนวโน้มของเทคโนโลยีในปัจจุบันมากยิ่งขึ้น โดยคำอธิบายรายวิชาของรายวิชาหนึ่ง ๆ จะบ่งบอกถึงองค์ความรู้ที่จะถูกสอน (ถ่ายทอด) ในวิชานั้น ๆ จากการทราบถึงคำอธิบายรายวิชาจะมีประโยชน์ในหลายแง่มุม อาทิเช่น 1) ช่วยให้นิสิต/นักศึกษาสามารถทำการตัดสินใจ ในการเลือกลงทะเบียนเรียนในรายวิชานั้น ๆ โดยพิจารณาจากเนื้อหาในคำอธิบายรายวิชาเป็นสำคัญ 2) ช่วยให้นิสิต/นักศึกษาสามารถทำความเข้าใจถึงเนื้อหาของวิชาที่ลงทะเบียนเรียนได้ล่วงหน้า และ อื่น ๆ

โดยปกติของการเปิดสอนของหลักสูตรหนึ่ง ๆ จะมีการปรับปรุงโครงสร้างหลักสูตรและเนื้อหา (คำอธิบายรายวิชา) อย่างน้อยหนึ่งครั้งในทุก ๆ 5 ปี โดยในการปรับปรุงเนื้อหาของรายวิชาหนึ่ง ๆ จะขึ้นอยู่กับผู้สอนและ/หรือคณะกรรมการประจำหลักสูตรเป็นผู้พิจารณา ซึ่งจากการดำเนินการดังกล่าวอาจทำให้รายวิชาหนึ่ง ๆ ที่เปิดสอนในมหาวิทยาลัยต่าง ๆ มีเนื้อหาของคำอธิบายรายวิชาที่เหมือนและแตกต่างกัน ดังตัวอย่างของภาพที่ 1 ซึ่งจะแสดงให้เห็นถึงส่วนที่เหมือนกันทั้ง 3 คำอธิบายรายวิชา (ในกรอบสีเขียวเข้ม), ส่วนที่เหมือนกันระหว่างเนื้อหาของคำอธิบายรายวิชาของมหาวิทยาลัยบูรพาและจุฬาลงกรณ์มหาวิทยาลัย (ในกรอบสีฟ้า), ส่วนที่เหมือนกันระหว่างเนื้อหาของคำอธิบายรายวิชาของมหาวิทยาลัยบูรพาและมหาวิทยาลัยเชียงใหม่ (ในกรอบสีเขียวอ่อน), ส่วนที่เหมือนกันระหว่างเนื้อหาของคำอธิบายรายวิชาของจุฬาลงกรณ์มหาวิทยาลัยและมหาวิทยาลัยเชียงใหม่ (ในกรอบสีนำเงิน) และส่วนที่แตกต่างของเนื้อหาทั้ง 3 คำอธิบายรายวิชา (สี

แดง) จากคำอธิบายรายวิชาของรายวิชา “Data Mining” จากมหาวิทยาลัยบูรพา, จุฬาลงกรณ์มหาวิทยาลัย และ มหาวิทยาลัยเชียงใหม่

คำอธิบายรายวิชาของวิชา “Data Mining” ของมหาวิทยาลัยบูรพา
Fundamental concepts of data mining, types of data for data mining, famous techniques for data mining, pattern mining and association-rule mining, classification, clustering, outlier analysis, anomaly detection, data mining tools

คำอธิบายรายวิชาของวิชา “Data Mining” ของจุฬาลงกรณ์มหาวิทยาลัย
Data mining, data preprocessing, data warehouse and OLAP technology, mining association rules and frequent patterns, classification, cluster analysis, applications and trends in data mining

คำอธิบายรายวิชาของวิชา “Data Mining” ของมหาวิทยาลัยเชียงใหม่
Basic concept of data mining, data preprocessing, dimensional data reduction, association rules mining, data clustering, data classification and data prediction

ภาพที่ 1 ตัวอย่างการเปรียบเทียบความเหมือนและความแตกต่างของคำอธิบายรายวิชาในรายวิชา “Data Mining” ของมหาวิทยาลัยบูรพา, จุฬาลงกรณ์มหาวิทยาลัย และ มหาวิทยาลัยเชียงใหม่

จากการที่เนื้อหาของคำอธิบายรายวิชาของรายวิชาหนึ่ง ๆ ของแต่ละมหาวิทยาลัยมีทั้งส่วนที่เหมือนและแตกต่างกัน ด้วยเหตุนี้ จึงนำมาซึ่งความแตกต่างกันขององค์ความรู้ที่จะได้รับจากการเรียนวิชานั้น ๆ ซึ่งจะส่งผลต่อความเหลื่อมล้ำทางด้านคุณภาพของบัณฑิตที่จบจากมหาวิทยาลัยที่แตกต่างกัน และอาจส่งผลต่อการมีองค์ความรู้ในบางเรื่องที่ไม่เพียงพอต่อการทำงานในภาคอุตสาหกรรม

จากปัญหาข้างต้น ผู้วิจัยจึงมีแนวคิดที่จะพัฒนาระบบสำหรับการวิเคราะห์และเปรียบเทียบคำอธิบายรายวิชาของรายวิชาหนึ่ง ๆ ในหลักสูตรวิทยาการคอมพิวเตอร์ที่เปิดสอนในมหาวิทยาลัยต่าง ๆ เพื่อที่จะแสดงให้เห็นถึงความเหมือนและความแตกต่างกันของคำอธิบายรายวิชาของรายวิชาหนึ่ง ๆ ในแต่ละมหาวิทยาลัย อันนำมาซึ่งการเป็นส่วนช่วยในการตัดสินใจด้านการพัฒนาและปรับปรุงเนื้อหา (คำอธิบายรายวิชา) ของรายวิชาหนึ่ง ๆ ให้มาตรฐานขององค์ความรู้ที่จะถูกสอน

(ถ่ายทอด) ให้กับนิสิต/นักศึกษามีความครบถ้วนสมบูรณ์และมีความสอดคล้อง/รองรับกับเทคโนโลยีในปัจจุบันมากยิ่งขึ้น อีกทั้งยังช่วยลดความเหลื่อมล้ำทางด้านคุณภาพของบัณฑิต (ในสาขาวิทยาการคอมพิวเตอร์) ที่จบจากมหาวิทยาลัยต่างที่แตกต่างกัน และส่งเสริมให้บัณฑิตมีคุณภาพเพียงพอต่อการทำงานในภาคอุตสาหกรรมสืบไป ที่ซึ่งจากการพัฒนาระบบข้างต้นทำให้สามารถ 1) เป็นส่วนช่วยในการตัดสินใจให้กับผู้สอนและ/หรือคณะกรรมการประจำหลักสูตรในการปรับปรุงเนื้อหาในรายวิชาหนึ่ง ๆ ให้เป็นมาตรฐาน จากการพิจารณาผลลัพธ์ที่ได้จากการวิเคราะห์และเปรียบเทียบคำอธิบายรายวิชา 2) นำแนวคิดและระบบสำหรับการวิเคราะห์และเปรียบเทียบคำอธิบายรายวิชาของรายวิชาหนึ่ง ๆ ไปพัฒนาต่อยอดกับคอร์สอบรมต่าง ๆ ที่อาจจะสามารถช่วยให้ได้คอร์สอบรมที่มีมาตรฐานและ/หรือมีเนื้อหาใหม่ ๆ น่าสนใจมากขึ้น 3) เป็นส่วนช่วยในการตัดสินใจในการเลือกมหาวิทยาลัยสำหรับบุคคลที่สนใจเข้าศึกษาต่อในระดับอุดมศึกษาด้วยผลลัพธ์จากการเปรียบเทียบหลักสูตรของแต่ละมหาวิทยาลัย

1.2 วัตถุประสงค์ของงานวิจัย

1. เพื่อพัฒนาระบบสำหรับการวิเคราะห์และเปรียบเทียบคำอธิบายรายวิชาในรายวิชาหนึ่ง ๆ ตั้งแต่ 2 คำอธิบายรายวิชาขึ้นไป ซึ่งจะแสดงให้เห็นถึงความเหมือนและความแตกต่างกันของเนื้อหาในคำอธิบายรายวิชาตั้งต้น โดยคำอธิบายรายวิชาที่นำมาเปรียบเทียบกันจะต้องเป็นคำอธิบายรายวิชาที่เป็นรายวิชาเดียวกันเท่านั้น
2. เพื่อสร้างต้นแบบงานวิจัยทางด้านการการวิเคราะห์และเปรียบเทียบระหว่างคำอธิบายรายวิชาในรายวิชาหนึ่ง ๆ ที่จะสามารถเปิดโอกาสให้ผู้ที่สนใจสามารถนำความคิดที่นำเสนอไปศึกษาเพิ่มเติม พัฒนาเป็นผลผลิตเพื่อใช้ในองค์กรของตนเองและเพื่อประยุกต์ใช้ในงานวิจัยของตนเองต่อไป

1.3 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้ขั้นตอนวิธีสำหรับการวิเคราะห์และเปรียบเทียบ เพื่อหาความเหมือนและความแตกต่างระหว่างคำอธิบายรายวิชาของรายวิชาหนึ่ง ๆ
2. ได้ระบบสำหรับการวิเคราะห์และเปรียบเทียบระหว่างคำอธิบายรายวิชาในรายวิชาหนึ่ง ๆ แบบอัตโนมัติ เพื่อแสดงให้เห็นถึงส่วนของเนื้อหาในคำอธิบายรายวิชาตั้งต้นที่เหมือนและแตกต่างกันกับคำอธิบายรายวิชาเปรียบเทียบ และแสดงเป็นข้อมูลเชิงสรุปที่จะบ่งบอกถึงอัตราร้อยละของความเหมือนและความแตกต่างจากการเปรียบเทียบ

3. ได้ผลงานวิจัยที่จะสามารถตีพิมพ์ในงานประชุมวิชาการหรือวารสารวิชาการ ดังนี้
    I. 2020, 7th International Conference on Advanced Informatics: Concept Theory and Applications
    II. 2021, 8th International Conference on Advanced Informatics: Concept Theory and Applications
4. สามารถนำแนวคิด, ขั้นตอนวิธี และระบบวิเคราะห์คำอธิบายรายวิชาไปพัฒนาต่อยอดเพื่อดำเนินงานวิจัยขั้นสูงในการวิเคราะห์และเปรียบเทียบคำอธิบายรายวิชาในหลักสูตรอื่น ๆ ได้

1.4 ขอบเขตของงานวิจัย

1. ในงานวิจัยนี้จะทำการสร้างระบบสำหรับการวิเคราะห์และเปรียบเทียบเนื้อหาของคำอธิบายรายวิชาในรายวิชาหนึ่ง ๆ ภายใต้หลักสูตรวิทยาการคอมพิวเตอร์
2. ในงานวิจัยนี้จะทำการรวบรวมเฉพาะคำอธิบายรายวิชาที่เป็นภาษาอังกฤษเท่านั้น
3. คำอธิบายรายวิชาที่นำมาใช้ในการทดสอบประสิทธิภาพการทำงานของระบบ รวบรวมได้จากมหาวิทยาลัยที่มีการเปิดสอนหลักสูตรวิทยาการคอมพิวเตอร์ และสามารถสืบค้นคำอธิบายรายวิชาได้จากเว็บไซต์ของของภาควิชานั้น ๆ โดยข้อมูลคำอธิบายรายวิชาที่นำมาใช้ในงานวิจัยนี้ เป็นข้อมูลคำอธิบายรายวิชาของมหาวิทยาลัย 10 อันดับแรกที่ถูกจัดอันดับจาก QS University Ranking of Asian[1]

[1] https://www.topuniversities.com/university-rankings/asian-university-rankings/2021

1.5 แผนการดำเนินงานวิจัย

งานวิจัยนี้มีขั้นตอนการดำเนินงานภายใต้ระยะเวลาที่ระบุไว้ดังตารางที่ 1

ตารางที่ 1 แผนการดำเนินงานวิจัย

ปี	การดำเนินงาน	เดือน											
		1	2	3	4	5	6	7	8	9	10	11	12
2560	ศึกษาและค้นคว้างานวิจัยที่เกี่ยวข้องกับการเปรียบเทียบข้อความ								↔	↔	↔	↔	↔
2561	กำหนดขอบเขตของระบบ	↔	↔										
	ออกแบบขั้นตอนวิธีและการดำเนินงานของระบบ			↔	↔								
	รวบรวมข้อมูลคำอธิบายรายวิชาจากมหาวิทยาลัยที่ถูกพิจารณาและคำศัพท์เฉพาะในศาสตร์ทางด้านวิทยาการคอมพิวเตอร์					↔							
	เขียนโปรแกรมสำหรับการวิเคราะห์และเปรียบเทียบระหว่างคำอธิบายรายวิชาในรายวิชาหนึ่ง ๆ						↔	↔	↔				
	ทำการทดสอบและตรวจสอบผลลัพธ์ที่ได้									↔			
	แก้ไขและปรับปรุงโปรแกรมสำหรับการวิเคราะห์และเปรียบเทียบระหว่างคำอธิบายรายวิชาในรายวิชาหนึ่ง ๆ										↔	↔	↔
2562	ทำการทดสอบและตรวจสอบผลลัพธ์ที่ได้	↔											

ตารางที่ 1 แผนการดำเนินงานวิจัย (ต่อ)

ปี	การดำเนินงาน	เดือน											
		1	2	3	4	5	6	7	8	9	10	11	12
2562	จัดทำเอกสารและสอบโครงร่างวิทยานิพนธ์		↔	↔	↔								
	รวบรวมข้อมูลคำอธิบายรายวิชาจากมหาวิทยาลัยที่ถูกพิจารณาและคำศัพท์เฉพาะในศาสตร์ทางด้านวิทยาการคอมพิวเตอร์เพิ่มเติม					↔							
	แก้ไขและปรับปรุงโปรแกรมสำหรับการวิเคราะห์และเปรียบเทียบระหว่างคำอธิบายรายวิชาในรายวิชาหนึ่ง ๆ						→	→	→	→	→	→	→
2563	สรุปการประมวลผลของระบบและจัดทำเอกสารงานวิจัยที่ 1	↔	↔	↔	↔	↔	↔						
	เผยแพร่งานวิจัยที่ 1							↔	↔				
	ศึกษาและค้นคว้างานวิจัยที่เกี่ยวข้องกับการเปรียบเทียบข้อความเพิ่มเติม								→	→	→	→	→
2564	รวบรวมข้อมูลและสร้างคลังพ้องความหมายของคำศัพท์เฉพาะและคลังพ้องความหมายของคำศัพท์ทั่วไป	↔	↔										
	พัฒนาโปรแกรมสำหรับการวิเคราะห์และเปรียบเทียบระหว่างคำอธิบายรายวิชาในรายวิชาหนึ่ง ๆ		↔	↔	↔								

ตารางที่ 1 แผนการดำเนินงานวิจัย (ต่อ)

ปี	การดำเนินงาน	เดือน											
		1	2	3	4	5	6	7	8	9	10	11	12
2564	สรุปการประมวลผลของระบบและจัดทำเอกสารงานวิจัยที่ 2					←	→						
	เผยแพร่งานวิจัยที่ 2							←	—	→			
	จัดทำเอกสารและสอบวิทยานิพนธ์									←	—	→	

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

บทนี้จะเป็นการอธิบายถึงทฤษฎีและงานวิจัยที่เกี่ยวข้องกับการวิเคราะห์และเปรียบเทียบระหว่างคำอธิบายรายวิชาของรายวิชาหนึ่ง ๆ ซึ่งผู้วิจัยได้ประยุกต์ใช้วิธีการต่าง ๆ เพื่อทำให้การประมวลผลสามารถทำงานได้อย่างมีประสิทธิภาพ โดยจะทำการอธิบายถึงวิธีการที่ได้ประยุกต์ใช้ดังต่อไปนี้

2.1 การประมวลผลข้อความเบื้องต้น (Text preprocessing)

การประมวลผลข้อความเบื้องต้นเป็นการเปลี่ยนแปลงข้อความให้อยู่ในรูปแบบที่สามารถนำไปวิเคราะห์และเพิ่มประสิทธิภาพในการดำเนินงานเชิงลึกได้ในระดับต่อไป โดยการประมวลผลข้อความเบื้องต้นสามารถทำได้หลากหลายวิธีการ อาทิเช่น การแก้ไขคำผิด, การกำจัดคำหยุด, การแปลงรูปคำให้อยู่ในรากศัพท์, การแบ่งคำ, การแบ่งประโยค เป็นต้น ซึ่งในงานวิจัยนี้ผู้วิจัยได้ประยุกต์ใช้วิธีการที่เกี่ยวข้องกับการประมวลผลข้อความเบื้องต้นดังนี้

2.1.1 การแปลงตัวอักษรพิมพ์ใหญ่ให้เป็นตัวอักษรพิมพ์เล็ก (Lowercase conversion)

คือ การพิจารณาถึงตัวอักษรตัวพิมพ์ใหญ่ในภาษาอังกฤษที่ปรากฏอยู่ในบทความหนึ่ง ๆ ซึ่งเมื่อตรวจพบตัวอักษรตัวพิมพ์ใหญ่ จะดำเนินการแทนที่ตัวอักษรตัวนั้นด้วยตัวอักษรตัวเดียวกันที่เป็นตัวพิมพ์เล็ก

2.1.2 การแบ่งประโยค (Sentence tokenization)

คือ การแบ่งประโยคแต่ละประโยคในบทความหนึ่ง ๆ ออกจากกัน โดยการพิจารณาถึงสัญลักษณ์ที่ปรากฏอยู่ท้ายประโยคในแต่ละประโยค เช่น เครื่องหมายมหัพภาค (‘.’), เครื่องหมายจุลภาคหรือเครื่องหมายลูกน้ำ (‘,’) หรือ เครื่องหมายอัฒภาค (‘;’) เป็นต้น จากนั้นทำการแบ่งประโยคแต่ละประโยคออกจากกัน

2.1.3 การแบ่งคำ (Word tokenization)

คือ การแบ่งคำแต่ละคำในประโยคหนึ่ง ๆ ออกจากกัน ซึ่งในการแบ่งคำจะทำการพิจารณาถึงช่องว่างระหว่างคำ (White space) จากนั้นดำเนินการแบ่งคำแต่ละคำออกจากกัน

2.1.4 การแก้ไขคำผิด (Word error correction)

คือ วิธีการตรวจสอบและแก้ไขการสะกดคำที่อยู่ในข้อความ ซึ่งจะทำการตรวจสอบการสะกดคำของคำหนึ่ง ๆ ในข้อความว่าเกิดการสะกดผิดขึ้นหรือไม่ โดยวิธีของการตรวจสอบความถูกต้องของการสะกดคำนั้นจะทำการตรวจสอบเชิงการเปรียบเทียบความเหมือนกัน ของตัวอักษรในคำที่พบว่ามีการสะกดที่ผิดกับคำศัพท์ในคลังคำศัพท์/พจนานุกรม โดยจะทำการเปรียบเทียบคำที่มีการสะกดผิดกับคำศัพท์ทุก ๆ คำ ซึ่งในการตรวจสอบความถูกต้องของการสะกดคำมีวิธีการที่ได้รับความนิยมในการตรวจสอบ คือ วิธีการตรวจสอบความถูกต้องของการสะกดคำแบบ N-gram อาทิเช่น Trigram เป็นต้น และทำการคำนวณหาความน่าจะเป็น (probability) จากการเปรียบเทียบกันระหว่างคำที่มีการสะกดผิดกับคำศัพท์ที่มีอยู่ จากนั้นทำการแทนที่คำที่มีการสะกดผิดด้วยคำศัพท์ที่ถูกต้องจากการพิจารณาคำศัพท์ที่มีค่าความน่าจะเป็นสูงที่สุด ซึ่งในกรณีที่พบว่าเป็นคำศัพท์เฉพาะในศาสตร์ด้านใดด้านหนึ่งหรือเป็นคำที่ไม่ได้อยู่ในคลังคำศัพท์/พจนานุกรม จะไม่สามารถดำเนินการตรวจสอบคำคำนั้นได้ว่ามีการสะกดคำที่ถูกต้องหรือไม่

2.1.5 การกำจัดคำหยุด (Stopword removal)

คือ วิธีการกำจัดคำที่อยู่ในประโยคหนึ่ง ๆ โดยที่คำนั้นจะต้องบ่งบอกถึงความหมายในตัวของมันเอง หรืออาจเป็นคำที่เมื่อกำจัดออกไปแล้วประโยคนั้น ๆ ยังคงไว้ซึ่งความหมายเดิมอยู่ อาทิเช่น คำว่า “the”, “a”, “with”, “be”, “in”, “for” เป็นต้น โดยในการกำจัดคำหยุดจะทำการตรวจสอบถึงคำหยุดที่อยู่ในประโยคนั้น ๆ จากการใช้คลังคำศัพท์ของคำหยุด ซึ่งเมื่อตรวจสอบพบคำหยุดที่อยู่ในประโยคนั้น ๆ จะดำเนินการลบคำที่เป็นคำหยุดออกจากประโยคไป

2.1.6 การแปลงรูปคำให้อยู่ในรากศัพท์ (Word stemming and lemmatization)

วิธีการแปลงรูปคำให้อยู่ในรากศัพท์เป็นหนึ่งในวิธีการของการประมวลผลภาษาธรรมชาติ (natural language processing) ที่ซึ่งจะทำการลดรูปของคำศัพท์คำหนึ่ง ๆ ให้อยู่ในรูปของรากศัพท์ อาทิเช่น คำว่า “stems”, “stemmer”, “stemming”, “stemmed” เมื่อทำดำเนินการลดรูปของคำให้อยู่ในรากศัพท์จะได้คำว่า “stem” ซึ่งเป็นรากศัพท์ของทั้ง 4 คำข้างต้น โดยวิธีในการดำเนินการลดรูปของคำให้อยู่ในรากศัพท์จะทำการพิจารณาถึงกลุ่มของตัวอักษรที่อยู่ในส่วนหน้าหรือท้ายของคำ ๆ นั้น จากกฎในการระบุถึงกลุ่มของตัวอักษรในส่วนท้ายที่สร้างขึ้นมาไว้ใช้ในการดำเนินการลดรูปของคำ เช่น กฎการพิจารณากลุ่มของตัวอักษร “*ing” เมื่อพบจะทำการเปลี่ยนกลุ่มของตัวอักษรที่พิจารณาเป็น “-” เช่น คำว่า “motoring” จะถูกลดรูปให้เป็นรากศัพท์คือว่าคำว่า

“motor”, กฎการพิจารณากลุ่มของตัวอักษร “*ational” เมื่อพบจะทำการเปลี่ยนกลุ่มของตัวอักษรที่พิจารณาเป็น “ate” เช่น คำว่า “relational” จะถูกลดรูปให้เป็นรากศัพท์คือว่าคำว่า “relate” กฎในการพิจารณาถึงกลุ่มของตัวอักษรหนึ่ง ๆ จะถูกสร้างขึ้นโดยผู้เชี่ยวชาญในด้านภาษาหรือการพิจารณาถึงกฎทางภาษาของภาษานั้น ๆ โดยมีตัวอย่างของกฎที่ใช้ในการลดรูปของคำให้อยู่ในรากศัพท์ ดังตารางที่ 2

ตารางที่ 2 ตัวอย่างของการของการระบุถึงกลุ่มของตัวอักษรที่อยู่ท้ายคำและการเปลี่ยนเปลี่ยนรูปของกลุ่มตัวอักษรที่ถูกระบุ

Suffix	Change suffix into	Example word	Root form of example word
sses	ss	caresses	caress
alize	al	formalize	formal
ical	ic	electrical	electric
ator	ate	operator	operate
alism	al	feudalism	feudal
iveness	ive	decisiveness	decisive
fulness	ful	hopefulness	hopeful
ation	ate	predication	predicate
s		stems	stem
ative		formative	form

2.1.7 การระบุหน้าที่ของคำ (Part-of-speech tagging)

เป็นเทคนิคที่ใช้ในการระบุถึงหน้าที่ของคำแต่ละคำที่ปรากฏอยู่ในประโยค โดยหน้าที่ของคำที่สามารถระบุได้ในภาษาอังกฤษมีทั้งหมด 8 ชนิด ดังต่อไปนี้

1. คำนาม (Noun) คือ คำที่ใช้เรียกแทนชื่อคน, สัตว์, สิ่งของ, สถานที่
2. คำสรรพนาม (Pronoun) คือ คำที่ใช้แทนคำนาม
3. คำกริยา (Verb) คือ คำที่แสดงอาการหรือการกระทำในประโยค
4. คำคุณศัพท์ (Adjective) คือ คำที่ทำหน้าที่ขยายคำนาม โดยตำแหน่งจะอยู่หน้าคำนามเสมอ
5. คำกริยาวิเศษณ์ (Adverb) คือ คำที่มีหน้าที่ขยายกริยา ขยายคุณศัพท์ และขยายกริยาวิเศษณ์ด้วยกันเอง
6. คำบุพบท (Preposition) คือ คำที่ใช้บอกตำแหน่ง วันเวลา ทิศทาง สถานที่ หรือแสดงความสัมพันธ์ระหว่างคำ กลุ่มคำ หรือประโยค

7. คำสันธาน (Conjunction) คือ คำที่ใช้เชื่อมคำกับคำ กลุ่มคำกับกลุ่มคำ หรือประโยคกับประโยค

8. คำอุทาน (Interjection) คือ คำที่ใช้บ่งบอกอารมณ์ ความรู้สึกที่เกิดขึ้นในตอนนั้น

2.2 งานวิจัยที่เกี่ยวข้อง

ในส่วนนี้จะเป็นการอธิบายถึงงานวิจัยที่เกี่ยวข้องการกับงานวิจัยนี้ โดยจะสามารถแบ่งออกเป็น 2 ส่วน คือ งานวิจัยที่เกี่ยวข้องกับการวิเคราะห์คำอธิบายรายวิชา และ งานวิจัยที่เกี่ยวข้องกับการเปรียบเทียบความเหมือนกันระหว่างคำ, ข้อความ หรือเอกสาร โดยจะอธิบายถึงงานวิจัยที่เกี่ยวข้องในแต่ละส่วน ดังต่อไปนี้

2.2.1 งานวิจัยที่เกี่ยวข้องกับการวิเคราะห์คำอธิบายรายวิชา

(Starr, Manaris, & Stalvey, 2008) นำเสนอวิธีการสำหรับการประเมินถึงการเรียนรู้ที่ผู้เรียนได้รับกับเนื้อหาที่ถูกสอน หลังจากที่ได้เรียนในชั้นเรียนนั้น ๆ โดยได้ทำการแบ่งกลุ่มระดับของการเรียนรู้ทั้งหมด 6 กลุ่ม ดังต่อไปนี้ 1) Recall เป็นระดับที่บ่งบอกว่าผู้เรียนเข้าใจถึงสิ่งที่ตนเองได้เรียนไป 2) Comprehension เป็นระดับที่ผู้เรียนสามารถถ่ายทอดความรู้ที่ตนเองได้รับให้กับผู้อื่นได้ 3) Application เป็นระดับที่ผู้เรียนสามารถนำความรู้ที่ตนเองได้รับไปประยุกต์ใช้ได้ 4) Analysis เป็นระดับที่ผู้เรียนสามารถวิเคราะห์ถึงโครงสร้างการการทำงานของสิ่งที่เรียนได้ 5) Synthesis เป็นระดับที่ผู้เรียนสามารถนำความรู้ต่าง ๆ ที่ได้เรียนรู้มาใช้งานร่วมกันได้ และ 6) Evaluation เป็นระดับที่ผู้เรียนสามารถทำการประเมินถึงแนวคิดของสิ่งที่เรียนรู้ได้ โดยในงานวิจัยนี้จะดำเนินการทดลองกับผู้เรียนในหลักสูตรวิทยาการคอมพิวเตอร์ ว่าผู้เรียนแต่ละคนมีความรู้ในสิ่งได้เรียนรู้ในระดับใด ซึ่งจากการประเมินถึงระดับความรู้ของผู้เรียนแต่ละคน จะช่วยให้ผู้สอนและ/หรืออาจารย์ที่ทำการสอนในรายวิชานั้น ๆ ทราบถึงเนื้อหาในส่วนที่ผู้เรียนมีระดับของการเรียนรู้ที่น้อย เพื่อนำมาปรับปรุงและพัฒนาให้เนื้อหาในส่วนนั้นมีความเหมาะสม และสามารถทำความเข้าใจได้ง่ายยิ่งขึ้น

(Homa et al., 2013) นำเสนอการวิเคราะห์เนื้อหาในรายวิชาทางด้านจิตวิทยาเบื้องต้น จากการวิเคราะห์ทำให้ผู้สอนและ/หรืออาจารย์ที่สอนในรายวิชานั้น ๆ ทราบถึงวัตถุประสงค์ของเนื้อหาที่จะถ่ายทอดให้ผู้เรียน เพื่อช่วยให้ผู้เรียนมีความรู้พื้นฐานและมีกระบวนการคิดวิเคราะห์เกี่ยวกับด้านจิตวิทยาเบื้องต้น อีกทั้งยังสามารถช่วยให้ผู้เรียนบางคนนำองค์ความรู้ที่ได้ไปต่อยอดกับรายวิชาขั้นสูงของศาสตร์ทางด้านจิตวิทยาได้ โดยในงานวิจัยนี้จะแบ่งกลุ่มของวัตถุประสงค์ที่ได้รับจากรายวิชาหนึ่ง ๆ ออกเป็น 7 กลุ่ม คือ 1) กลุ่มรายวิชาที่สอนเกี่ยวกับประวัติและขอบเขตของจิตวิทยา เป็น

รายวิชาที่สอนเกี่ยวกับประวัติศาสตร์ของจิตวิทยาและอาชีพที่สามารถทำได้ 2) กลุ่มรายวิชาที่สอนเกี่ยวกับกฎระเบียบในการวิจัย ซึ่งประกอบด้วย วิธีการ, สถิติ, การคิดเชิงวิพากษ์ และรูปแบบการเขียน 3) กลุ่มรายวิชาที่สอนเกี่ยวกับสรีรวิทยา ประกอบด้วย ประสาทวิทยา, สติ, ความรู้สึก และ การรับรู้ 4) กลุ่มรายวิชาที่สอนเกี่ยวกับการเรียนรู้ ซึ่งประกอบด้วย ความจำ, การคิด, ความฉลาดและภาษา 5) กลุ่มรายวิชาที่สอนเกี่ยวกับการรักษาผู้ป่วย 6) กลุ่มรายวิชาที่สอนเกี่ยวกับสังคม, การพัฒนาบุคลิกภาพ, ความสัมพันธ์ระหว่างบุคคล 7) กลุ่มรายวิชาที่สอนเกี่ยวกับพัฒนาการในช่วงเวลาต่าง ๆ รวมถึงการเลี้ยงดูบุตร โดยในการทดลองจะนำรายวิชาทั้งสิ้น 158 รายวิชามาทำการจัดกลุ่มตามที่ได้แบ่งไว้ข้างต้น ซึ่งผลลัพธ์จากการทดลองแสดงให้เห็นว่ารายวิชาส่วนใหญ่จะเป็นรายวิชาที่สอนเกี่ยวกับการเรียนรู้ (กลุ่มที่ 4) นอกจากนี้ยังทำการวิเคราะห์ถึงเวลาที่ใช้ในการสอนในแต่ละกลุ่ม ซึ่งจากการทดลองทำให้ทราบว่ารายวิชาที่สอนเกี่ยวกับการเรียนรู้เป็นรายวิชาที่ต้องใช้เวลาสอนมากกว่ารายวิชาในกลุ่มอื่น ๆ

(Chung & Kim, 2016) นำเสนอการวิเคราะห์รายวิชาในหลักสูตรหนึ่ง ๆ ซึ่งจะเป็นการวิเคราะห์ถึงวัตถุประสงค์ที่ผู้เรียนจะได้รับจากการเรียนรู้ในแต่ละรายวิชา โดยการนำรายวิชาทั้งหมดในหลักสูตรมาสร้างเป็นออนโทโลยีของหลักสูตรนั้น ๆ ซึ่งออนโทโลยีที่สร้างขึ้นจะประกอบด้วย ชื่อหลักสูตร, รายวิชาในหลักสูตร จะสามารถแบ่งได้เป็น 2 ส่วนคือ รายวิชาปกติ และ รายวิชาที่ต้องผ่านการเรียนรู้จากรายวิชาหนึ่ง ๆ มาก่อน และ วัตถุประสงค์ที่จะถูกสอนในรายวิชานั้น จากนั้นนำออนโทโลยีที่สร้างขึ้นมาวิเคราะห์ร่วมกับโมเดลความก้าวหน้าในการเรียนรู้ของผู้เรียนที่สร้างขึ้นโดยโมเดลของ Bloom จากการนำข้อมูลทั้ง 2 ส่วนมาวิเคราะห์ร่วมกัน ทำให้สามารถแนะนำรายวิชาที่เกี่ยวข้องและเหมาะสมกับผู้เรียนคนคนหนึ่ง ๆ ได้

(Dai, Asano, & Yoshikawa, 2016) นำเสนอระบบแนะนำคอร์สเรียนออนไลน์ให้กับผู้เรียน โดยให้การสมมติฐานว่าในปัจจุบันคอร์สเรียนออนไลน์ได้มีอยู่เป็นจำนวนมาก ทำให้ผู้เรียนไม่สามารถที่จะตัดสินใจในการเลือกลงเรียนคอร์สที่เหมาะสมกับตัวของผู้เรียนได้ จึงจำเป็นต้องสร้างระบบสำหรับการแนะนำคอร์สเรียนออนไลน์ที่เหมาะสมกับผู้เรียน โดยคอร์สเรียนออนไลน์ที่นำมาใช้ในงานวิจัยนี้จะเป็นคอร์สที่เกี่ยวกับศาสตร์ทางด้านวิทยาการคอมพิวเตอร์ โดยการนำคอร์สเรียนต่าง ๆ มาทำการวิเคราะห์ถึงเนื้อหาที่ถูกสอนและประโยชน์ที่ผู้เรียนจะได้รับ โดยการสกัดกลุ่มของคำนามที่ปรากฏขึ้นในแต่ละเนื้อหาของคอร์สนั้น ๆ จากนั้นนำกลุ่มคำที่สกัดได้มาทำการวิเคราะห์ร่วมกับข้อมูลของผู้เรียนแต่ละคน ได้แก่ วัตถุประสงค์ที่จะได้รับจากการเรียน และ ความสามารถในการเรียนรู้ เพื่อทำการแนะนำคอร์สเรียนออนไลน์ที่มีความเหมาะสมให้กับผู้เรียนที่สนใจจะลงทะเบียนเรียน

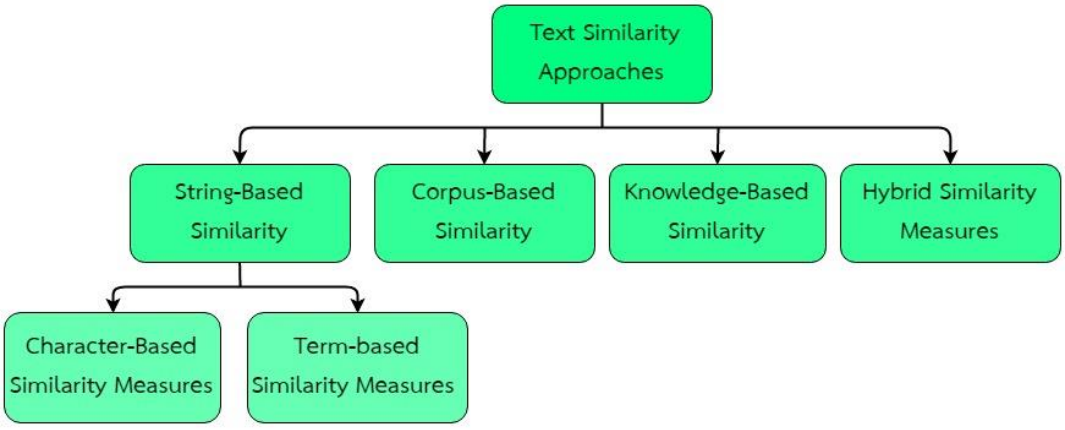
(Nuntawong, Namahoot, & Brückner, 2017) นำเสนอวิธีการวิเคราะห์เนื้อหาในคำอธิบายรายวิชาหนึ่ง ๆ จากการประยุกต์ใช้เทคนิคการเปรียบเทียบระหว่างออนโทโลยีแบบผสมผสาน ในงานวิจัยนี้จะทำการนำออนโทโลยีของคำอธิบายรายวิชาในหลักสูตรวิทยาการคอมพิวเตอร์ที่ถูกสร้างขึ้นโดย TQF: HEd มาเปรียบเทียบกับออนโทโลยีของคำอธิบายรายวิชาในหลักสูตรวิทยาการคอมพิวเตอร์ที่ถูกสร้างขึ้นโดยงานวิจัยนี้ โดยการเปรียบเทียบรายวิชาหนึ่ง ๆ ระหว่าง 2 ออนโทโลยีจะมีวิธีเปรียบเทียบ 2 วิธีการ คือ 1) วิธีการเปรียบเทียบเชิงความหมายระหว่างโหนดหนึ่ง ๆ ที่เป็นเนื้อหาของคำอธิบายรายวิชา และ 2) วิธีการเปรียบเทียบเชิงโครงสร้าง โดยการเปรียบเทียบทั้ง 2 ออนโทโลยีกับออนโทโลยีที่ถูกสร้างขึ้นจากขึ้นจากเนื้อหาของ Computer Science Curricula 2013 ซึ่งเป็นแนวทางในการสร้างหลักสูตรวิทยาการคอมพิวเตอร์ที่เขียนขึ้นโดย ACM และ IEEE โดยจะถูกเรียกว่า SKOS ถ้าพบว่าออนโทโลยีของ TQF: HEd และออนโทโลยีที่สร้างขึ้นโดยงานวิจัยนี้ มีโครงสร้างที่เหมือนกันกับออนโทโลยีของ SKOS จะถือว่าออนโทโลยีของ TQF: HEd และออนโทโลยีที่สร้างขึ้นโดยงานวิจัยนี้มีความเหมือนกัน จากการนำออนโทโลยีของคำอธิบายรายวิชาหนึ่ง ๆ ทางด้านวิทยาการคอมพิวเตอร์ทั้ง 2 ออนโทโลยีมาเปรียบเทียบกัน จะช่วยให้เนื้อหาของคำอธิบายรายวิชาในรายวิชาหนึ่ง ๆ มีความเป็นมาตรฐานมากยิ่งขึ้น

(Shamsi, ul Hassan, Bawany, & Shoaib, 2018) นำเสนอการออกแบบเนื้อหาในรายวิชา Big Data เนื่องจากเทคโนโลยีที่เกี่ยวข้องกับ Big Data มีการปรับปรุงและพัฒนาอยู่สม่ำเสนอ ดังนั้นจึงมีการออกแบบเนื้อหาของรายวิชาที่เกี่ยวข้องกับ Big Data ให้มีความเหมาะสมกับเทคโนโทยีในปัจจุบันมากยิ่งขึ้น โดยในการออกแบบเนื้อของรายวิชาจะมีการตั้งเป้าหมายอยู่ 4 เป้าหมาย คือ 1) การทำให้ผู้เรียนได้รับองค์ความรู้พื้นฐานที่เกี่ยวข้องกับ Big Data อย่างชัดเจน 2) การทำให้ผู้เรียนเข้าใจถึงข้อมูลเชิงลึกในศาสตร์ทางด้าน Big Data 3) การพัฒนาทักษะด้วยการลงมือปฏิบัติ และ 4) การประยุกต์ใช้ทักษาะที่ได้เรียนรู้ในชั้นเรียนในการแก้ไขปัญหาจริง โดยที่ในแต่ละเป้าหมายจะประกอบด้วยเนื้อหาต่าง ๆ ที่จะถูกสอนในชั้นเรียน ในการดำเนินการทดลองจะทำการสอนรายวิชา Big Data ที่ได้ออกแบบด้วยเป้าหมายข้างต้นในชั้นเรียน จากนั้นทดสอบประสิทธิภาพของผู้เรียนที่ได้รับองค์ความรู้จากชั้นเรียนผ่านการเก็บคะแนนจาก การปฏิบัติในห้องแลป, การสอบกลางภาค, การสอบปลายภาค และ คะแนนจากโปรเจ็ค จากการเก็บคะแนนทั้ง 4 ส่วน จะแสดงให้เห็นว่าผู้เรียนที่ได้รับองค์ความรู้จากรายวิชาที่ได้ออกแบบไว้สามารถทำคะแนนในแต่ละส่วนได้เป็นจำนวนมาก

(Apatu et al., 2020) นำเสนอวิธีการวิเคราะห์คำอธิบายรายวิชาในหลักสูตรทางด้านสาธารณสุข เพื่อให้ทราบถึงวัตถุประสงค์ที่ได้เรียนรู้ในรายวิชาหนึ่ง ๆ มีความสอดคล้องกับความต้องการในการทำงานด้านสาธารณสุขมากน้อยเพียงใด โดยรายวิชาที่นำมาใช้ในงานวิจัยนี้จะเป็นรายวิชาในหลักสูตรทางด้านสาธารณสุขของประเทศแคนนาดา ที่มีทั้งสิ้น 267 รายวิชา จาก 18 มหาวิทยาลัย จากการวิเคราะห์คำอธิบายวิชาทั้งหมดทำให้ทราบว่า รายวิชาที่เกี่ยวข้องกับวิทยาศาสตร์สาธารณสุขและการประเมินและการวิเคราะห์ เป็นรายวิชาที่ถูกสอนมากที่สุดในทุก ๆ มหาวิทยาลัย ซึ่งบ่งบอกได้ว่าการทำงานในด้านสาธารณสุขมีความต้องการบุคลากรที่มีองค์ความรู้ที่เกี่ยวข้องกับวิทยาศาสตร์สาธารณสุข และการประเมินและการวิเคราะห์เป็นจำนวนมาก อาทิเช่น องค์ความรู้ทางด้านระบาดวิทยา, ชีวสถิติ, วิธีการวิจัย และพื้นฐานด้านสาธารณสุข โดยองค์ความรู้เหล่านี้เป็นองค์ความรู้ที่สำคัญสำหรับบทบาทในการทำงานด้านสาธารณสุข เช่น นักระบาดวิทยา นักวิเคราะห์การวิจัย และ โปรโมเตอร์สุขภาพ เป็นต้น

2.2.2 งานวิจัยที่เกี่ยวข้องกับการเปรียบเทียบความเหมือนกันระหว่างคำ, ข้อความ หรือเอกสาร

ปัจจุบันวิธีการเปรียบเทียบเทียบเพื่อหาความเหมือนกันระหว่างคำ, ข้อความ หรือเอกสาร ได้ถูกนำมาประยุกต์ใช้ในหลาย ๆ การดำเนินงาน ทำให้เกิดวิธีการเปรียบเทียบในรูปแบบต่าง ๆ ขึ้นอย่างมากมาย จากเทคนิคที่มีอยู่อย่างหลากหลายในปัจจุบัน (Gomaa & Fahmy, 2013; Vijaymeena & Kavitha, 2016; J. Wang & Dong, 2020) ได้ทำการจัดหมวดหมู่ของวิธีการเปรียบเทียบข้อความไว้ 4 วิธี ซึ่งจะแสดงในภาพที่ 2 ดังต่อไปนี้



ภาพที่ 2 วิธีการเปรียบเทียบข้อความในรูปแบบต่าง ๆ

2.2.2.1 การเปรียบเทียบแบบสายอักขระ (String-Based Similarity)

เป็นเทคนิคการเปรียบเทียบระหว่างคำหรือข้อความในลักษณะของการดำเนินการเปรียบเทียบแบบลำดับของตัวอักษร ตำแหน่งของตัวอักษร ความเหมือนกันของคำในระหว่าง 2 ข้อความ ซึ่งการดำเนินการเปรียบเทียบระหว่างคำจะทำการสร้างเมทริกซ์ (metric) ขึ้นมาเพื่อใช้ในการเปรียบเทียบเพื่อหาค่าของความเหมือนกันของคำทั้ง 2 โดยการเปรียบเทียบแบบสายอักขระสามารถแบ่งออกได้ 2 ส่วนดังนี้

a) การเปรียบเทียบความเหมือนกันของอักขระ (Character-Based Similarity Measures)

เป็นการคำนวณหาความเหมือนกันระหว่างคำทั้ง 2 ในลักษณะของการเปรียบเทียบแบบตัวอักษร อาทิเช่น การเปรียบเทียบความเหมือนกันในการดำเนินการเพิ่ม, ลบ หรือแทนที่ ในแต่ละตัวอักษรระหว่างคำ (Hall & Dowling, 1980) การเปรียบเทียบความเหมือนกันระหว่างคำ 2 คำจากกลุ่มของอักขระ (N-gram) ที่เหมือนกันระหว่างคำ 2 คำ (Barrón-Cedeno, Rosso, Agirre, & Labaka, 2010) เป็นต้น โดยการเปรียบเทียบความเหมือนกันของอักขระมีวิธีการคำนวณหาค่าความเหมือนกันอยู่หลายวิธีการ เช่น วิธีการ Jaro distance (Jaro, 1989, 1995) คือวิธีการที่ใช้วัดค่าความเหมือนกันระหว่างตัวอักษรจากจำนวนตัวอักษรที่มีเหมือนกันในทั้ง 2 คำ โดยในการคำนวณจะทำการพิจารณาถึงตำแหน่งของตัวอักษรและความยาวของตัวอักษร โดยมีสมการในการคำนวณหาค่าความเหมือนกันดังนี้

$$J(X,Y) = \frac{1}{3} \times \left(\frac{C}{|X|} + \frac{C}{|Y|} + \frac{C-T}{C}\right)$$

กำหนดให้ $|X|$ คือ ความยาวของตัวอักษรของคำที่ 1

$|Y|$ คือ ความยาวของตัวอักษรของคำที่ 2

C คือ จำนวนของตัวอักษรที่เหมือนกันและมีตำแหน่งตรงกัน

T คือ จำนวนเต็มครึ่งหนึ่งของตัวอักษรที่ตรงกันแต่มีลำดับไม่ตรงกัน ซึ่งหากทั้ง 2 คำมีตัวอักษรที่เหมือนกันแต่มีตำแหน่งไม่ตรงกัน ถ้าระยะห่างระหว่างอักษรที่เหมือนกันมีค่าไม่เกิน r จะถือว่าอักษรนั้นมีตำแหน่งตรงกัน โดยที่สามารถคำนวณหาค่าของ r ได้จากสมการ

$$r = \left[\frac{max(|X|,|Y|)}{2}\right] - 1$$

วิธีการ Bi-Jaccard ซึ่งเป็นวิธีการคำนวณหาค่าความเหมือนกันของคำแบบกลุ่มของตัวอักษรในลักษณะของการแบ่งตัวอักษรแบบ 2 ตัวอักษร (bi-grams) โดยการประยุกต์ใช้วิธีการคำนวณแบบ Jaccard similarity เข้ามาร่วมดำเนินการหาค่าความเหมือนด้วย ดังสมการต่อไปนี้

$$Bi - Jaccard(X,Y) = \frac{|bigr(X) \cap bigr(Y)|}{|bigr(X) \cup bigr(Y)|}$$

กำหนดให้ $bigr(X)$ คือ กลุ่มของตัวอักษรที่ถูกแบ่งแบบ 2 ตัวอักษรของคำที่ 1

$bigr(Y)$ คือ กลุ่มของตัวอักษรที่ถูกแบ่งแบบ 2 ตัวอักษรของคำที่ 2

วิธีการ Bi-Dice (Brew & McKelvie, 1996) ซึ่งเป็นวิธีการคำนวณหาค่าความเหมือนกันของคำแบบกลุ่มของตัวอักษรในลักษณะของการแบ่งตัวอักษรแบบ 2 ตัวอักษร (bi-grams) โดยการประยุกต์ใช้วิธีการคำนวณแบบ Dice's coefficient เข้ามาร่วมดำเนินการหาค่าความเหมือนด้วย ดังสมการต่อไปนี้

$$Bi - Dice(X,Y) = \frac{2 \times |bigr(X) \cap bigr(Y)|}{|bigr(X)| + |bigr(Y)|}$$

กำหนดให้ $bigr(X)$ คือ กลุ่มของตัวอักษรที่ถูกแบ่งแบบ 2 ตัวอักษรของคำที่ 1

$bigr(Y)$ คือ กลุ่มของตัวอักษรที่ถูกแบ่งแบบ 2 ตัวอักษรของคำที่ 2

b) การเปรียบเทียบความเหมือนกันของคำ (Term-Based Similarity Measures)

คือการคำนวณหาค่าความเหมือนกันของคำระหว่างข้อความ 2 ข้อความ เช่น วิธีการ Jaccard similarity ที่เป็นการคำนวณหาความเหมือนกันของคำสำคัญหรือคำหลักของทั้ง 2 ข้อความ ดังสมการ

$$Jaccard\ (X,Y) = \frac{|X \cap Y|}{|X \cup Y|}$$

กำหนดให้ X คือ ข้อความที่ 1

Y คือ ข้อความที่ 2

วิธีการคำนวณหาความเหมือนกันของคำระหว่างข้อความ 2 ข้อความ อาทิเช่น วิธีการ Dice's coefficient (Dice, 1945) โดยมีการคำนวณหาความเหมือนกันของคำดังสมการต่อไปนี้

$$Dice\ (X,Y) = \frac{2 \times |X \cap Y|}{|X| + |Y|}$$

กำหนดให้ X คือ ข้อความที่ 1

Y คือ ข้อความที่ 2

วิธีการ Matching Coefficient โดยมีการคำนวณหาความเหมือนกันของคำดังสมการต่อไปนี้

$$MC\ (X,Y) = \frac{|X \cap Y|}{max\ (|X|,|Y|)}$$

กำหนดให้ X คือ ข้อความที่ 1

Y คือ ข้อความที่ 2

วิธีการ Overlap coefficient โดยมีการคำนวณหาความเหมือนกันของคำดังสมการต่อไปนี้

$$OC\ (X,Y) = \frac{|X \cap Y|}{min\ (|X|,|Y|)}$$

กำหนดให้ X คือ ข้อความที่ 1

Y คือ ข้อความที่ 2

ซึ่งวิธีการการเปรียบเทียบแบบสายอักขระมีงานวิจัยที่เกี่ยวข้องดังนี้

(Metzler, Dumais, & Meek, 2007) ได้นำเสนอวิธีการเปรียบเทียบเพื่อหาความเหมือนกันระหว่างคำค้นและคำค้นในระบบการค้นคืนข้อมูล (information retrieval) ซึ่งได้แบ่งการเปรียบเทียบออกเป็น 2 ส่วนคือ 1) การเปรียบเทียบความเหมือนกันของคำระหว่างคำค้น 2 คำค้น โดยสามารถแบ่งเกณฑ์การเปรียบเทียบออกเป็น 3 แบบคือ i) Exact คือการเปรียบเทียบในแบบที่คำค้นตั้งต้นและคำค้นที่นำมาเปรียบเทียบต้องเป็นคำคำเดียวกัน เช่น คำค้นตั้งต้นคือ "seattle mariners tickets" และคำค้นที่นำมาเปรียบเทียบคือ "seattle mariners tickets" ii) Phrase คือการเปรียบเทียบในแบบที่คำค้นที่นำมาเปรียบเทียบเป็นสับเซ็ตของคำค้นตั้งต้นและต้องมีลำดับของคำ

ที่เหมือนกัน เช่น คำค้นตั้งต้นคือ “seattle mariners tickets” และคำค้นที่นำมาเปรียบเทียบคือ “seattle mariners” iii) Subset คือการเปรียบเทียบในแบบที่คำค้นที่นำมาเปรียบเทียบเป็นสับเซ็ตของคำค้นตั้งต้นแต่ไม่ต้องมีลำดับของคำที่เหมือนกัน เช่น คำค้นตั้งต้นคือ “seattle mariners tickets” และคำค้นที่นำมาเปรียบเทียบคือ “tickets seattle” และส่วนที่ 2) คือการเปรียบเทียบความเหมือนกันระหว่างคำค้นโดยวิธีการคำนวณหาความน่าจะเป็นของ 2 คำค้นที่นำมาเปรียบเทียบกับ โดยผลลัพธ์ของการทดลองแสดงให้ถึงประสิทธิภาพและความถูกต้องของการดำเนินการเปรียบเทียบจากการกับวิธีการเปรียบเทียบแบบต่าง ๆ

(Xu & Xu, 2011) นำเสนอวิธีการเปรียบเทียบระหว่างคำค้น 2 คำค้น ที่ซึ่งจะทำการหาความเหมือนกันของคำค้นในลักษณะของการเปรียบเทียบแบบ n-gram โดยการสร้างเวกเตอร์ของแต่ละคำค้นที่นำมาเปรียบเทียบ จากนั้นนำเวกเตอร์ที่ได้มาทำการเปรียบเทียบเพื่อหาความเหมือนกันของทั้ง 2 คำค้นโดยการประยุกต์ใช้วิธีการคำนวณหาค่าของความเหมือนกันระหว่างคำแบบ cosine similarity นอกจากนี้ยังทำการประยุกต์ใช้ชุดข้อมูลทดสอบของการวัดความเหมือนกันของคำค้นจากการรวบรวมคู่ของคำค้นที่นำมาเปรียบเทียบจากข้อมูลของการคลิ๊กลิงค์เข้าหน้าเว็บไซต์หนึ่ง ๆ เมื่อใช้คำค้นหนึ่ง ๆ จากการทดลองแสดงให้เห็นว่าการเปรียบเทียบระหว่างคำค้นโดยการเปรียบเทียบในลักษณะของ n-gram สามารถทำการเปรียบเทียบระหว่างคู่ของคำค้นที่มีการสะกดผิด การใช้คำย่อ คำที่เป็นรากศัพท์ คำที่มีความหมายคล้ายกัน และลำดับของตัวเลขได้เป็นที่น่าพึงพอใจ นอกจากนี้ยังทำการเรียงลำดับของคำค้นที่มีความเกี่ยวข้องกับคำค้นที่เป็นข้อมูลนำเข้า ที่ซึ่งจะแสดงให้เห็นถึงความถูกต้อง (recall) ของคำค้นที่ได้รับ และยังได้แสดงให้เห็นถึงประสิทธิภาพของการดำเนินงานจากการเปรียบเทียบวิธีที่ได้นำเสนอกับการคำนวณหาค่าความเหมือนระหว่างข้อความแบบการใช้ cosine similarity แบบเดี่ยว ๆ และการใช้ Pearson coefficient แบบเดี่ยว ๆ ที่ซึ่งวิธีการที่นำเสนอสามารถดำเนินงานได้ดีกว่าทั้ง 2 วิธีที่กล่าวมาข้างต้น

(Islam, Milios, & Kešelj, 2012) นำเสนอวิธีการเปรียบเทียบระหว่างข้อความในลักษณะของการเปรียบเทียบแบบ n-gram ที่ซึ่งได้ประยุกต์ใช้วิธีการของ Google Tri-grams เข้ามาใช้ในการดำเนินการเปรียบเทียบระหว่างข้อความ โดยจะทำการแบ่งแบบ tri-grams จากตัวอักษรเริ่มต้นจนถึงอักขระตัวสุดท้ายของข้อความแต่ละข้อความที่นำมาเปรียบเทียบ จากนั้นนำมาคำนวณหาความเหมือนกันของข้อความที่นำมาเปรียบเทียบกัน ที่ซึ่งจากการดำเนินการทดลองแสดงให้เห็นถึงประสิทธิภาพของการเปรียบเทียบข้อความที่นำเสนอมีประสิทธิภาพที่เหนือกว่าการดำเนินการ

เปรียบเทียบในรูปแบบต่าง ๆ และยังได้ค่าของผลลัพธ์จากการคำนวณหาความเหมือนกันระหว่างข้อความที่ใกล้เคียงกับการตัดสินโดยผู้เชี่ยวชาญ

(Gali, Mariescu-Istodor, & Fränti, 2016) ได้ทำการประยุกต์ใช้วิธีการคำนวณ 21 วิธีการ ซึ่งเป็นวิธีการที่ใช้ในการคำนวณหาค่าความเหมือนกันจากการเปรียบเทียบข้อความแบบสายอักขระ อาทิเช่น Jaro-Winkler, Bi-Jaccard, Trigrams, Jaccard, Dice, Matching Coefficient, Overlap Coefficient, Rouge-N, TF-IDF, Euclidean, Manhattan, Soft-TFIDF เป็นต้น เพื่อดูว่าวิธีการใดที่เหมาะสมกับการเปรียบเทียบข้อมูลที่เป็นแบบข้อความที่มีขนาดสั้น โดย Najlah et al., ได้ทำการแบ่งกลุ่มวิธีการคำนวณหาความเปรียบเทียบของข้อความไว้ 4 กลุ่มดังนี้ i) Character-based measures, ii) Q-grams, iii) Token-based measures และ iiii) Mixed measures โดยผลลัพธ์จากทดลองทำให้พบว่าวิธีการคำนวณแบบ Soft-TFIDF ซึ่งเป็นวิธีการคำนวณที่รวมวิธีการ TF-IDF (term frequency-inverse document frequency) ที่อยู่ในกลุ่มของเปรียบเทียบข้อความแบบ Token-based measures และวิธีการ Jaro-Winkler ที่อยู่ในกลุ่มของการเปรียบเทียบข้อความแบบ Character-based measures เข้าไว้ด้วยกัน เป็นวิธีการเปรียบเทียบข้อความที่เหมาะสมและสามารถดำเนินการได้ดีที่สุดกับข้อมูลแบบข้อความที่มีขนาดสั้น

(Maher & Joshi, 2016) ได้นำเสนอวิธีจัดกลุ่มเอกสารโดยการประยุกต์ใช้วิธีการคำนวณหาค่าความเหมือนกันของคำเข้ามาทำการหาค่าความเหมือนกันของเอกสาร ซึ่งได้แก่วิธีการคำนวณแบบ Euclidean Distance, Cosine Similarity และ Similarity Measure for Text Processing จากนั้นนำวิธีการคำนวณแต่ละวิธีการมาดำเนินการร่วมกับวิธีการการจัดกลุ่มแบบ K-NN based classification, Naïve Bayes classification และ K-means clustering ที่ซึ่งผลลัพธ์จากการดำเนินการทดลองแสดงให้เห็นว่าการนำวิธีการคำนวณหาค่าความเหมือนกันของคำแบบ Similarity Measure for Text Processing เมื่อนำมาประยุกต์ใช้ร่วมกับวิธีการจัดกลุ่มทั้ง 3 วิธีที่กล่าวมาข้างต้น สามารถดำเนินการได้อย่างมีประสิทธิภาพมากกว่าการนำวิธีการคำนวณหาค่าความเหมือนกันของคำแบบ Euclidean Distance และ Cosine Similarity เข้ามาประยุกต์ใช้ร่วมกับวิธีการจัดกลุ่มทั้ง 3 วิธี

(Tessem, 2019) นำเสนอขั้นตอนวิธีสำหรับการแนะนำบทความข่าวต่าง ๆ ให้กับนักข่าว โดยการพิจารณาถึงแง่มุมของข่าวที่นักข่าวได้ทำการระบุไว้กับบทความข่าวที่มีอยู่ จากการวิเคราะห์และเปรียบเทียบแง่มุมจากนักข่าวกับเนื้อหาของข่าวในบทความข่าวหนึ่ง ๆ ซึ่งในการดำเนินการเปรียบเทียบ ได้ประยุกต์ใช้วิธีการเปรียบเทียบโดยการนับความถี่ (TF-IDF) ของคำที่เหมือนกัน

ระหว่างแง่มุมของนักข่าวและเนื้อหาของข่าวในบทความต่าง ๆ จากนั้นนำแง่มุมของนักข่าวและเอกสารที่ได้จากการเปรียบเทียบ มาคำนวณหาค่าความเหมือนกันเพื่อที่จะทำการจัดอันดับของบทความข่าวให้กับนักข่าวคนนั้น ๆ โดยบทความข่าวที่จะถูกส่งให้กับนักข่าวจะเป็นบทความข่าวที่มีคะแนนของความเหมือนกัน 10 อันดับแรกเท่านั้น

2.2.2.2 การเปรียบเทียบเชิงความหมายจากแหล่งข้อมูลที่สร้างขึ้นเอง (Corpus-Based Similarity)

เป็นการคำนวณหาความเหมือนกันของคำโดยการเปรียบเทียบความหมายระหว่างคำจากคลังคำศัพท์หรือแหล่งข้อมูลที่สร้างขึ้นโดยผู้เชี่ยวชาญทางด้านภาษาหนึ่ง ๆ และ/หรือศาสตร์หนึ่ง ๆ โดยวิธีการเปรียบเทียบเชิงความหมายจากแหล่งข้อมูลที่สร้างขึ้นเองมีงานวิจัยที่เกี่ยวข้องดังต่อไปนี้

(Das, Chong, Eadon, & Srinivasan, 2004) ได้นำเสนอวิธีการเปรียบเทียบเชิงความหมายเพื่อใช้ในการค้นคืนข้อมูลจากฐานข้อมูล โดยการนำข้อมูลที่อยู่ในฐานข้อมูลมาทำการสร้างออนโทโลยีเพื่อทำให้ทราบถึงความสัมพันธ์ของแต่ละข้อมูล ซึ่งเมื่อนำข้อมูลมาสร้างเป็นออนโทโลยีแล้วจะช่วยให้สามารถทำการค้นคืนข้อมูลที่เกี่ยวข้องและ/หรือที่มีความสัมพันธ์กับคำค้นได้มีประสิทธิภาพยิ่งขึ้น อาทิเช่น เมื่อผู้ใช้งานป้อนคำค้นเข้ามาเพื่อหาข้อมูลหนึ่ง ๆ ถ้าฐานข้อมูลที่มีอยู่ไม่มีข้อมูลที่ตรงกับคำค้น การค้นคืนข้อมูลจะไม่สามารถทำได้ แต่ถ้านำข้อมูลจากฐานข้อมูลมาสร้างเป็นออนโทโลยีเมื่อผู้ใช้งานป้อนคำค้นเข้ามา เพื่อหาข้อมูลหนึ่ง ๆ ซึ่งเมื่อคำค้นที่ป้อนเข้ามาไม่ตรงกับข้อมูลที่มีอยู่ในฐานข้อมูล การค้นคืนจะทำการเปรียบเทียบความหมายและความสัมพันธ์ของคำค้นกับข้อมูลที่มีอยู่แล้วแสดงผลลัพธ์ที่เกี่ยวข้องกับคำค้นนั้น ๆ ให้กับผู้ใช้งาน ซึ่งงานวิจัยนี้ได้ทำการสร้างออนโทโลยีจากข้อมูลของประเภทอาหารในร้านอาหารต่างๆ จากนั้นทำการทดลองโดยการใช้ภาษา SQL ในการป้อนคำค้นกับออนโทโลยีของฐานข้อมูลที่สร้างขึ้น จากผลการทดลองได้แสดงให้เห็นถึงเวลาในการค้นคืนข้อมูลที่รวดเร็วยิ่งขึ้น

(Guo, Fan, Ai, & Croft, 2016) ได้นำเสนอการเปรียบเทียบทางความหมายสำหรับการค้นคืนข้อมูล ซึ่งเป็นการเปรียบเทียบจากความหมายระหว่างคำค้นและเอกสาร โดยเริ่มจากทำการตัด (pruning) เอกสารที่มีคำที่เกี่ยวข้องกับคำค้นที่น้อยมากออกไปเพื่อลดความซ้ำซ้อนในการคำนวณ โดยการเปรียบเทียบความหมายของคำค้นกับหัวข้อของเอกสารและคำที่อยู่ในเอกสาร ซึ่งเมื่อมีคำค้นเข้ามาจะทำการกำหนดเอกสารที่เกี่ยวข้องกับคำค้นนั้น ๆ จากนั้นทำการทำดัชนี (indexing) โดยใช้วิธี k-nearest-neighbor กับแต่ละคำในเอกสารหนึ่ง ๆ เพื่อจัดลำดับเอกสารที่เกี่ยวข้องกับคำค้น

โดยเอกสารที่มีหัวข้อและคำที่เกี่ยวข้องกับคำค้นมาก ๆ จะทำการจัดเก็บไว้เพื่อทำดัชนีและความรวดเร็วในการเข้าถึง โดยในการทดลอง Guo et al., ได้ทำการทดลองกับ 3 ชุดข้อมูล ได้แก่ Robust04, GOV2, Clueweb-09-Cat-B และทำการเปรียบเทียบโมเดลที่ทำขึ้นกับโมเดลวิธีการต่าง ๆ ใน 3 ชุดข้อมูลซึ่งจากการทดลองและเปรียบเทียบกับโมเดลของวิธีการต่าง ๆ จากผลการทดลองได้แสดงให้เห็นถึงการดำเนินงานที่มีประสิทธิภาพมากกว่าโมเดลของวิธีการต่าง ๆ

(Liu, Xiong, Sun, & Liu, 2018) ได้นำเสนอขั้นตอนวิธี “EDRM (Entity-Duet Neural Ranking Model)” ซึ่งเป็นขั้นตอนวิธีสำหรับการค้นคืนเอกสารจากคำค้นที่ผู้ใช้งานทำการค้นหาเอกสารบนหน้าเว็บไซต์ โดยการขั้นตอนวิธี “EDRM” เป็นการดำเนินการร่วมกันระหว่าง 2 วิธีการคือ 1) การเปรียบเทียบเชิงความหมายระหว่างคำค้นและเนื้อหาในเอกสารหนึ่ง ๆ โดยการประยุกต์ใช้วิธีการ embedding เข้ามาช่วยในการดำเนินการเปรียบเทียบเชิงความหมายระหว่างคำค้นและเอกสาร และ 2) การจัดอันดับความเกี่ยวของเอกสารกับคำค้น ซึ่งจากการเปรียบเทียบเชิงความหมายระหว่างคำค้นและเอกสารก็เป็นส่วนช่วยที่ดีในการจัดอันดับของเอกสาร นอกจากนี้ในการจัดอันดับของเอกสารยังทำการพิจารณาถึงประวัติการค้นหาเข้ามาร่วมจัดดับด้วย ซึ่งผลลัพธ์ที่ได้จากขั้นตอนวิธี “EDRM” Liu et al., ได้นำมาดำเนินการเปรียบเทียบกับผลลัพธ์ที่ได้จากขั้นตอนวิธี “K-NRM” และขั้นตอนวิธี “Conv-KNRM” เป็นเป็นขั้นตอนวิธีสำหรับการค้นคืนเอกสารเช่นกัน เมื่อพิจารณาถึงผลลัพธ์ที่ได้จากการเปรียบเทียบพบว่าผลลัพธ์ของเอกสารที่ได้จากการค้นคืนด้วยขั้นตอนวิธี “EDRM” ได้ประสิทธิภาพที่เหนือกว่าทั้ง 2 ขั้นตอนวิธี

(Lenz, Ollinger, Sahitaj, & Bergmann, 2019) นำเสนอวิธีการค้นคืนข้อมูลสำหรับระบบการค้นคืนข้อมูล ซึ่งขั้นตอนวิธีสำหรับการค้นคืนข้อมูลในงานวิจัยจะทำการประยุกต์ใช้การสร้างออนโทโลยีของเอกสารหนึ่ง ๆ จากนั้นนำคำค้นที่ผู้ใช้งานป้อนเข้ามาเพื่อค้นหาข้อมูล มาทำการเปรียบเทียบเชิงความหมายระหว่างออนโทโลยีของเอกสารหนึ่ง ๆ กับคำค้น เพื่อให้ได้มาซึ่งเอกสารที่ตรงตามความต้องการของผู้ใช้งานมากที่สุด โดยในการเปรียบเทียบเชิงความหมายจะทำการใช้วิธีการสร้างเวคเตอร์ของระหว่างคู่ของคำหรือประโยคที่นำมาเปรียบเทียบกัน โดยการประยุกต์ใช้ขั้นตอนวิธีของ Word2vec Skip-gram ซึ่งจะเป็นการสร้างเวคเตอร์ของคำจากข้อมูลคลังคำศัพท์ที่ได้จัดเตรียมไว้ จากนั้นทำการคำนวณหาค่าความเหมือนกันโดยการประยุกต์ใช้วิธีการคำนวณแบบ Cosine similarity จากการที่สร้างออนโทโลยีของเอกสารหนึ่ง ๆ แล้วนำออนโทโลยีที่สร้างมาทำการเปรียบเทียบเชิงความหมายกับคำค้นที่ผู้ใช้งานป้อนเข้ามา ทำให้ผลลัพธ์ที่เป็นข้อมูลการค้นคืนที่เกี่ยวข้องกับคำค้นนั้น ๆ มีความถูกต้องและมีความใกล้เคียงกับสิ่งที่ผู้ใช้งานต้องการมากที่สุด

(Sitikhu, Pahi, Thapa, & Shakya, 2019) นำเสนอขั้นตอนวิธีสำหรับการเปรียบเทียบเชิงความหมายเพื่อหาความเหมือนกันระหว่างบทความข่าว โดยในงานวิจัยนี้ได้ทำการคิดค้นวิธีการเปรียบเทียบ 3 วิธี ได้แก่ 1) การเปรียบเทียบเชิงความหมายจากการสร้างเวคเตอร์ของคำด้วยการประยุกต์ใช้วิธีการ TF-IDF (Term Frequency-Inverse Document Frequency) จากนั้นนำเวคเตอร์ของทุกคำในเอกสารมาหาค่าเฉลี่ย เพื่อให้ได้มาซึ่งเวคเตอร์ของแต่ละเอกสาร ต่อมาจะใช้วิธีการคำนวณเพื่อหาค่าความเหมือนกันของเอกสารด้วยวิธีการคำนวณแบบ Cosine similarity 2) การเปรียบเทียบเชิงความหมายจากการสร้างเวคเตอร์ของคำด้วยการประยุกต์ใช้วิธีการของ Word2Vec ซึ่งจะเป็นการสร้างเวคเตอร์ของคำจากข้อมูลของคลังคำศัพท์ที่ได้จัดเตรียมไว้ จากนั้นนำเวคเตอร์ของทุกคำในเอกสารมาหาค่าเฉลี่ย เพื่อให้ได้มาซึ่งเวคเตอร์ของแต่ละเอกสาร ต่อมาจะใช้วิธีการคำนวณเพื่อหาค่าความเหมือนกันของเอกสารด้วยวิธีการคำนวณแบบ Cosine Similarity และ 3) การเปรียบเทียบเชิงความหมายจากการสร้างเวคเตอร์ของคำด้วยการประยุกต์ใช้วิธีการของ Word2Vec ซึ่งจะเป็นการสร้างเวคเตอร์ของคำจากข้อมูลของคลังคำศัพท์ที่ได้จัดเตรียมไว้ จากนั้นนำเวคเตอร์ของทุกคำในเอกสารมาหาค่าเฉลี่ย เพื่อให้ได้มาซึ่งเวคเตอร์ของแต่ละเอกสาร ต่อมาจะใช้วิธีการคำนวณเพื่อหาค่าความเหมือนกันของเอกสารด้วยวิธีการคำนวณแบบ Soft Cosine Similarity โดยในการทดลองของงานวิจัยนี้ จะทำการทดลองเปรียบเทียบความเหมือนกันของบทความข่าวด้วยขั้นตอนวิธีทั้ง 3 วิธีการ ซึ่งจากการทดลองแสดงให้เห็นว่า วิธีการเปรียบเทียบเชิงความหมายในขั้นตอนวิธีที่ 1) คือการสร้างเวคเตอร์ของคำในแต่ละเอกสารด้วยวิธีการของ TF-IDF จากนั้นมาเวคเตอร์ของคำทุกคำมาหาต่าเฉลี่ยเพื่อสร้างเป็นเวคเตอร์ของแต่ละเอกสาร แล้วนำมาคำนวณหาค่าความเหมือนกันด้วยวิธีการของ Cosine similarity ให้ผลลัพธ์ที่มีความถูกต้องเหนือกว่า 2 ขั้นตอนวิธี

(Mohammed & Kadhim, 2020) นำเสนอวิธีการสรุปความจากหลาย ๆ เอกสาร โดยการเปรียบเทียบความเหมือนกันจากคำในเนื้อหาของเอกสารที่นำมาสรุป โดยเริ่มจากการประยุกต์ใช้เทคนิคการประมวลข้อความเบื้องต้น ซึ่งได้แก่ การแบ่งประโยคแต่ละประโยคออกจากกัน, การแปลงตัวอักษรพิมพ์ใหญ่ในภาษษอังกฤษให้เป็นตัวอักษรพิมพ์เล็กในภาษษอังกฤษ, การลบเครื่องหมายวรรคตอน, การกำจัดคำหยุดที่ปรากฏขึ้นในแต่ละประโยค และ การลดรูปของคำให้อยู่ในรูปของรากศัพท์ จากนั้นนำมาเปรียบเทียบส่วนของเนื้อที่เหมือนกันโดยการนับความถี่ของคำที่มีเหมือนกันในเอกสารที่นำมาเปรียบเทียบกัน สุดท้ายทำการประยุกต์ใช้สูตรการคำนวณหาค่าความเหมือนกันระหว่างประโยคแบบ Cosine similarity, Jaccard similarity และ Dice similarity จากการ

ดำเนินการทดลองทำให้ได้มาซึ่งผลลัพธ์ของการสรุปความจากหลาย ๆ เอกสาร ที่มีประสิทธิที่ดีกว่าเมื่อนำไปเปรียบเทียบกับผลลัพธ์ที่ได้จากขั้นตอนวิธีการสรุปความของ SOEA-based และ SOO

(Qurashi, Holmes, & Johnson, 2020) ได้นำเสอนวิธีการเปรียบเทียบความเหมือนกันระหว่างเอกสาร โดยการเปรียบเทียบประสิทธิภาพของผลลัพธ์ที่ได้จากการวิธีการเปรียบเทียบโดยการคำนวณความเหมือนกันระหว่างคำแบบ Jaccard Similarity ซึ่งเป็นวิธีการคำนวณความเหมือนกันระหว่างคำหรือประโยคโดยการพิจารณาถึงลำดับการเกิดขึ้นของตัวอักษรระหว่างคำ, ตัวอักษรที่ปรากฏระหว่าง, ความเหมือนกันของคำระหว่างประโยค เป็นต้น กับวิธีการเปรียบเทียบโดยการคำนวณความเหมือนกันระหว่างคำแบบ Cosine similarity ซึ่งจะเป็นการแปลงคำหนึ่ง ๆ ในประโยคให้อยู่ในรูปของเวคเตอร์ โดยการประยุกต์ใช้วิธีการแปลงคำให้เป็นเวคเตอร์ด้วยวิธีการ Word2vec จากนั้นนำเวคเตอร์ที่ได้มาคำนวณเพื่อหาค่าความเหมือนกันด้วยสมการ ของ Cosine similarity ซึ่งจากผลลัพธ์ของการดำเนินการเปรียบเทียบทั้ง 2 แบบที่กล่าวมาข้างต้น แสดงให้เห็นว่าผลลัพธ์จากการเปรียบเทียบด้วยวิธีการเปรียบเทียบโดยการคำนวณความเหมือนกันระหว่างคำแบบ Cosine similarity ให้ประสิทธิภาพที่ดีกว่าผลลัพธ์วิธีการเปรียบเทียบโดยการคำนวณความเหมือนกันระหว่างคำแบบ Jaccard similarity

(Singh & Singh, 2021) นำเสอนวิธีการเปรียบเทียบความเหมือนกันระหว่างบทความข่าวโดยการนำผลลัพธ์ที่ได้จากวิธีการเปรียบเทียบโดยการคำนวณหาค่าความเหมือนกันแบบ Cosine similarity, วิธีการเปรียบเทียบโดยการคำนวณหาค่าความเหมือนกันแบบ Jaccard similarity และวิธีการเปรียบเทียบโดยการคำนวณหาค่าความเหมือนกันแบบ Euclidean distance มาทำการเปรียบเทียบประสิทธิภาพกัน โดยเริ่มจากการประยุกต์ใช้เทคนิคการประมวลผลข้อความเบื้องต้นต่อมาทำการสกัดกลุ่มของคำนามในแต่ละบทความ จากนั้นนำบทความข่าว 2 บทความที่ผ่านการดำเนินงานทั้ง 2 ขั้นตอนข้างต้นแล้วมาเปรียบเทียบกันด้วยวิธีการเปรียบเทียบโดยการคำนวณหาค่าความเหมือนกันทั้ง 3 วิธีการ โดยจะทำการกำหนดให้วิธีการคำนวณหาค่าความเหมือนกันแบบ Cosine similarity และ Jaccard similarity ให้คำนวณหาค่าความเหมือนกันระหว่างบทความข่าวด้วยการประยุกต์ใช้วิธีการเปรียบเทียบด้วยขั้นตอนวิธี TF-IDF และ วิธีการเปรียบเทียบโดยการคำนวณหาค่าความเหมือนกันแบบ Euclidean distance ให้คำนวณหาค่าความเหมือนกันระหว่างบทความข่าวด้วยการประยุกต์ใช้วิธีการเปรียบเทียบด้วยขั้นตอนวิธี Bag-of-Words ซึ่งผลลัพธ์ที่ได้จากการเปรียบเทียบทั้ง 3 การคำนวณจะแสดงให้เห็นว่าผลลัพธ์ที่ได้จากการวิธีการคำนวณหาค่าความเหมือนกันแบบ Cosine similarity มีประสิทธิภาพที่เหนือว่าผลลัพธ์ที่ได้จากวิธีการคำนวณหา

ค่าความเหมือนกันแบบ Jaccard similarity และ Euclidean distance ทั้งในด้านของความถูกต้อง, ความแม่นยำ และประสิทธิภาพโดยรวม

2.2.2.3 การเปรียบเทียบเชิงความหมายจากแหล่งข้อมูลออนไลน์ (Knowledge-Based Similarity)

เป็นวิธีการคำนวณหาความเหมือนกันระหว่างคำ โดยการเปรียบเทียบจากความหมายของคำจากแหล่งข้อมูลออนไลน์ อาทิเช่น WordNet (Miller, 1995) ซึ่งเป็นคลังคำศัพท์ทางภาษาอังกฤษแบบออนไลน์ที่มีขนาดใหญ่และนิยมนำมาประยุกต์ใช้ในการเปรียบเทียบเชิงความหมายจากแหล่งข้อมูลออนไลน์ โดยที่ WordNet จะบอกถึงความหมายของคำ คำที่มีความหมายเหมือนกัน คำที่มีความหมายคล้ายกัน คำที่มีความหมายตรงข้ามกัน เป็นต้น โดยวิธีการเปรียบเทียบเชิงความหมายจากแหล่งข้อมูลออนไลน์สามารถแบ่งออกกลุ่มเป็น 2 กลุ่ม ได้แก่ 1) การเปรียบเทียบความเหมือนกันของความหมาย และ 2) การเปรียบเทียบความเหมือนกันในระดับความสัมพันธ์ของความหมาย ซึ่งวิธีการเปรียบเทียบเชิงความหมายจากแหล่งข้อมูลที่สร้างขึ้นเองมีงานวิจัยที่เกี่ยวข้องดังต่อไปนี้

(Corley & Mihalcea, 2005) ได้นำเสนอปัญหาของการเปรียบเทียบความเหมือนกันของสายอักขระระหว่าง 2 ข้อความที่ซึ่งเมื่อทำการเปรียบเทียบแล้ว ผลลัพธ์จากการเปรียบเทียบยังคงมีส่วนผิดพลาดอยู่บ้าง ดังนั้น Corley & Mihalcea จึงได้นำเสนอวิธีการเปรียบเทียบเชิงความหมายแบบ Knowledge-Based Similarity โดยได้ประยุกต์ใช้ WordNet ซึ่งเป็นคลังคำศัพท์ออนไลน์เข้ามาประยุกต์ใช้ในการดำเนินการเปรียบเทียบเชิงความหมายระหว่างคำ วิธีการดำเนินงานเริ่มต้นจากการแบ่งคำแต่ละคำในประโยคออกจากกัน จากนั้นทำการระบุถึงหน้าที่ของคำแต่ละคำสุดท้ายทำการจับคู่ของคำที่จะทำการเปรียบเทียบโดยขึ้นอยู่กับหน้าที่ของคำ อาทิเช่น คู่ของคำนาม, คู่ของคำกิริยา เป็นต้น เมื่อจับคู่ของคำได้แล้วจะนำคู่ของคำแต่ละคู่มาทำการเปรียบเทียบเชิงความหมายเพื่อหาความเหมือนกันของแต่ละคู่ ที่ซึ่งในการดำเนินการหาค่าความเหมือนกัน Corley & Mihalcea ได้ประยุกต์ใช้วิธีการคำนวณของ Wu & Palmer ซึ่งเป็นการคำนวณหาค่าความเหมือนกันของคำจากวิธีการเปรียบเทียบเชิงความหมาย มาร่วมดำเนินการกับวิธีการ Inverse Document Frequency (IDF) เพื่อนำมาใช้ในการหาค่าความเหมือนของแต่ละคู่ของคำที่นำมาเปรียบเทียบกัน จากการดำเนินการทดลอง Courtney ได้นำวิธีการเปรียบเทียบเชิงความเหมือนที่นำเสนอมาทำการประเมินผลเทียบกับวิธีการคำนวณหาค่าความเหมือนกันของคำของวิธีการเปรียบเทียบเชิงความหมายทั้ง 6 วิธี ซึ่งได้แก่ Leacock & Chodorow, Lesk, Wu & Palmer, Resnik, Lin และ Jiang & Conrath นอกจากนี้ยังทำการประเมินผลเทียบกับวิธีการคำนวณหาค่าความเหมือนกันของวิธีการ

เปรียบเทียบแบบสายอักขระและการคำนวณหาค่าความเหมือนกันโดยวิธีการของ cosine similarity ที่ซึ่งผลลัพธ์จากการทดลองแสดงให้เห็นถึงประสิทธิภาพของความถูกต้องของการเปรียบเทียบเชิงความหมายจากวิธีการที่นำเสนอ ที่สามารถดำเนินการได้เหนือกว่าทั้ง 8 วิธีการที่ได้กล่าวมาข้างต้น

(Devlin, Chang, Lee, & Toutanova, 2018) นำเสนอขั้นตอนวิธีเปรียบเทียบเชิงความหมายที่ชื่อว่า “BERT (Bidirectional Encoder Representations from Transformers)” โดยการประยุกต์ใช้คลังคำศัพท์ของ BooksCorpus ซึ่งประกอบด้วยคำศัพท์ 800 ล้านคำ และคลังคำศัพท์ของ English Wikipedia ซึ่งประกอบด้วยคำศัพท์ 2,500 ล้านคำ เข้ามาร่วมดำเนินการวิเคราะห์และเปรียบเทียบเชิงความหมายระหว่างคำ จากนั้นทำการประยุกใช้ WordPiece embeddings เข้ามาเพื่อทำการคำนวณหาค่าของความเหมือนกันในเชิงควาหมายระหว่างคำ ซึ่งจากการดำเนินการเปรียบเทียบเชิงความหมายระหว่างคำหรือข้อความจากขั้นตอนวิธีการของ BERT ทำให้ได้ผลลัพธ์ที่มีประสิทธิภาพที่ดีกว่าเมื่อนำไปเปรียบเทียบกับผลลัพธ์จากการเปรียบเทียบเชิงความหมายระหว่างคำหรือข้อความจากวิธีการของ GLUE, MultiNLI, SQuAD v1.1 และ SQuAD v2.0

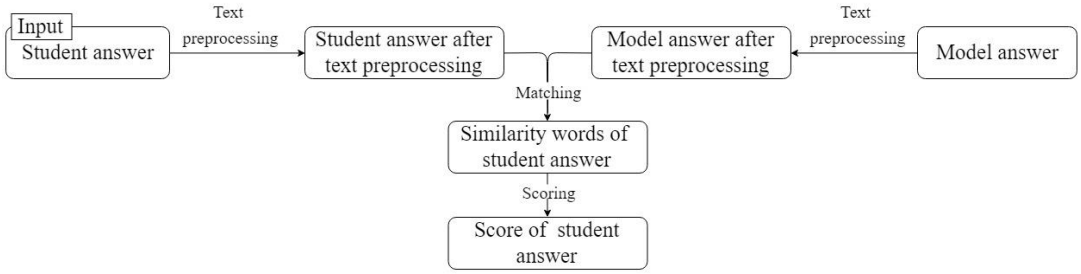
(Cross, Mokrenko, Crockett, & Adel, 2020) นำเสนอขั้นตอนวิธีการเปรียบเทียบระหว่างข้อความแบบเชิงความหมายโดยมีการนำการคำนวณความเหมือนกันแบบ fuzzy โดยมีวิธีการคำนวณได้แก่ 1) Sup-Min, 2) Jaccard similarity, 3) Geometric Fuzzy Similarity และ 4) Similarity on Type-2 เข้ามาร่วมดำเนินการเปรียบเทียบเทียบด้วย ขั้นตอนวิธีที่นำเสนอจะมีด้วยกันทั้งสิ้น 3 ขั้นตอนวิธี คือ 1) STASIS เป็นวิธีการเปรียบเทียบเชิงความหมายระหว่างคำโดยการประยุกต์ใช้ออนโทโลยีที่สร้างโดย WordNet และคลังคำศัพท์ของ Brown Corpus, 2) FAST เป็นขั้นตอนวิธีที่พัฒนาขึ้นจาก STASIS โดยทำการจัดหมวดหมู่ของข้อมูลที่จะนำมาเปรียบเทียบ และ 3) FUSE ในขั้นตอนวิธีนี้จะมีการนำผู้เชี่ยวชาญเข้ามาทำการพัฒนาออนโทโลยีที่ใช้ดำเนินงานอยู่ จากการที่นำผู้เชี่ยวชาญเข้ามาพัฒนาออนโทโลยีทำให้สามารถเพิ่มจำนวนคำศัพท์ในออนโทโลยีที่ใช้อยู่ถึง 57% โดยผลลัพธ์จากการดำเนินการทดลองจะแสดงให้เห็นถึงผสมผสานระหว่างขั้นตอนวิธีทั้ง 3 ขั้นตอนร่วมกับวิธีการคำนวณแบบ fuzzy ทั้ง 4 ขั้นตอน

2.2.2.4 การเปรียบเทียบแบบผสมผสาน (Hybrid Similarity Measures)

เป็นการรวมวิธีการต่าง ๆ ในการคำนวณหาความเหมือนกันของคำเข้าไว้ด้วยกัน ซึ่งจะทำให้ประสิทธิภาพในการดำเนินการเปรียบเทียบทำได้ดีกว่าในวิธีการเปรียบเทียบแบบเดี่ยว ๆ ซึ่งวิธีการเปรียบเทียบแบบผสมผสานมีงานวิจัยที่เกี่ยวข้องดังต่อไปนี้

(Mihalcea, Corley, & Strapparava, 2006) ได้นำเสนอการเปรียบเทียบข้อความในด้านของการเปรียบเทียบความหมาย ที่ซึ่งใช้วิธีการ Corpus-Based และ Knowledge-based Similarity เข้ามาเพื่อทำการเปรียบเทียบหาความเหมือนกันระหว่างข้อความจากความหมาย โดยการนำข้อมูลข่าวสารจากเว็บไซต์มาทำการเปรียบเทียบกัน ซึ่งก่อนการจะนำข้อความที่ได้มาทำการเปรียบเทียบกันเพื่อหาความหมายนั้นจะต้องได้รับการระบุจากผู้เชี่ยวชาญว่าในระหว่างข้อความ 2 ข้อความมีคำใดบ้างที่ควรนำมาเปรียบเทียบกัน จากนั้นจะนำคำที่ระบุได้ระหว่าง 2 ข้อความมาทำการเปรียบเทียบกันเชิงความหมาย ซึ่งจากการทดลองแสดงให้เห็นว่าการเปรียบเทียบกันเชิงความหมายในด้านการเปรียบเทียบแบบ Corpus-Based Similarity ดำเนินการได้ดีกว่าในด้านของความถูกต้อง (recall) และการเปรียบเทียบกันเชิงความหมายในด้านการเปรียบเทียบแบบ Knowledge-Based Similarity สามารถดำเนินการได้ดีกว่าในด้านของความแม่นยำ (precision) นอกจากนี้แล้ว Mihalcea et al., ยังได้ทำการรวบทั้ง 2 วิธีการเข้าด้วยกันซึ่งทำให้เห็นว่าสามารถดำเนินการได้ดีกว่าการเปรียบเทียบข้อความแบบ ต่าง ๆ

(Gomaa & Fahmy, 2012) ได้นำเสนอระบบการให้คะแนนแบบอัตโนมัติจากการดำเนินการเปรียบเทียบระหว่างคำตอบของนักเรียนและเฉลย โดยคำตอบที่สามารถนำมาดำเนินการเปรียบเทียบจะต้องเป็นคำตอบแบบข้อความที่มีขนาดสั้น เพื่อที่จะสามารถนำไปเปรียบเทียบหาความเหมือนกันระหว่างคำตอบและเฉลยได้ ซึ่งมีขั้นตอนการทำงานดังภาพที่ 3



ภาพที่ 3 ขั้นตอนการทำงานของระบบการให้คะแนนแบบอัตโนมัติ

โดยการดำเนินงานจะเริ่มต้นที่การประยุกต์ใช้เทคนิคการประมวลผลภาษาธรรมชาติ ได้แก่ การกำจัดคำหยุดและการเปลี่ยนรูปของคำให้อยู่ในรากศัพท์ เพื่อนำมาใช้ในการประมวลผลข้อความเบื้องต้นของคำตอบที่ได้จากนักเรียนและเฉลย จากนั้นทำการประยุกต์วิธีการเปรียบเทียบข้อความแบบ String-based similarity และ Corpus-based similarity เข้ามาช่วยในการดำเนินการเปรียบเทียบระหว่างคำตอบและเฉลย ซึ่งในส่วนของ String-Based Similarity ถูกประยุกต์ใช้เพื่อนำมาเปรียบเทียบหาความเหมือนของสายอักขระและคำในคำตอบและเฉลย โดยได้ประยุกต์ใช้คำนวณเพื่อหาค่าความเหมือนแบบต่าง ๆ เช่น Jaro algorithm, Jaro–Winkler distance, N-gram, Needleman-Wunsch, Cosine similarity, Dice's coefficient, Euclidean distance, Jaccard similarity, Matching Coefficient, Overlap coefficient เป็นต้น และในส่วนของ Corpus-based similarity ได้ถูกประยุกต์ใช้เพื่อนำมาหาความเหมือนกันในระดับของความหมายระหว่างคำตอบและเฉลย ที่ซึ่งได้ทำการประยุกต์ใช้วิธีการคำนวณหาค่าความเหมือนกันของวิธีการ DICSO1 และ DISCO2 ซึ่งเป็นวิธีการคำนวณหาค่าความเหมือนกันเชิงความเหมือนในกลุ่มของ Corpus-based similarity เมื่อทำการเปรียบเทียบความเหมือนกันของคำตอบและเฉลยเสร็จแล้ว จากนั้นจะนำมาทำการคำนวณหาคะแนนของคำตอบโดยการประยุกต์ใช้วิธีการหาค่าสัมประสิทธิ์สหสัมพันธ์ตามวิธีของเพียร์สัน (Pearson's Correlation Coefficient) ที่ซึ่งจากการดำเนินการทดลองแสดงให้เห็นว่าการนำวิธีการเปรียบเทียบแบบ String-based similarity และ Corpus-based similarity เข้ามาใช้ดำเนินการร่วมกันสามารถดำเนินการได้ดีกว่าการใช้วิธีการเปรียบเทียบแบบ String-based similarity หรือ Corpus-based similarity แบบเดี่ยว ๆ อาทิเช่น การประยุกต์ใช้วิธีการเปรียบเทียบแบบ N-gram ที่อยู่ในกลุ่มของ String-based similarity ร่วมกับวิธีการเปรียบเทียบแบบ DICSO1 ที่อยู่ในกลุ่มของ String-based similarity มาทำการคำนวณหาค่าความเหมือนกันจากการเปรียบเทียบระหว่างคำตอบและเฉลย ที่ซึ่งจากการนำทั้ง 2 วิธีมาดำเนินการร่วมกันทำให้ได้มาซึ่งค่าของผลลัพธ์จากการดำเนินการเปรียบเทียบที่ดีที่สุด

(Mumtaz & Giese, 2020) นำเสนอวิธีการเปรียบเทียบข้อความแบบผสมผสาน โดยการนำวิธีการเปรียบเทียบข้อความ 2 วิธีการมาดำเนินงานร่วมกัน ได้แก่ 1) วิธีการเปรียบเทียบข้อความโดยการนับความถี่ของคำที่เหมือนกันระหว่าง 2 บทความ และ 2) การเปรียบเทียบคำระหว่าง 2 บทความแบบเชิงความหมาย โดยในการเปรียบเทียบแบบเชิงความหมาย Mumtaz et al., ได้ทำการสร้างออนโทโลยีของข้อมูลเพื่อแสดงให้เห็นถึงความสัมพันธ์ของข้อมูล จากนั้นนำออนโทโลยีที่สร้างขึ้นเข้ามาช่วยดำเนินการเปรียบเทียบเชิงความหมาย ซึ่งผลลัพธ์ที่ได้จากการเปรียบเทียบด้วยวิธีการของ

Mumtaz et al., สามารถให้ผลลัพธ์ที่มีความแม่นยำสูงสุด เมื่อนำไปเทียบกับผลลัพธ์ที่ได้จากวิธีการเปรียบเทียบข้อความด้วยขั้นตอนวิธีของ ICS, HS, OF, CMS และ Eskin

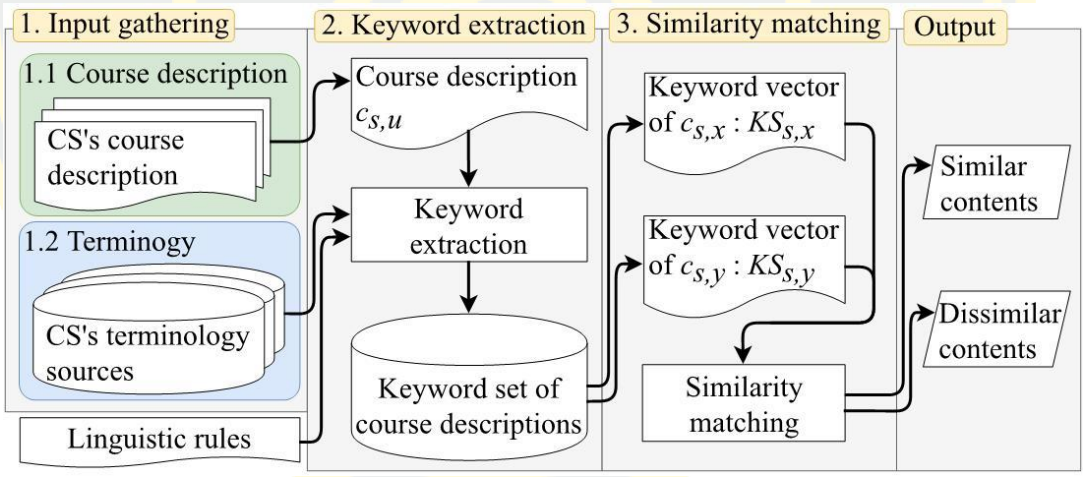
(Lukyamuzi, Ngubiri, & Okori, 2020) ได้นำเสนอวิธีการจัดกลุ่มเอกสารที่เกี่ยวข้องกับความไม่มั่นคงทางอาหาร (Food Insecurity) โดยทำการรวบรวมข้อมูลของเอกสารที่ต้องการจาก www.monitor.co.ug จากนั้นประยุกต์ใช้วิธีการประมวลผลข้อความเบื้องต้นเข้ามาทำการประมวลผลข้อความในเอกสารที่รวบรวมมาได้ โดยเริ่มจากการแบ่งคำ, การกำจัดคำหยุด, การลดรูปของคำให้อยู่ในรากศัพท์ และ การกำหนดเวคเตอร์ของคำ ต่อมาทำการคำนวณความเหมือนกันของเอกสารเพื่อจัดกลุ่มเอกสารที่อยู่ในหมวดหมู่เดียวกัน โดยในการคำนวณความเหมือนกัน Lukyamuzi et al., ได้ประยุกต์ใช้วิธีการหาค่าความเหมือนกันของ “Cosine similarity” และการนับความถี่ของคำที่เหมือนกันในเอกสารโดยวิธีการของ TF-IDF ในการดำเนินการทดสอบในด้านของประสิทธิภาพการทำงาน Lukyamuzi et al., ได้ทำการทดสอบกับการจัดกลุ่มเอกสารด้วยวิธีการของ NAÏVE BAYES และ KNN พบว่าประสิทธิภาพการทำงานของขั้นตอนวิธีที่ได้ออกแบบไว้สามารถดำเนินงานได้ดีกว่าวิธีการจัดกลุ่มทั้ง 2 วิธีการ

(Farouk, 2020) นำเสนอวิธีการเปรียบเทียบความเหมือนกันระหว่างประโยค โดยการผสมผสานวิธีการ คือ 1) การเปรียบเทียบความเหมือนกันจากโครงสร้างของประโยค โดยการนำประโยคแต่ละประโยคมาวิเคราะห์หาความสัมพันธ์ระหว่างคำที่ปรากฏในประโยค จากนั้นนำคำและความสัมพันธ์ที่ได้มาสร้างเป็นกราฟความสัมพันธ์ของประโยคนั้น ๆ ต่อมานำกราฟของทั้ง 2 ประโยคที่นำมาเปรียบเทียบกันมาคำนวณหาความเหมือนกันโดยการสร้างเมทริกซ์ของความสัมพันธ์จากทั้ง 2 ประโยค เพื่อแสดงให้เห็นถึงความเหมือนกันของความสัมพันธ์ในทั้ง 2 ประโยค จากนั้นนำคำนวณหาค่าความเหมือนกันระหว่างประโยคโดยใช้การคำนวณแบบ Cosine similarity, 2) วิธีการเปรียบเทียบความเหมือนกันจากเวคเตอร์ของคำ ซึ่งในการสร้างเวคเตอร์ของคำ Mamdouh Farouk ได้ทำการประยุกต์ใช้ Google’s pre-trained word embedding vectors เข้ามาสร้างเวคเตอร์ของแต่ละคำในประโยค จากนั้นนำทั้ง 2 ประโยคที่ผ่านการสร้างเวคเตอร์แล้วมาคำนวณหาค่าความเหมือนกันจากการคำนวณแบบ Cosine similarity และ 3) วิธีการเปรียบเทียบความเหมือนกันจากลำดับการเกิดขึ้นของคำ เนื่องจากในบางประโยคที่นำมาเปรียบเทียบกันอาจมีรายการของคำที่เหมือนกันแต่ก็อาจให้ความหมายของประโยคที่แตกต่างกันได้ ดังนั้น Mamdouh Farouk จึงทำการนำลำดับการเกิดขึ้นของคำมาคำนวณเพื่อให้ได้ผลลัพธ์ของการเปรียบเทียบที่มีประสิทธิภาพมากยิ่งขึ้น

จากงานวิจัยที่เกี่ยวข้องกับการวิเคราะห์คำอธิบายรายวิชาที่ได้ทำการศึกษามาข้างต้น ส่วนใหญ่แล้วจะมุ่งเน้นที่การถึงวิเคราะห์วัตถุประสงค์ที่ได้รับจากการเรียนรู้ในชั้นเรียนร่วมกับข้อมูลเชิงลึกข้อผู้เรียนแต่ละคน เพื่อทำการวัดประสิทธิภาพองค์ความรู้ของผู้เรียนแต่ละคนที่ได้รับจากการถ่ายทอดในชั้นเรียนนั้น ๆ, เพื่อการแนะนำคอร์สเรียนที่เหมาะสมกับผู้เรียนแต่ละคน หรือการวิเคราะห์คำอธิบายรายวิชาเพื่อให้ทราบถึงความต้องการขององค์ความรู้ในสายงานหนึ่ง ๆ ซึ่งแนวคิดทั้งหมดก็ยังไม่ได้มุ่งเน้นที่การวิเคราะห์คำอธิบายรายวิชาเพื่อหาความเหมือนและความแตกต่างระหว่างคำอธิบายรายวิชาในรายวิชาหนึ่ง ๆ ด้วยเหตุนี้ ผู้วิจัยจึงทำการสร้างระบบสำหรับการวิเคราะห์และเปรียบเทียบคำอธิบายรายวิชา เพื่อแสดงให้เห็นถึงความเหมือนและความแตกต่าง ของเนื้อหาในคำอธิบายรายวิชาในรายวิชาหนึ่ง ๆ ที่ซึ่งจะเป็นส่วนช่วยให้คณะกรรมการและ/หรืออาจารย์ผู้สอนในรายวิชานั้น ๆ สามารถนำข้อมูลที่ได้ ไปปรับใช้ในการแก้ไขและปรับปรุงคำอธิบายรายวิชาให้มีเนื้อหาที่เหมาะสมมากยิ่งขึ้น และจากการศึกษางานวิจัยที่เกี่ยวข้องกับการเปรียบเทียบความเหมือนกันของข้อความที่ได้กล่าวมาข้างต้น สามารถแบ่งวิธีการเปรียบเทียบความเหมือนกันของข้อความได้ 4 วิธีได้แก่ 1) การเปรียบเทียบแบบสายอักขระ 2) การเปรียบเทียบเชิงความหมายจากแหล่งข้อมูลที่สร้างขึ้นเอง 3) การเปรียบเทียบเชิงความหมายจากแหล่งข้อมูลออนไลน์ และ 4) การเปรียบเทียบแบบผสมผสาน ทำให้ผู้วิจัยได้ทำการประยุกต์ใช้ วิธีการเปรียบเทียบแบบสายอักขระ ทั้งในส่วนของการเปรียบเทียบความเหมือนกันของสายอักขระและคำ, วิธีการเปรียบเทียบเชิงความหมายจากแหล่งข้อมูลที่สร้างขึ้นเอง และ วิธีการเปรียบเทียบเชิงความหมายจากแหล่งข้อมูลออนไลน์ เข้ามาใช้ในการดำเนินการเปรียบเทียบระหว่างคำอธิบายรายวิชา โดยผลลัพธ์ที่ได้จากการเปรียบเทียบระหว่างคำอธิบายรายวิชาในรายวิชาหนึ่ง ๆ จะแสดงให้เห็นถึงส่วนของเนื้อหาในคำอธิบายรายวิชาตั้งต้นที่เหมือนและแตกต่างกันกับคำอธิบายรายวิชาเปรียบเทียบ และแสดงเป็นข้อมูลเชิงสรุปที่จะบ่งบอกถึงอัตราร้อยละของความเหมือนและความแตกต่างจากการเปรียบเทียบ

บทที่ 3
ระบบวิเคราะห์คำอธิบายรายวิชาในหลักสูตรวิทยาการคอมพิวเตอร์

บทนี้จะอธิบายถึงการดำเนินงานของ “ระบบวิเคราะห์คำอธิบายรายวิชาในหลักสูตรวิทยาการคอมพิวเตอร์ (*Computer Science Course Description Analysis system*)” หรือเรียกว่า “ระบบ *CSCDA*” ซึ่งเป็นระบบสำหรับวิเคราะห์และเปรียบเทียบเนื้อหาจาก 2 (หรือกลุ่มของ) คำอธิบายรายวิชาของรายวิชาหนึ่ง ๆ ในศาสตร์ด้านวิทยาการคอมพิวเตอร์ โดยคำอธิบายรายวิชาที่นำมาวิเคราะห์และเปรียบเทียบกันจะต้องเป็นคำอธิบายของรายวิชาที่เหมือนหรือสอดคล้องกันเท่านั้น การดำเนินงานของระบบ *CSCDA* ดังแสดงในภาพที่ 4 ซึ่งประกอบด้วย 3 ขั้นตอน ได้แก่ 1) การรวบรวมข้อมูลนำเข้า (Input gathering) 2) การสกัดคำสำคัญจากคำอธิบายรายวิชา (Keyword extraction) และ 3) การเปรียบเทียบคำสำคัญระหว่างคำอธิบายรายวิชา (Similarity Matching) ตามลำดับ



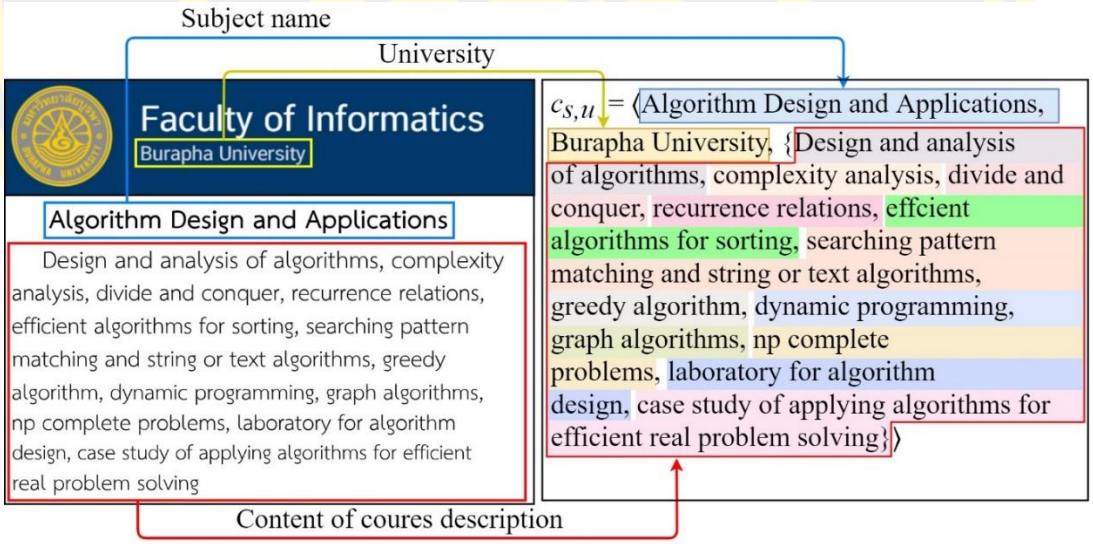
ภาพที่ 4 โครงสร้างของระบบ CSCDA

3.1 การรวบรวมข้อมูลนำเข้า

จะเป็นการรวบรวมข้อมูล 3 ส่วน คือ 1) คำอธิบายรายวิชา 2) คำศัพท์เฉพาะทางด้านวิทยาการคอมพิวเตอร์ และ 3) กฎทางภาษาศาสตร์ โดยแต่ละส่วนมีรายละเอียดดังนี้

3.1.1 การรวบรวมคำอธิบายรายวิชา

ในขั้นตอนแรกของระบบ *CSCDA* จะเป็นการรวบรวมคำอธิบายรายวิชาของรายวิชาในหลักสูตรวิทยาการคอมพิวเตอร์ จากมหาวิทยาลัยที่เปิดสอนหลักสูตรวิทยาการคอมพิวเตอร์ โดยคำอธิบายรายวิชาที่จะสามารถรวบรวมได้ จะเป็นคำอธิบายรายวิชาที่สามารถสืบค้นได้จากเว็บไซต์ของภาควิชานั้น ๆ ซึ่งในการรวบรวมข้อมูลคำอธิบายรายวิชาหนึ่ง ๆ จะดำเนินการรวบรวมข้อมูล 3 ส่วน แล้วจัดเก็บอยู่ในรูปแบบ 3-tuple : $< s, u, cc >$ เมื่อ s หมายถึง ชื่อรายวิชา (ภาษาอังกฤษ) u หมายถึง ชื่อมหาวิทยาลัย (ภาษาอังกฤษ) และ cc หมายถึง เนื้อหาของคำอธิบายรายวิชา (ภาษาอังกฤษ) ตามลำดับ จากนั้นทำการแบ่งเนื้อหาในคำอธิบาย (กล่าวคือ cc) ให้ออกเป็นหัวข้อย่อย ๆ และจัดเก็บหัวข้อย่อยเหล่านั้นในรูปแบบลิสต์ : $TP = < tp_1, tp_2, \dots, tp_n >$ โดยการการแบ่งเนื้อหาในคำอธิบายออกเป็นหัวข้อย่อย จะทำให้โครงสร้างการจัดเก็บข้อมูลคำอธิบายรายวิชาหนึ่งๆ เปลี่ยนจาก $< s, u, cc >$ เป็น $< s, u, TP = < tp_1, tp_2, \dots, tp_n >>$ หลังจากเก็บรวบรวมคำอธิบายรายวิชาจากมหาวิทยาลัยต่าง ๆ แล้ว จะได้คลังข้อมูลของคำอธิบายรายวิชา ที่เรียกว่า *CS-CDC* (CS course description corpus) ดังแสดงตัวอย่างในภาพ ที่ 5



ภาพที่ 5 ตัวอย่างการรวบรวมคำอธิบายรายวิชาของระบบ CSCDA

จากภาพที่ 5 จะแสดงให้เห็นถึงการรวบรวมคำอธิบายรายวิชาของวิชา “Algorithm Design and Applications” จากมหาวิทยาลัยบูรพา เมื่อทำการรวบรวมข้อมูลคำอธิบายรายวิชาทั้ง 3 ส่วนแล้ว จากนั้นทำการแบ่งหัวข้อในเนื้อหาคำอธิบายรายวิชา ออกเป็น 12 หัวข้อย่อย

3.1.2 การรวบรวมคำศัพท์เฉพาะทางด้านวิทยาการคอมพิวเตอร์

เนื่องจากเนื้อหาในคำอธิบายรายวิชาทางด้านวิทยาการคอมพิวเตอร์มักประกอบด้วยคำศัพท์เฉพาะ อาทิเช่น ‘data science’ ‘big data’ ‘database’ ‘data mining’ ‘data warehouse’ เป็นต้น จึงเป็นเหตุให้ต้องมีการเก็บรวบรวมข้อมูลคำศัพท์เฉพาะเพื่อนำมาใช้สำหรับการวิเคราะห์เนื้อหาในรายวิชาหนึ่ง ๆ ซึ่งสามารถรวบรวมได้โดยประยุกต์ใช้วิธีการสกัดข้อความบนเว็บไซต์ (Web Scraping) เพื่อรวบรวมคำศัพท์เฉพาะจากคลังคำศัพท์เฉพาะ 8 คลังคำศัพท์ ได้แก่ 1) A Dictionary of Computer Science[2], 2) Computer dictionary[3], 3) Computer Science Glossary[4], 4) Glossary of Computer Related Terms[5], 5) Glossary of computer science[6], 6) Labautopedia[7], 7) PC Glossary[8], และ 8) Techtarget[9] ตามลำดับ จากการรวบรวมคำศัพท์เฉพาะทำให้ได้มาซึ่งคำศัพท์เฉพาะจำนวนทั้งสิ้น 28,392 คำ และเก็บไว้ในคลังคำศัพท์เฉพาะที่เรียกว่า “CS’s terminology corpus”

3.1.3 การรวบรวมกฎทางภาษาศาสตร์

ส่วนสุดท้ายของขั้นตอนการรวบรวมข้อมูลนำเข้าจะเป็นการรวบรวมกฎทางภาษาศาสตร์ที่ถูกสร้างขึ้นโดยระบบ *SBS* (*Supplementary Book Suggestion system*) (Chaisoongnoen, Amphawan, & Bunpeng, 2018) ซึ่งเป็นกฎทางภาษาศาสตร์ที่ถูกนำมาใช้สำหรับการสกัดคำสำคัญจากเนื้อหาในคำอธิบายรายวิชาหนึ่ง ๆ โดยสามารถจำแนกประเภทของกฎได้เป็น 2 ประเภท คือ

1) กฎทางภาษาศาสตร์ที่พิจารณาร่วมกับคำศัพท์เฉพาะ กฎนี้จะทำการพิจารณาถึงคำศัพท์เฉพาะที่ปรากฏขึ้นในหัวข้อย่อยหนึ่ง ๆ เป็นลำดับแรก ต่อมาจะทำการพิจารณาในส่วนของคำต่าง ๆ (ที่อยู่ในหัวข้อย่อยเดียวกัน) ที่ไม่ได้เป็นคำศัพท์เฉพาะ อาทิเช่น คำนาม (Noun)

---

[2] http://www.oxfordreference.com/view/10.1093/acref/\\9780199688975.001.0001/acref-9780199688975

[3] https://www.computerhope.com/jargon/ja.htm

[4] https://www.computerscience.gcse.guru/glossary

[5] http://www.math.utah.edu/~wisnia/glossary.html

[6] https://en.wikipedia.org/wiki/Glossary_of_computer_science

[7] http://www.labautopedia.org/mw/List_of_programming_and_computer_science_terms

[8] https://pc.net/glossary/

[9] https://whatis.techtarget.com/definitions/A

คำบุพบท (Preposition) หรือ คำคุณศัพท์ (Adjective) เป็นต้น ตัวอย่างเช่น กฎ “ Adjective (JJ) + Terminology (TE1) + Preposition (IN) + Terminology (TE2)” ⟶ ⟨‘Terminology (TE2) + Adjective (JJ)’, ‘Adjective (JJ) + Terminology (TE1)’, ‘Terminology (TE2) + Terminology (TE1)’⟩ ซึ่งเป็นกฎที่ประกอบด้วย 4 ส่วน คือ 1. คำคุณศัพท์ (Adjective) 2. คำศัพท์เฉพาะ (Terminology) 3. คำบุพบท (Preposition) และ 4. คำศัพท์เฉพาะ (Terminology) ตามลำดับ โดยที่คำทั้ง 4 ประเภทจะต้องปรากฏร่วมกันในหัวข้อย่อยเดียวกันแบบเรียงลำดับ จากการใช้กฎข้างต้นในการสกัดคำสำคัญจะทำให้ได้มาซึ่งคำสำคัญทั้งหมด 3 คำ ได้แก่ i) คำสำคัญที่ประกอบด้วย ‘คำศัพท์เฉพาะ (TE2) + คำคุณศัพท์ (JJ)’ ii) คำสำคัญที่ประกอบด้วย ‘คำคุณศัพท์ (JJ) + คำศัพท์เฉพาะ (TE1)’ และ iii) คำสำคัญที่ประกอบด้วย ‘คำศัพท์เฉพาะ (TE2) + คำศัพท์เฉพาะ (TE1)’

2) กฎทางภาษาศาสตร์ที่ไม่ได้พิจารณาร่วมกับคำศัพท์เฉพาะ กฎประเภทนี้จะถูกนำมาสกัดคำสำคัญสำหรับหัวข้อย่อยที่ไม่มีคำศัพท์เฉพาะปรากฏขึ้น โดยจะทำการพิจารณาถึงกลุ่มของคำนาม (Noun phrase) กล่าวคือ คำนาม (Noun) ที่ปรากฏขึ้นร่วมกันกับคำนาม, คำคุณศัพท์ (Adjective) หรือ คำกริยา (Verb) ในประโยค แล้วทำการระบุคำสำคัญ ตัวอย่างเช่น กฎ : “ Adjective (JJ) + Noun (NN)” ⟶ ⟨‘Adjective (JJ) + Noun (NN)’⟩ เป็นกฎที่ประกอบด้วยคำที่มีหน้าที่ของคำ 2 แบบ คือ 1. คำคุณศัพท์ (Adjective) และ 2. คำนาม (Noun) โดยคำทั้ง 2 จะต้องปรากฏร่วมกันในหัวข้อย่อยเดียวกันแบบเรียงลำดับ จากการใช้กฎข้างต้นในการสกัดคำสำคัญจะทำให้ได้มาซึ่งคำสำคัญทั้งหมด 1 คำ ได้แก่ i) คำสำคัญที่ประกอบด้วย ‘คำคุณศัพท์ (JJ) + คำนาม (NN)’

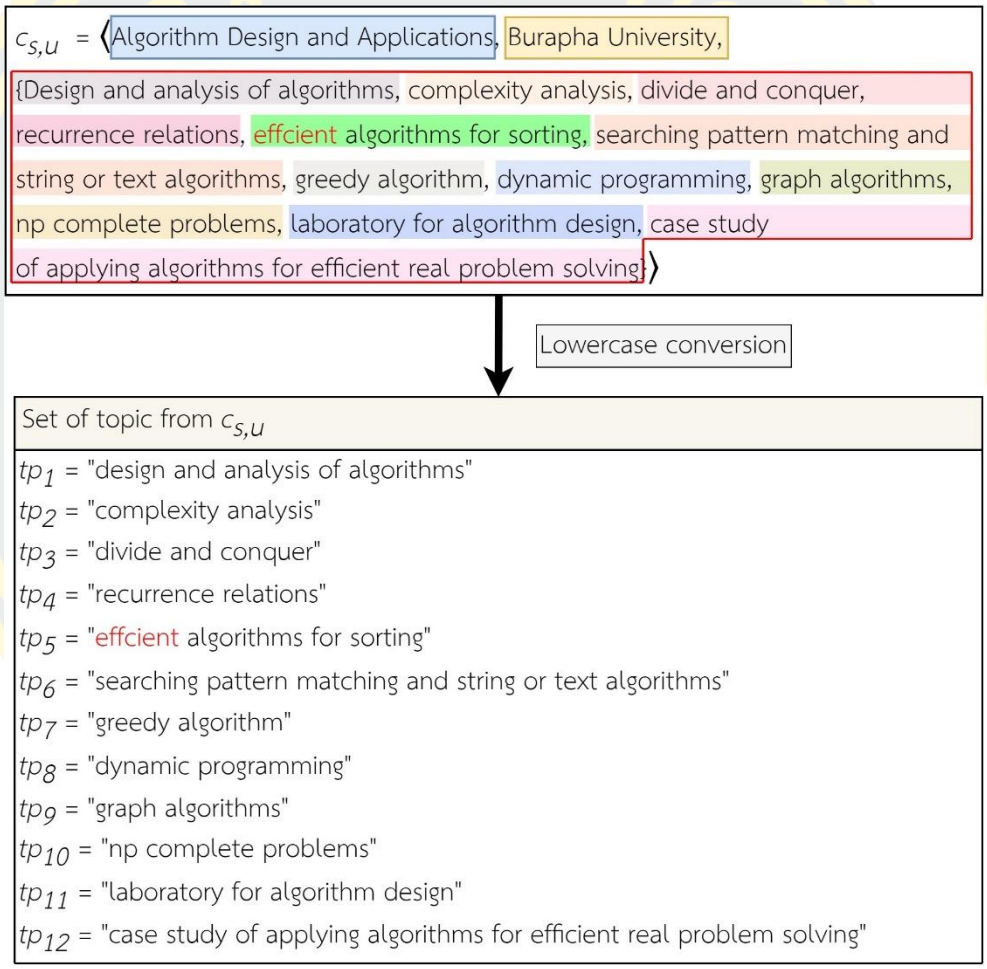
3.2 การสกัดคำสำคัญจากคำอธิบายรายวิชา

หลังจากทำการรวบรวมคำอธิบายรายวิชาจากมหาวิทยาลัยต่าง ๆ และทำการแบ่งเนื้อหาในคำอธิบายให้ออกเป็นหัวข้อย่อย ๆ แล้ว จากนั้นจะทำการสกัดเนื้อหาที่สำคัญในคำอธิบายรายวิชาด้วย “การสกัดคำสำคัญ” ซึ่งจะเริ่มจากการประยุกต์ใช้เทคนิคการประมวลผลข้อความเบื้องต้นกับแต่ละหัวข้อย่อยของคำอธิบายรายวิชาดังนี้

3.2.1 การประมวลผลข้อความเบื้องต้น

3.2.1.1 การแปลงตัวอักษรพิมพ์ใหญ่ให้เป็นตัวพิมพ์เล็ก (Lowercase conversion)

การแปลงตัวอักษรพิมพ์ใหญ่ให้เป็นตัวพิมพ์เล็กในภาษาอังกฤษเป็นเทคนิคการตรวจสอบถึงตัวอักษรที่ตัวตัวพิมพ์ใหญ่ในภาษาอังกฤษ เมื่อตรวจสอบพบจะทำการแปลงตัวอักษรตัวนั้น ๆ ให้กลายเป็นตัวอักษรตัวเดียวกันที่เป็นตัวพิมพ์เล็ก โดยมีตัวอย่างการแปลงตัวอักษรพิมพ์ใหญ่ให้เป็นตัวพิมพ์เล็ก ดังภาพที่ 6



ภาพที่ 6 ตัวอย่างการแปลงตัวอักษรพิมพ์ใหญ่ให้เป็นตัวพิมพ์เล็กในแต่ละหัวข้อย่อยของระบบ CSCDA

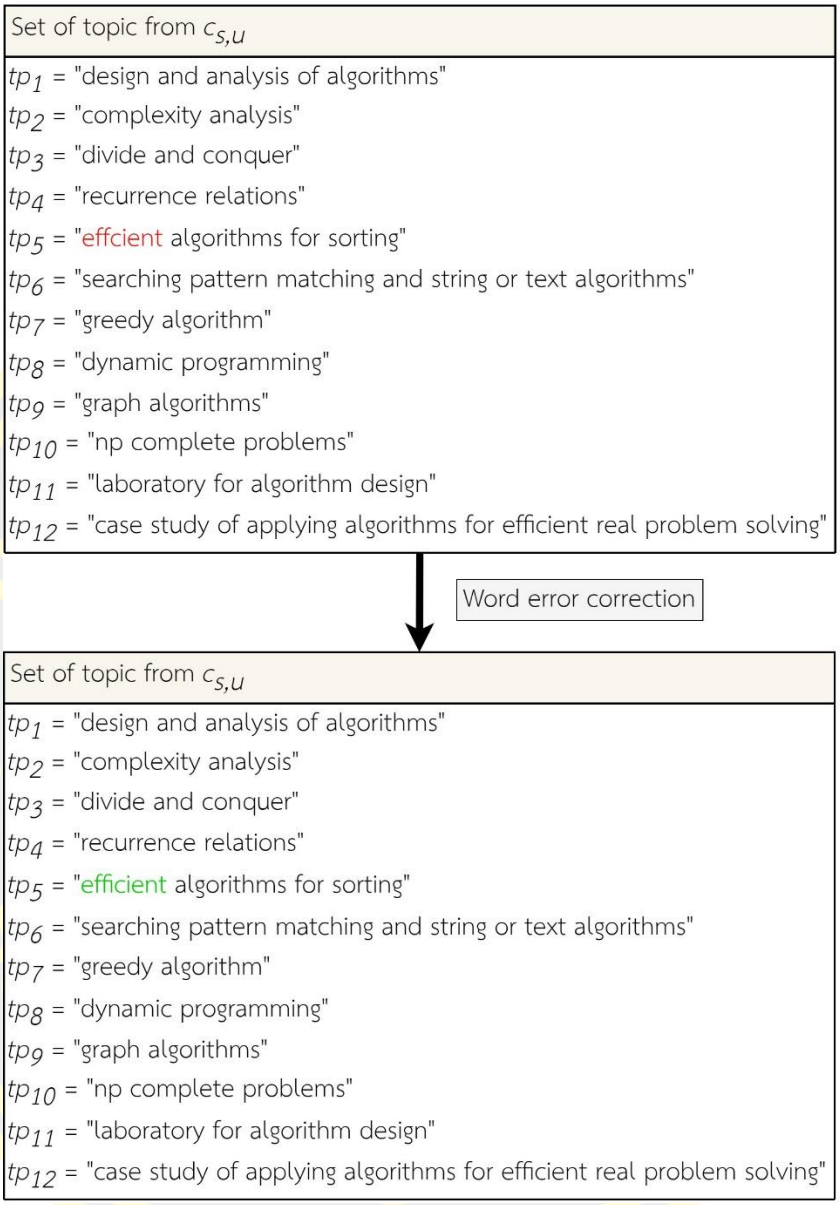
จากภาพที่ 6 เป็นตัวอย่างการแปลงตัวอักษรพิมพ์ใหญ่ให้เป็นตัวพิมพ์เล็กเนื้อหาในแต่ละหัวข้อย่อยของคำอธิบายรายวิชา โดยเมื่อพิจารณาจะพบว่าหัวข้อย่อยที่ 1 คือ “Design and analysis of algorithms” เมื่อตรวจสอบจะพบว่ามีตัวอักษรพิมพ์ใหญ่ปรากฏอยู่ ได้แก่ ตัว “D” จึง

ทำการแปลงตัวอักษรตัวนี้ให้กลายเป็นตัวอักษรตัวเดียวกันที่เป็นตัวพิมพ์เล็ก ได้แก่ ตัว “d” ดังนั้น ผลลัพธ์ที่ได้จากการแปลงตัวอักษรพิมพ์ใหญ่ให้เป็นตัวพิมพ์เล็กในหัวข้อย่อยที่ 1 คือ “design and analysis of algorithms” ในทางกลับกัน เมื่อทำการพิจารณาหัวข้อย่อยอื่น ๆ จะไม่พบว่ามีตัวอักษรพิมพ์ใหญ่ปรากฏอยู่เลย ทำให้เนื้อหาของหัวย่อยที่ 2 - 12 ยังคงเดิม

3.2.1.2 การแก้ไขคำผิด (Word error correction)

คำอธิบายรายวิชาหนึ่ง ๆ อาจมีคำศัพท์ที่สะกดผิดเกิดขึ้น จึงจำเป็นจะต้องได้รับการแก้ไข ซึ่งสามารถประยุกต์ใช้วิธีการแก้ไขคำผิด (Spelling Mistake Correction (SMC)) (Gupta, 2015) เพื่อตรวจสอบและแก้ไขให้แต่ละคำในคำอธิบายรายวิชามีความถูกต้องของการสะกดคำ โดยในการประมวลผลจะทำการพิจารณากลุ่มคำที่ปรากฏในแต่ละหัวข้อย่อย ของคำอธิบายรายวิชาด้วยวิธีการ N-gram โดยเทียบกับคำศัพท์ในพจนานุกรมภาษาอังกฤษ ซึ่งหากพบคำศัพท์ที่มีการสะกดผิด จะทำการแทนที่คำศัพท์นั้นด้วยคำที่ถูกต้องจากพจนานุกรม ซึ่งจะแสดงตัวอย่างของการแก้ไขคำผิดในแต่ละหัวข้อย่อย

โดยมีตัวอย่างการแก้ไขคำผิด ดังภาพที่ 7 ซึ่งเป็นตัวอย่างการการแก้ไขคำผิดในแต่ละหัวข้อย่อย โดยเมื่อทำการตรวจสอบการสะกดคำในแต่ละหัวข้อย่อยแล้ว จะพบว่าเนื้อหาในหัวข้อย่อยที่ 5 คือ “effcient algorithms for sorting” เมื่อตรวจสอบจึงพบคำว่า ‘effcient’ มีการสะกดคำที่ผิดเกิดขึ้น (คำที่เป็นสีแดง) จึงต้องทำการแก้ไขคำคำนี้ให้มีการสะกดคำที่ถูกต้อง โดยการเปรียบเทียบคำที่พบว่ามีการสะกดผิดกับคำในพจนานุกรม ซึ่งเมื่อดำเนินการเปรียบเทียบแล้ว พบว่าคำที่มีการสะกดคำที่ถูกต้องของคำคำนี้คือ ‘efficient’ (คำที่เป็นสีเขียว) ดังนั้นผลลัพธ์ที่ได้จากการแก้ไขคำผิดของเนื้อหาในหัวข้อย่อย ที่ 5 คือ “efficient algorithms for sorting”



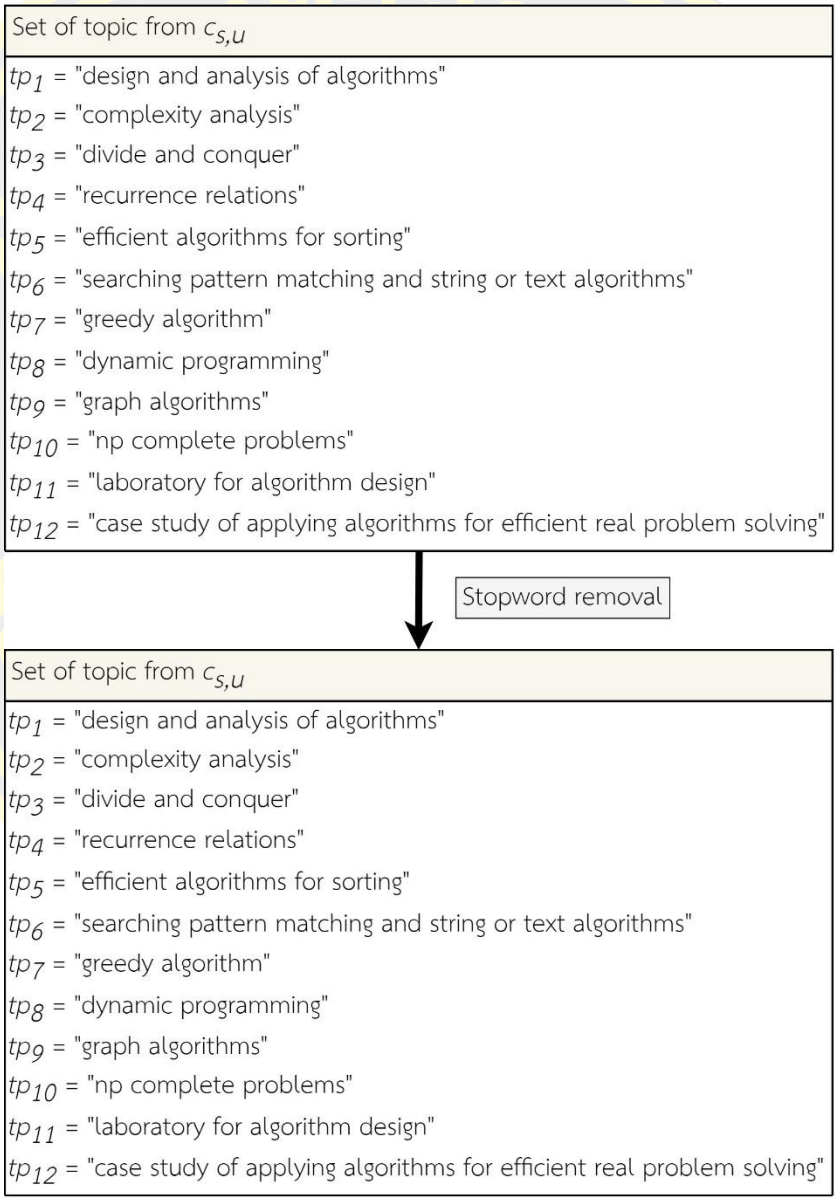
ภาพที่ 7 ตัวอย่างการการแก้ไขคำผิดในแต่ละหัวข้อย่อยของระบบ CSCDA

3.2.1.3 การกำจัดคำหยุด (Stopword removal)

ในเทคนิคนี้จะเป็นการกำจัดคำหนึ่ง ๆ ในแต่ละหัวข้อย่อย โดยคำที่ถูกกำจัดออกจะต้องเป็นคำที่ไม่ได้มีความหมายในตัวของมันเอง และเป็นคำที่เมื่อกำจัดออกไปแล้วความหมายของประโยคจะไม่เกิดการเปลี่ยนแปลงไป เช่น คำว่า ‘a’ ‘an’ ‘all’ ‘any’ ‘more’ ‘most’ ‘only’ ‘same’ ‘that’ ‘the’ ‘their’ ‘them’ ‘very’ เป็นต้น โดยในงานวิจัยนี้ได้กำหนดเซตของคำหยุดที่ต้องลบออกจากการพิจารณา ได้แก่ คำหยุดประเภทคำบุพบท (Preposition) อาทิเช่น คำว่า ‘by’, ‘for’, ‘in’, ‘on’

หรือ ‘to’ เป็นต้น และ คำหยุดประเภทคำสันธาน (Conjunction) อาทิเช่น คำว่า ‘and’ หรือ ‘or’ เป็นต้น ซึ่งคำหยุดทั้ง 2 ประเภทนี้จะไม่ถูกกำจัดออกไปจากเนื้อหาในคำอธิบายรายวิชา

โดยมีตัวอย่างของการกำจัดคำหยุดในแต่ละหัวข้อย่อย ดังภาพที่ 8 ซึ่งเมื่อทำการตรวจสอบในแต่ละหัวข้อย่อยแล้ว จะพบว่าในแต่ละหัวข้อย่อยไม่มีคำหยุดที่สมควรกำจัดออกปรากฏอยู่ในเนื้อหาเลย ดังนั้นทำให้ผลลัพธ์ที่ได้จากการกำจัดคำหยุดในทุก ๆ หัวข้อย่อยยังคงมีเนื้อหาเช่นเดิม



ภาพที่ 8 ตัวอย่างการกำจัดคำหยุดในแต่ละหัวข้อย่อยของระบบ CSCDA

3.2.1.4 การระบุหน้าที่ของคำ (Part-of-speech tagging)

การระบุถึงหน้าที่ของคำที่ปรากฏในหัวข้อย่อยหนึ่ง ๆ จะช่วยให้เข้าใจเกี่ยวกับหน้าที่ของคำที่ช่วยระบุถึงเนื้อหาที่สำคัญ โดยในระบบ *CSCDA* ได้ประยุกต์ใช้การระบุหน้าที่ของคำจาก Stanford part of speech tagger (Toutanova, Klein, Manning, & Singer, 2003) ซึ่งสามารถช่วยระบุหน้าที่ของคำในลักษณะต่าง ๆ โดยในการระบุถึงหน้าที่ของคำจะใช้อักษรย่อกำกับไว้หลังคำหนึ่ง ๆ เช่น

1) คำนาม Possessive pronoun ประกอบด้วย
   ‘NN’ (Noun, singular เช่น (‘word’, ‘NN’)) และ
   ‘NNS’ (Noun, plural เช่น (‘words’, ‘NNS’))
2) คำนามชี้เฉพาะ (Proper noun) ประกอบด้วย
   ‘NNP’ (Proper noun, singular เช่น (‘Smith’, ‘NNP’))
   ‘NNPS’ (Proper noun, plural เช่น (‘Europeans’, ‘NNPS’))
3) คำสรรพนาม (Pronoun) ประกอบด้วย
   ‘PRP’ (Personal pronoun เช่น (‘she’, ‘PRP’))
   ‘PRP$’ (Possessive pronoun เช่น (‘hers’, ‘PRP$’))
4) คำกริยา (Verb) ประกอบด้วย
   ‘VB’ (Verb, base form เช่น (‘get’, ‘VB’)),
   ‘VBD’ (Verb, past tense เช่น (‘got’, ‘VBD’)),
   ‘VBG’ (Verb, present participle เช่น (‘getting’, ‘VBG’)),
   ‘VBN’ (Verb, past participle เช่น (‘gotten’, ‘VBN’)),
   ‘VBP’ (Verb, non-3rd person singular present เช่น (‘get’, ‘VBP’)) และ
   ‘VBZ’ (Verb, 3rd person singular present เช่น (‘gest’, ‘VBZ’))
5) คำคุณศัพท์ (Adjective) ประกอบด้วย
   ‘JJ’ (Adjective เช่น (‘good’, ‘JJ’)),
   ‘JJR’ (Adjective, comparative เช่น (‘better’, ‘JJR’)) และ
   ‘JJS’ (Adjective, superlative เช่น (‘best’, ‘JJS’))
6) คำกริยาวิเศษณ์ (Adverb) ประกอบด้วย
   ‘RB’ (Adverb เช่น (‘badly’, ‘JJR’))
   ‘RBR’ (Adverb, comparative เช่น (‘worse’, ‘JJR’))

‘RBS’ (Adverb, superlative เช่น (‘worst’, ‘JJR’))

7) คำบุพบท (Preposition) ประกอบด้วย

‘IN’ (Preposition เช่น (‘on’, ‘IN’))

8) คำสันธาน (Conjunction) ประกอบด้วย

‘CC’ (Coordinating conjunction เช่น (‘or’, ‘CC’))

9) คำอุทาน (Interjection) ประกอบด้วย

‘UH’ (Interjection เช่น (‘wow’, ‘UH’))

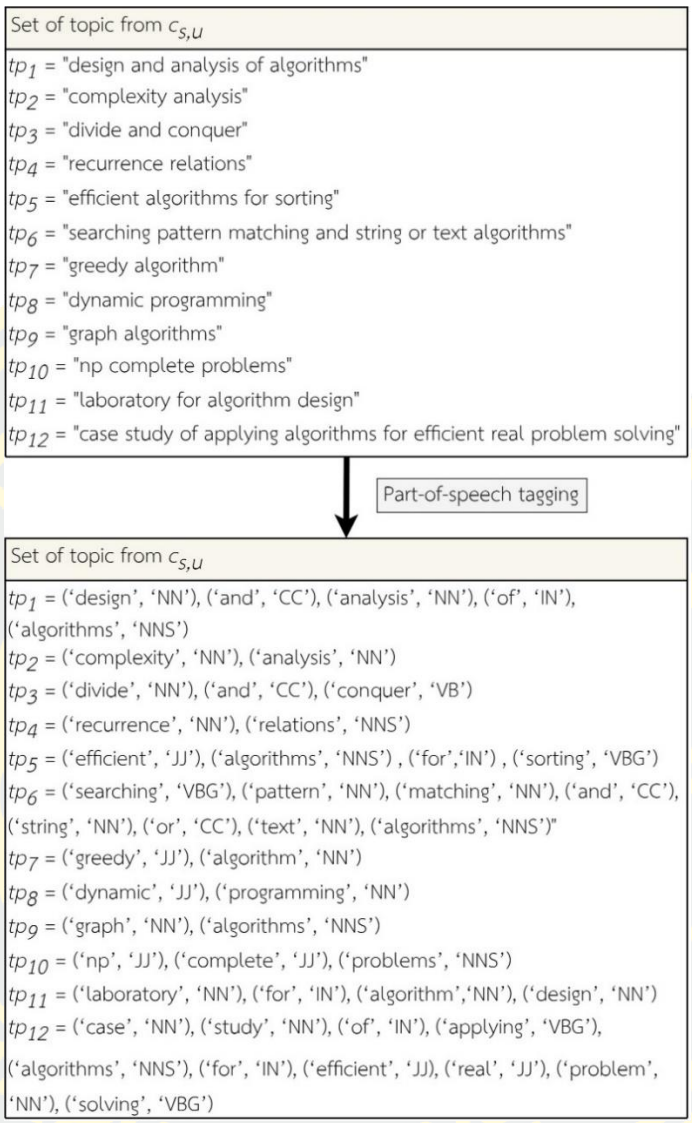
10) ตัวเลข (Digit) ประกอบไปด้วย

‘CD’ (cardinal digit เช่น (‘1’, ‘CD’))

11) คำกำกับคำนาม (Determiner) ประกอบไปด้วย

‘DT’ (determiner เช่น (‘an’, ‘DT’))

โดยจะแสดงตัวอย่างของการระบุหน้าที่ของคำในทุก ๆ หัวข้อย่อย ดังภาพที่ 9



ภาพที่ 9 ตัวอย่างการระบุหน้าที่ของคำในแต่ละหัวข้อย่อยของระบบ CSCDA

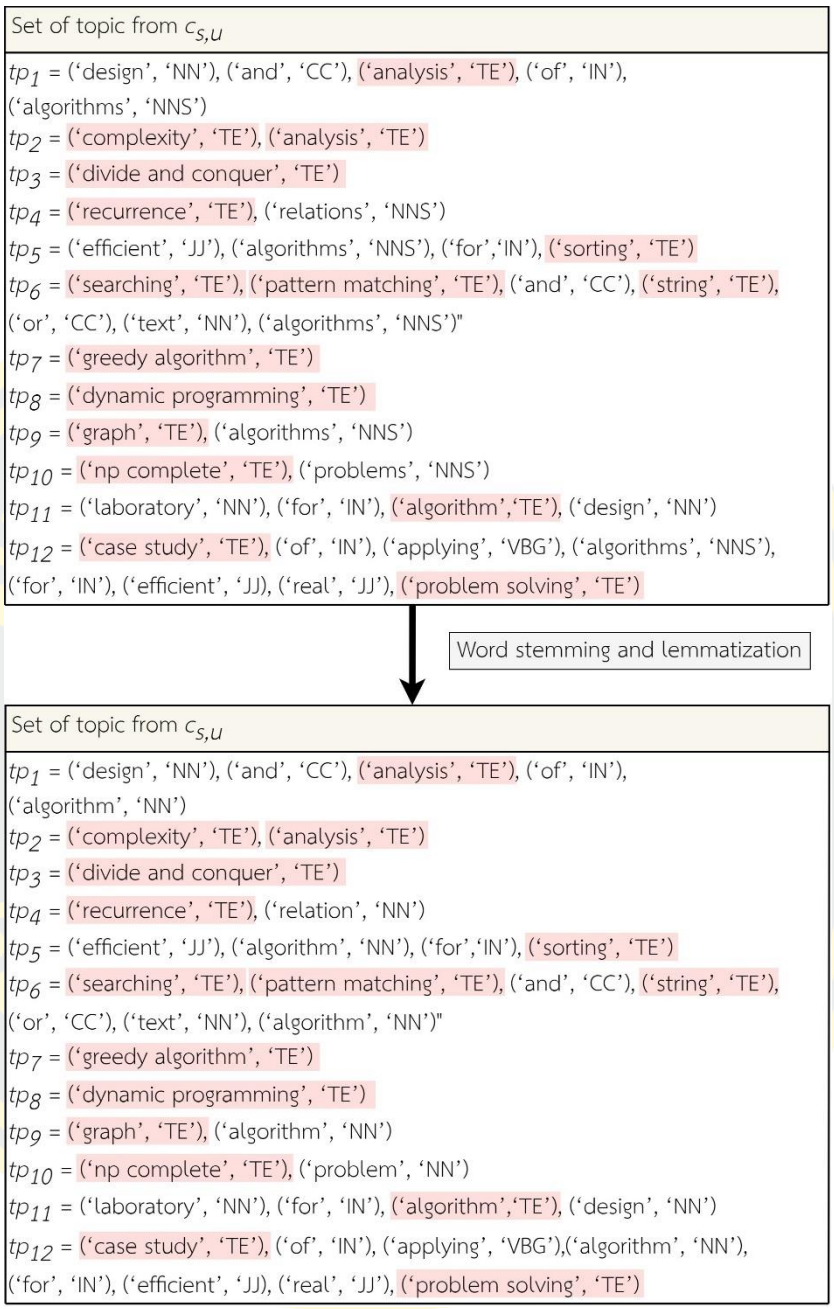
3.2.2 การระบุคำศัพท์เฉพาะ

หลังจากการประมวลข้อความเบื้องต้น จะเป็นการระบุถึงเนื้อหาที่สำคัญ ผ่านการระบุถึงคำศัพท์เฉพาะที่ปรากฏอยู่ในแต่ละหัวข้อย่อยของคำอธิบายรายวิชาหนึ่ง ๆ โดยการดำเนินการดังกล่าว จะประยุกต์ใช้วิธีการ N-gram (Lopez-Gazpio, Maritxalar, Lapata, & Agirre, 2019) เพื่อพิจารณากลุ่มคำที่ปรากฏในหัวข้อย่อยหนึ่ง ๆ และทำการเปรียบเทียบกับคำศัพท์เฉพาะที่รวมรวบไว้ใน “CS’s terminology corpus” (รวบรวมไว้ในขั้นตอนที่ 3.1.2) หากกลุ่มของคำหนึ่ง ๆ เหมือนกับคำศัพท์เฉพาะหนึ่ง ๆ จะทำการระบุว่ากลุ่มคำนั้นเป็นคำศัพท์เฉพาะ และทำการเปลี่ยนป้ายกำกับหน้าที่ของกลุ่มคำนั้นให้เป็น ‘TE’ (Terminology) แต่ในทางกลับกัน คำที่ไม่ได้ถูกระบุว่า

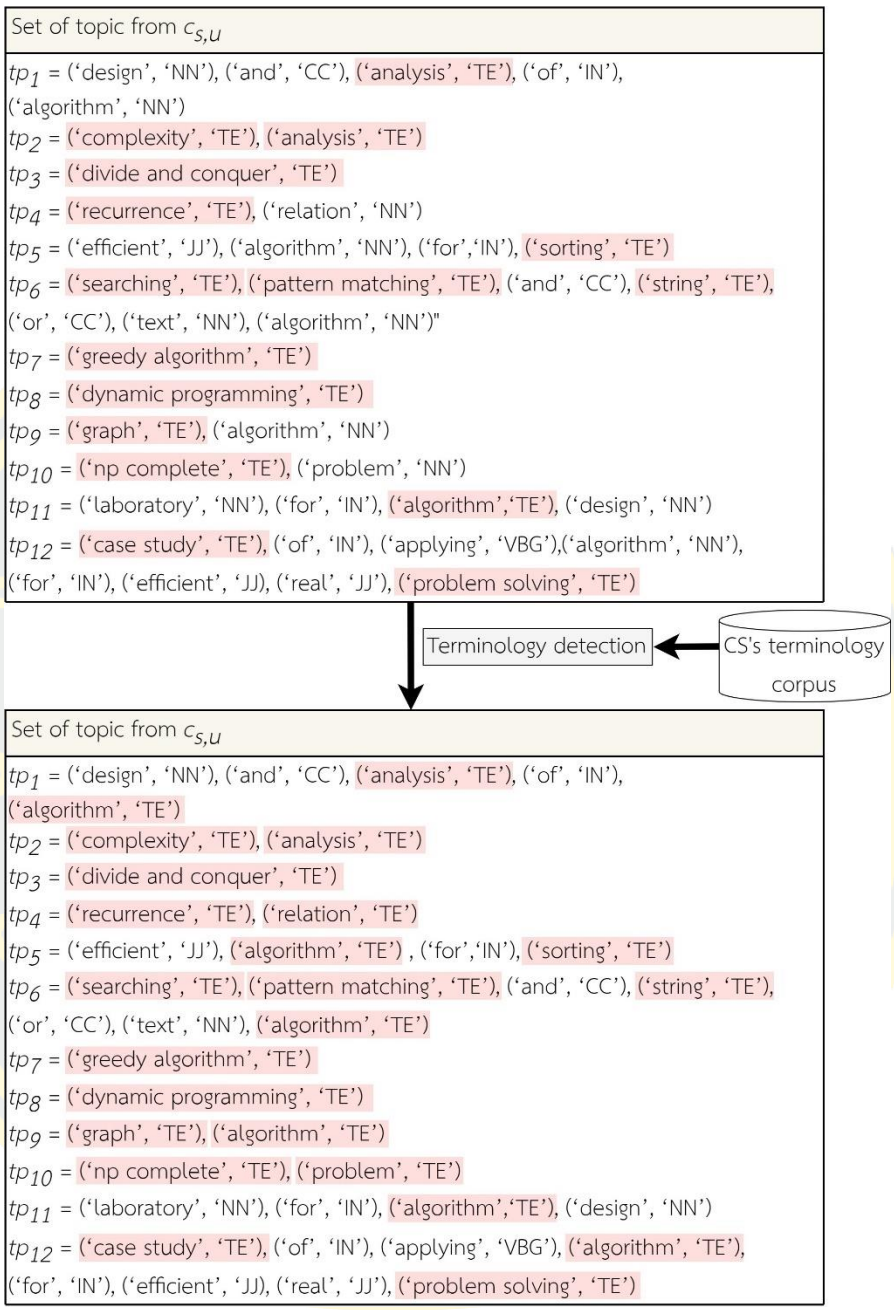
เป็นคำศัพท์เฉพาะจะถูกแปลงให้อยู่ในรูปรากศัพท์ ด้วยการใช้เทคนิคการแปลงรูปคำให้อยู่ในรากศัพท์ (Word stemming and lemmatization) (Balakrishnan & Lloyd-Yemoh, 2014) จากนั้นจะทำการพิจารณากลุ่มของคำและเปรียบเทียบกับคำศัพท์เฉพาะที่เก็บรวมรวมไว้อีกครั้งหนึ่ง เพื่อที่จะทำให้สามารถระบุถึงคำศัพท์เฉพาะได้แม่นยำมากขึ้น

โดยตัวอย่างของการระบุคำศัพท์เฉพาะในแต่ละหัวข้อจะถูกแสดงดังภาพที่ 10 ซึ่งเป็นการนำผลลัพธ์ที่ได้จากการดำเนินงานในขั้นตอนของการประมวลผลข้อความเบื้องต้นที่กล้าวมาข้างต้น มาดำเนินการระบุถึงคำศัพท์เฉพาะในแต่ละหัวข้อย่อย ซึ่งเมื่อพบว่าคำหรือกลุ่มคำนั้น ๆ เป็นคำศัพท์เฉพาะจะทำการปรับเปลี่ยนป้ายกำกับหน้าที่ของคำหรือกลุ่มคำนั้น ๆ ให้เป็น ‘TE’ (คำที่มีพื้นหลังสีแดง) อาทิเช่น หัวข้อย่อยที่ 1 คือ “(‘design’, ‘NN’), (‘and’, ‘CC’), (‘analysis’, ‘NN’), (‘of’, ‘IN’), (‘algorithms’, ‘NNS’)” เมื่อทำการตรวจสอบแต่ละคำในหัวข้อย่อยแล้ว ทำให้พบคำว่า (‘analysis’, ‘NN’) เป็นคำศัพท์เฉพาะ ดังนั้นจึงทำการเปลี่ยนป้ายกำกับหน้าที่ของคำจากเดิมคือ ‘NN’ ให้กลายเป็น ‘TE’ จะได้เป็น (‘analysis’, ‘TE’) ดังนั้นผลลัพธ์ที่ได้ที่ได้จากการระบุคำศัพท์เฉพาะในครั้งที่ 1 ของหัวข้อย่อยที่ 1 คือ “(‘design’, ‘NN’), (‘and’, ‘CC’), (‘analysis’, ‘TE’), (‘of’, ‘IN’), (‘algorithms’, ‘NNS’)” ต่อมาในภาพที่ 11 จะเป็นการดำเนินการแปลงรูปคำแต่ละคำที่ยังไม่ได้ถูกระบุว่าเป็นคำศัพ์เฉพาะ (คำที่ไม่ได้มีป้ายกำกับหน้าที่ของคำเป็น ‘TE’) ให้อยู่ในรูปของรากศัพท์ของคำคำนั้น อาทิเช่น หัวข้อย่อยที่ 1 คือ “(‘design’, ‘NN’), (‘and’, ‘CC’), (‘analysis’, ‘TE’), (‘of’, ‘IN’), (‘algorithms’, ‘NNS’)” เมื่อนำคำแต่ละคำมาทำการแปลงให้อยู่ในรูปของรากศัพท์ ทำให้คำว่า (‘algorithms’, ‘NNS’) ถูกแปลงให้กลายเป็นคำว่า (‘algorithm’, ‘NN’) ซึ่งผลลัพธ์ที่ได้จากการแปลงรูปคำให้อยู่ในรากศัพท์ในหัวข้อย่อยที่ 1 จะได้เป็น “(‘design’, ‘NN’), (‘and’, ‘CC’), (‘analysis’, ‘TE’), (‘of’, ‘IN’), (‘algorithm’, ‘NN’)” สุดท้ายในภาพที่ 12 จะเป็นการระบุถึงคำศัพท์เฉพาะในทุก ๆ หัวข้อย่อยอีกครั้งหนึ่ง อาทิเช่น หัวข้อย่อยที่ 1 คือ “(‘design’, ‘NN’), (‘and’, ‘CC’), (‘analysis’, ‘TE’), (‘of’, ‘IN’), (‘algorithm’, ‘NN’)” เมื่อทำการตรวจสอบเพื่อระบุถึงคำศัพท์เฉพาะจะพบคำว่า (‘algorithm’, ‘NN’) เป็นคำศัพท์เฉพาะ ดังนั้นจึงทำการเปลี่ยนป้ายกำกับหน้าที่ของคำจากเดิมคือ ‘NN’ ให้กลายเป็น ‘TE’ จะได้เป็น (‘algorithm’, ‘TE’) จากการระบุถึงคำศัพท์เฉพาะในครั้งที่ 2 ของหัวข้อย่อที่ 1 ทำให้ได้มาซึ่งผลลัพธ์คือ “(‘design’, ‘NN’), (‘and’, ‘CC’), (‘analysis’, ‘TE’), (‘of’, ‘IN’), (‘algorithm’, ‘TE’)”

ภาพที่ 10 ตัวอย่างการระบุถึงคำศัพท์เฉพาะในแต่ละหัวข้อย่อยของระบบ CSCDA



ภาพที่ 11 ตัวอย่างการระบุถึงคำศัพท์เฉพาะในแต่ละหัวข้อย่อยของระบบ CSCDA (ต่อ)



ภาพที่ 12 ตัวอย่างการระบุถึงคำศัพท์เฉพาะในแต่ละหัวข้อย่อยของระบบ CSCDA (ต่อ)

3.2.3 การสกัดคำสำคัญ

กฎทางภาษาศาสตร์ที่ได้รวบรวมไว้ในส่วนที่ 3.1.3 จะถูกประยุกต์ใช้เพื่อทำการสกัดคำสำคัญที่อาจปรากฏอยู่ในแต่ละหัวข้อย่อยของคำอธิบายรายวิชา โดยการประมวลผลเริ่มจากพิจารณากลุ่มของคำที่ปรากฏในแต่ละหัวข้อย่อย จากนั้นนำแต่ละกลุ่มคำมาประยุกต์เข้ากับกฎที่เตรียมไว้ก่อนหน้า โดยผลลัพธ์จากการประยุกต์ใช้กฎจะเป็นการระบุได้ว่ากลุ่มของคำที่พิจารณาเป็นคำสำคัญหรือไม่ หากกลุ่มของคำที่พิจารณาเป็นคำสำคัญจะทำการกำหนดป้ายกำกับคำให้กับกลุ่มคำนั้นเป็น ‘KW’ (Keyword) โดยมีตัวอย่างการการประยุกต์ใช้กฎทางภาษาศาสตร์ในการสกัดคำสำคัญของแต่ละหัวข้อย่อย ดังต่อไปนี้

หัวข้อย่อยที่ 1 คือ “(‘design’, ‘NN’), (‘and’, ‘CC’), (‘analysis’, ‘TE’), (‘of’, ‘IN’), (‘algorithm’, ‘TE’)” เมื่อทำการตรวจสอบพบว่ามีคำศัพท์เฉพาะปรากฏอยู่คือ (‘analysis’, ‘TE’) และ (‘algorithm’,‘TE’) ดังนั้นในการสกัดคำสำคัญจะทำการประยุกต์ใช้กฎทางภาษาศาสตร์ประเภทที่ 1 คือ

กฎ : “Noun (NN) + Conjunction (CC) + Terminology (TE1) + Preposition (IN) + Terminology (TE2)” ⟶ ⟨‘Terminology (TE2) + Noun (NN)’, ‘Terminology (TE2) + Terminology (TE1)’⟩

จากการประยุกต์ใช้กฎทางภาษาศาสตร์ข้างต้นในการสกัดคำสำคัญ ทำให้คำสำคัญที่สกัดได้คือ ‘algorithm design(KW)’ และ ‘algorithm analysis(KW)’ ดังนั้นผลลัพธ์จากการสกัดคำสำคัญในหัวข้อย่อยที่ 1 จะได้ดังต่อไปนี้ ⟨‘algorithm design(KW)’, ‘algorithm analysis(KW)’⟩

หัวข้อย่อยที่ 2 คือ “(‘complexity’, ‘TE’), (‘analysis’, ‘TE’)” เมื่อทำการตรวจสอบพบว่ามีคำศัพท์เฉพาะปรากฏอยู่คือ (‘complexity’, ‘TE’) และ (‘analysis’, ‘TE’) ดังนั้นในการสกัดคำสำคัญจะทำการประยุกต์ใช้กฎทางภาษาศาสตร์ประเภทที่ 1 คือ

กฎ : “Terminology (TE1) + Terminology (TE2)” ⟶ ⟨‘Terminology (TE1) + Terminology (TE2)’⟩

จากการประยุกต์ใช้กฎทางภาษาศาสตร์ข้างต้นในการสกัดคำสำคัญ ทำให้คำสำคัญที่สกัดได้คือ‘complexity analysis(KW)’ ดังนั้นผลลัพธ์จากการสกัดคำสำคัญในหัวข้อย่อยที่ 2 จะได้ดังต่อไปนี้ ⟨‘complexity analysis(KW)’⟩

หัวข้อย่อยที่ 3 คือ “(‘divide and conquer’, ‘TE’)” เมื่อทำการตรวจสอบพบว่ามีคำศัพท์เฉพาะปรากฏอยู่คือ (‘divide and conquer’, ‘TE’) ดังนั้นในการสกัดคำสำคัญจะทำการประยุกต์ใช้กฎทางภาษาศาสตร์ประเภทที่ 1 คือ

กฎ : “Terminology (TE)” ⟶ ⟨‘Terminology’⟩

จากการประยุกต์ใช้กฎทางภาษาศาสตร์ข้างต้นในการสกัดคำสำคัญ ทำให้คำสำคัญที่สกัดได้คือ‘divide and conquer(TE)’ ดังนั้นผลลัพธ์จากการสกัดคำสำคัญในหัวข้อย่อยที่ 3 จะได้ดังต่อไปนี้ ⟨‘divide and conquer(TE)’⟩

หัวข้อย่อยที่ 4 คือ “(‘recurrence’, ‘TE’), (‘relation’, ‘TE’)” เมื่อทำการตรวจสอบพบว่ามีคำศัพท์เฉพาะปรากฏอยู่คือ (‘recurrence’, ‘TE’) และ (‘relation’, ‘TE’) ดังนั้นในการสกัดคำสำคัญจะทำการประยุกต์ใช้กฎทางภาษาศาสตร์ประเภทที่ 1 คือ

กฎ : “ Terminology (TE1) + Terminology (TE2)” ⟶ ⟨‘Terminology (TE1) + Terminology (TE2)’⟩

จากการประยุกต์ใช้กฎทางภาษาศาสตร์ข้างต้นในการสกัดคำสำคัญ ทำให้คำสำคัญที่สกัดได้คือ ‘recurrence relation(KW)’ ดังนั้นผลลัพธ์จากการสกัดคำสำคัญในหัวข้อย่อยที่ 4 จะได้ดังต่อไปนี้ ⟨‘recurrence relation(KW)’⟩

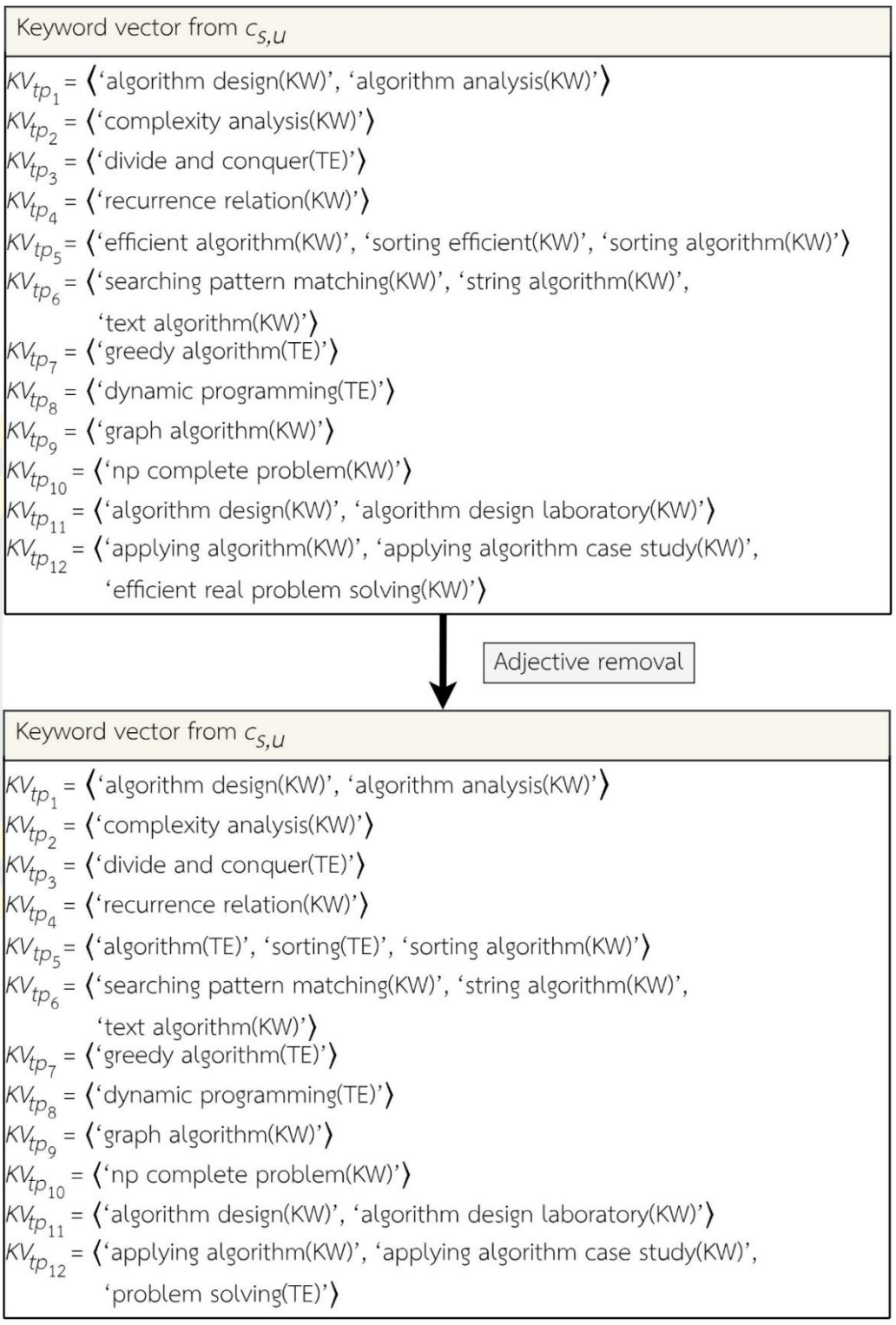
โดยจะแสดงให้เห็นถึงการสกัดคำสำคัญในแต่ละหัวข้อย่อย ดังภาพที่ 13

Set of topic from $c_{s,u}$
tp_1 = (‘design’, ‘NN’), (‘and’, ‘CC’), (‘analysis’, ‘TE’), (‘of’, ‘IN’),(‘algorithm’, ‘TE’) Linguistic rule : (W_0 =‘NN’) + (W_1 =‘CC’) + (W_2 =‘TE1’) + (W_3 =‘IN’) + (W_4 =‘TE2’) ⟶ ⟨‘TE2 + NN’, ‘TE2 + TE1’⟩ KV_{tp_1} = ⟨‘algorithm design(KW)’, ‘algorithm analysis(KW)’⟩
tp_2 = (‘complexity’, ‘TE’), (‘analysis’, ‘TE’) Linguistic rule : (W_0 =‘TE1’) + (W_1 =‘TE2’) ⟶ ⟨‘TE1 + TE2’⟩ KV_{tp_2} = ⟨‘complexity analysis(KW)’⟩
tp_3 = (‘divide and conquer’, ‘TE’) Linguistic rule : (W_0 =‘TE’) ⟶ ⟨‘TE’⟩ KV_{tp_3} = ⟨‘divide and conquer(TE)’⟩
tp_4 = (‘recurrence’, ‘TE’), (‘relation’, ‘TE’) Linguistic rule : (W_0 =‘TE1’) + (W_1 =‘TE2’) ⟶ ⟨‘TE1 + TE2’⟩ KV_{tp_4} = ⟨‘recurrence relation(KW)’⟩
tp_5 = (‘efficient’, ‘JJ’), (‘algorithm’, ‘TE’) , (‘for’,‘IN’), (‘sorting’, ‘TE’) Linguistic rule : (W_0 =‘JJ’) + (W_1 =‘TE’) + (W_2 =‘IN’) + (W_3 =‘TE’) ⟶ ⟨‘JJ + TE1’, ‘TE2 + JJ’, ‘TE2 + TE1’⟩ KV_{tp_5} = ⟨‘efficient algorithm(KW)’, ‘sorting efficient(KW)’, ‘sorting algorithm(KW)’⟩
tp_6 = (‘searching’, ‘TE’), (‘pattern matching’, ‘TE’), (‘and’, ‘CC’), (‘string’, ‘TE’), (‘or’, ‘CC’), (‘text’, ‘NN’), (‘algorithm’, ‘TE’) Linguistic rule : (W_0 =‘TE1’) + (W_1 =‘TE2’) + (W_2 =‘CC’) + (W_3 =‘TE3’) + (W_4 =‘CC’) + (W_5 =‘NN’) + (W_6 =‘TE4’) ⟶ ⟨‘TE1 + TE2’, ‘TE3 + TE4’, ‘TE3 + NN’⟩ KV_{tp_6} = ⟨‘searching pattern matching(KW)’, ‘string algorithm(KW)’, ‘text algorithm(KW)’⟩
tp_7 = (‘greedy algorithm’, ‘TE’) Linguistic rule : (W_0 =‘TE’) ⟶ ⟨‘TE’⟩ KV_{tp_7} = ⟨‘greedy algorithm(TE)’⟩
tp_8 = (‘dynamic programming’, ‘TE’) Linguistic rule : (W_0 =‘TE’) ⟶ ⟨‘TE’⟩ KV_{tp_8} = ⟨‘dynamic programming(TE)’⟩
tp_9 = (‘graph’, ‘TE’), (‘algorithm’, ‘TE’) Linguistic rule : (W_0 =‘TE1’) + (W_1 =‘TE2’) ⟶ ⟨‘TE1 + TE2’⟩ KV_{tp_9} = ⟨‘graph algorithm(KW)’⟩
tp_{10} = (‘np complete’, ‘TE’), (‘problem’, ‘TE’) Linguistic rule : (W_0 =‘TE1’) + (W_1 =‘TE2’) ⟶ ⟨‘TE1 + TE2’⟩ $KV_{tp_{10}}$ = ⟨‘np complete problem(KW)’⟩
tp_{11} = (‘laboratory’, ‘NN’), (‘for’, ‘IN’), (‘algorithm’,‘TE’), (‘design’, ‘NN’) Linguistic rule : (W_0 =‘NN1’) + (W_1 =‘IN’) + (W_2 =‘TE’) + (W_3 =‘NN2’) ⟶ ⟨‘TE + NN2’, ‘TE + NN2 + NN1’⟩ $KV_{tp_{11}}$ = ⟨‘algorithm design(KW)’, ‘algorithm design laboratory(KW)’⟩
tp_{12} = (‘case study’, ‘TE’), (‘of’, ‘IN’), (‘applying’, ‘VBG’), (‘algorithm’, ‘TE’), (‘for’, ‘IN’), (‘efficient’, ‘JJ), (‘real’, ‘JJ’), (‘problem solving’, ‘TE’) Linguistic rule : (W_0 =‘TE1’) + (W_1 =‘IN’) + (W_2 =‘VBG’) + (W_3 =‘TE2’) + (W_4 =‘IN’) + (W_5 =‘JJ’) + (W_6 =‘JJ’) + (W_7 =‘TE3’) ⟶ ⟨‘VBG + TE2’, ‘VBG + TE2 + TE1’, ‘JJ + JJ + TE3’⟩ $KV_{tp_{12}}$ = ⟨‘applying algorithm(KW)’, ‘applying algorithm case study(KW)’, ‘efficient real problem solving(KW)’⟩

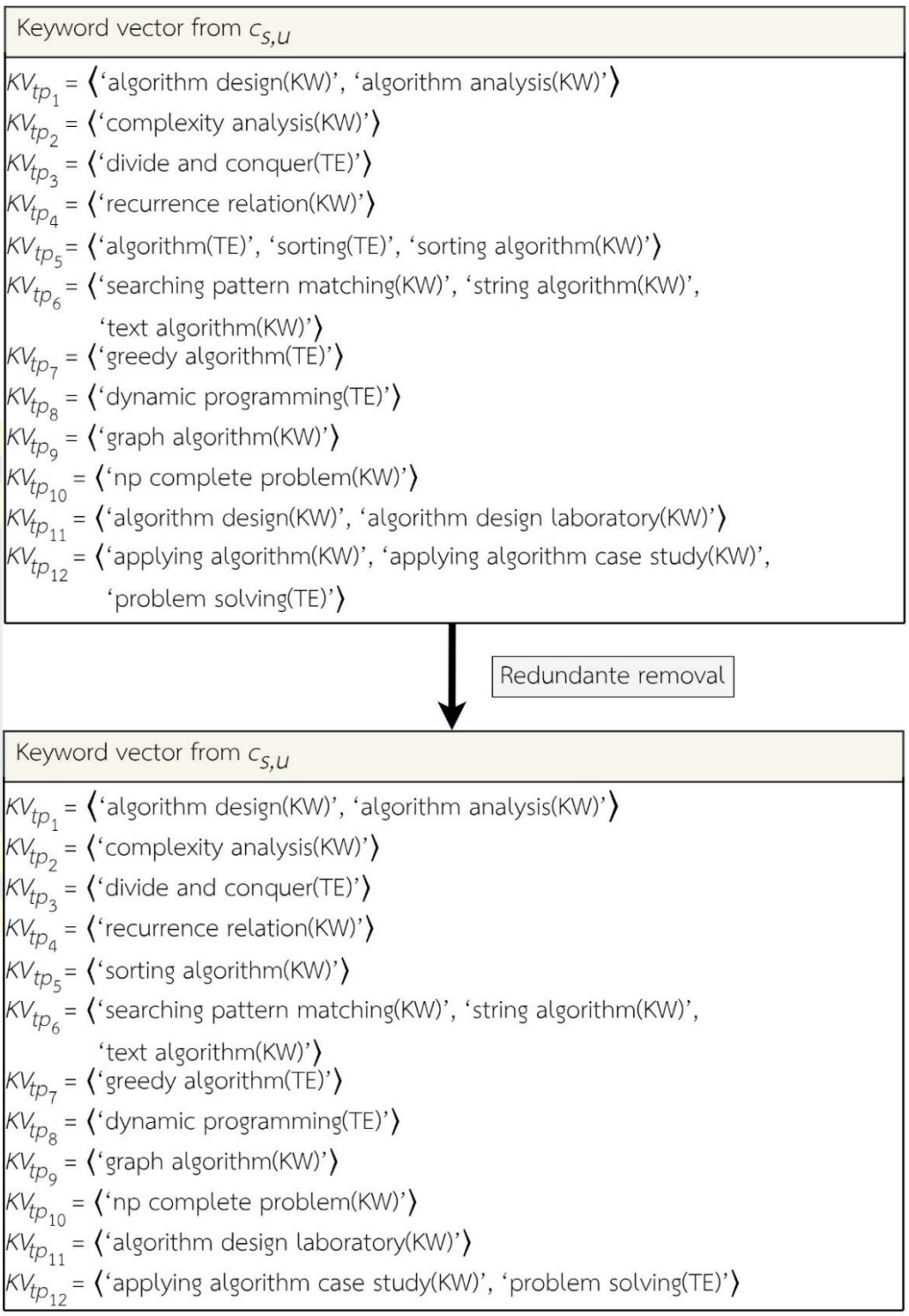
ภาพที่ 13 ตัวอย่างคำสำคัญที่สกัดได้ในแต่ละหัวข้อย่อยของระบบ CSCDA

3.2.4 การลดทอนเนื้อหาที่ไม่สำคัญ

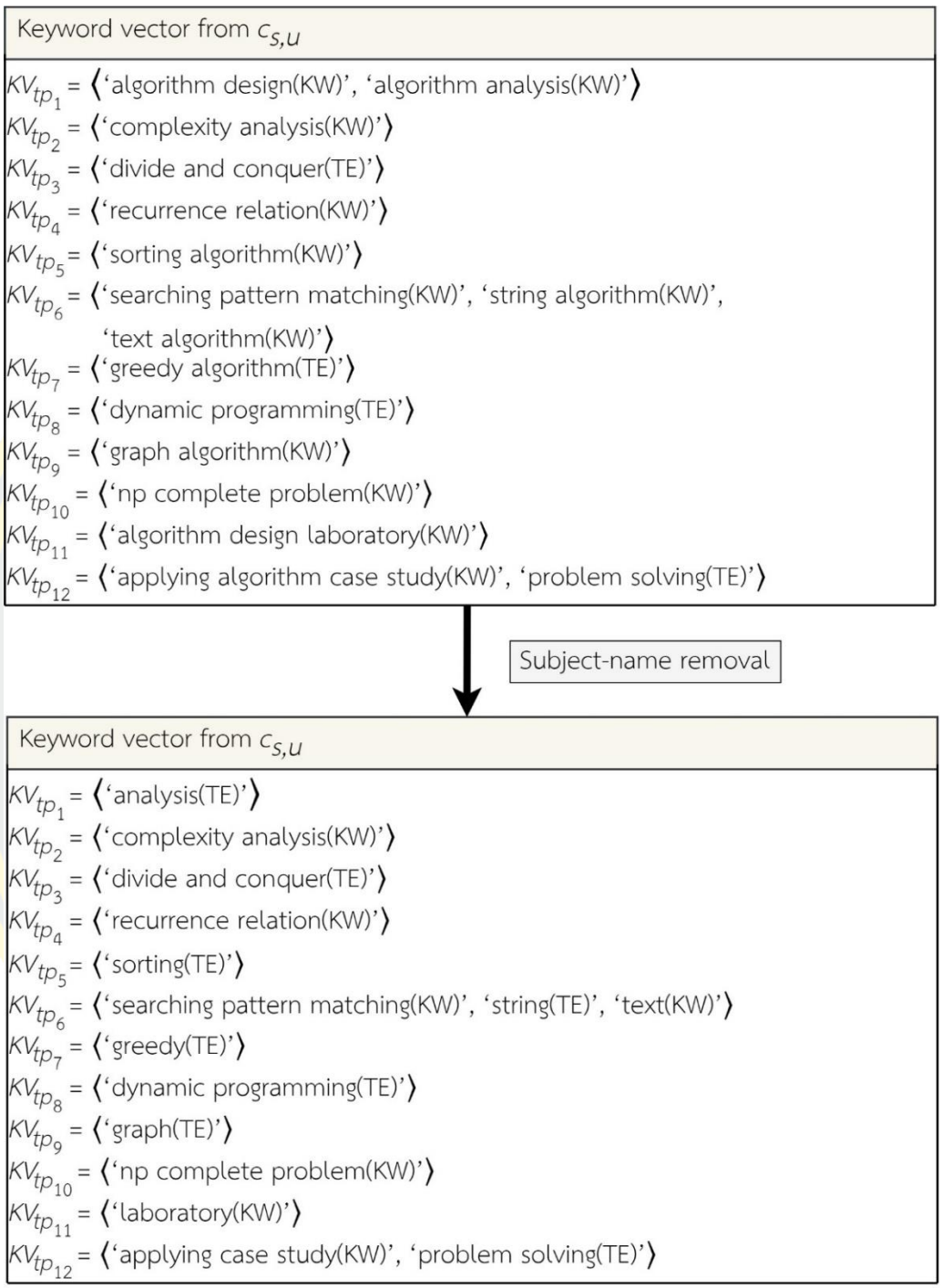
หลังจากการประยุกต์ใช้กฎทางภาษาศาสตร์ในการสกัดคำสำคัญแล้ว หัวข้อย่อยหนึ่ง ๆ ของคำอธิบายรายวิชาหนึ่ง ๆ อาจประกอบด้วย คำทั่วไป (เช่น คำคุณศัพท์ (Adjective), คำนาม (Noun), คำกริยา (Verb), คำคุณศัพท์ (Adjective) + คำนาม (Noun), คำกริยา (Verb) + คำนาม (Noun), คำนาม (Noun) + คำนาม (Noun), คำคุณศัพท์ (Adjective) + คำนาม (Noun) + คำนาม (Noun), คำกริยา (Verb) + คำนาม (Noun) + คำนาม (Noun) เป็นต้น) คำศัพท์เฉพาะ (Terminology) และคำสำคัญ (Keyword) ผสมกันอยู่ ถึงแม้ว่าคำศัพท์เฉพาะและคำสำคัญสามารถช่วยบ่งบอกถึงเนื้อหาที่สำคัญได้ แต่อย่างไรก็ตาม หัวข้อย่อยนั้น ๆ อาจมีคำที่ไม่ได้สื่อถึงเนื้อหาที่สำคัญ ด้วยเหตุนี้ จึงควรที่จะพิจารณาลดทอนคำที่ไม่สำคัญออกจากหัวข้อย่อยนั้น ๆ โดยแบ่งการลดทอนออกเป็น 3 กรณี คือ 1) การลบส่วนของคำคุณศัพท์ 2) การลบส่วนของคำสำคัญที่ซ้ำซ้อนกัน และ 3) การลบส่วนของคำที่เป็นชื่อรายวิชา ตามลำดับ โดยจะแสดงตัวอย่างการลดทอนเนื้อหาที่ไม่สำคัญจากคำสำคัญที่สกัดในแต่ละหัวข้อย่อย ดังภาพที่ 14, 15 และ 16



ภาพที่ 14 ตัวอย่างการลบส่วนของคำคุณศัพท์ออกจากคำสำคัญในแต่ละหัวข้อย่อยของระบบ CSCDA



ภาพที่ 15 ตัวอย่างการลบส่วนของคำสำคัญที่ซ้ำซ้อนกันในแต่ละหัวข้อย่อยของระบบ CSCDA



ภาพที่ 16 ตัวอย่างการลบส่วนของคำที่เป็นชื่อรายวิชาในแต่ละหัวข้อย่อยของระบบ CSCDA

จากภาพที่ 14 จะแสดงให้เห็นถึงการลดทอนเนื้อหาที่ไม่สำคัญในส่วนของการลบคำคุณศัพท์ออกจากรายการของคำสำคัญที่สกัดได้ในแต่ละหัวข้อย่อย ซึ่งเมื่อพิจารณาถึงคำคุณศัพท์แล้วจะพบว่าคำสำคัญในหัวข้อย่อยที่ 5 ได้แก่ ⟨'efficient algorithm(KW)', 'sorting efficient(KW)', 'sorting algorithm(KW)'⟩ มีคำคุณศัพท์ปรากฏอยู่ในคำสำคัญ 'efficient algorithm(KW)' และ 'sorting

efficient(KW)’ คือคำว่า ‘efficient’ จึงทำการลบคำคุณศัพท์ที่พบออกจากคำสำคัญ ทำให้รายการคำสำคัญของหัวข้อย่อยที่ 5 หลังจากลบคำคุณศัพท์ออกไปแล้วคือ ⟨‘algorithm(TE)’, ‘sorting(TE)’, ‘sorting algorithm(KW)’⟩ และเมื่อพิจารณาหัวข้อย่อยต่อไป จะพบว่ารายการคำสำคัญในหัวข้อย่อที่ 12 คือ ⟨‘applying algorithm case study(KW)’, ‘efficient real problem solving(KW)’⟩ มีคำคุณศัพท์ปรากฏอยู่ในคำสำคัญ ‘efficient real problem solving(KW)’ คือคำว่า ‘efficient’ และ ‘real’ จึงทำการลบคำทั้ง 2 ออกจากคำสำคัญ ทำให้รายการคำสำคัญของหัวข้อย่อยที่ 12 หลังจากลบคำคุณศัพท์ออกไปแล้วคือ ⟨‘applying algorithm case study(KW)’, ‘problem solving(TE)’⟩ ต่อมาในภาพที่ 15 จะเป็นการลดทอนเนื้อหาที่ไม่สำคัญในส่วนของการลบส่วนของคำสำคัญที่ซ้ำซ้อนกันในแต่ละหัวข้อย่อย เมื่อพิจารณาจึงพบว่ารายการคำสำคัญในหัวข้อย่อยที่ 5 คือ ⟨‘algorithm(TE)’, ‘sorting(TE)’, ‘sorting algorithm(KW)’⟩ มีคำสำคัญที่ซ้ำซ้อนกันอยู่ กล่าวคือ คำสำคัญ ‘algorithm(TE)’ และ ‘sorting(TE)’ มีความซ้ำซ้อนกันกับคำสำคัญ ‘sorting algorithm(KW)’ ดังนั้นจึงต้องทำการลบคำสำคัญ ‘algorithm(TE)’ และ ‘sorting(TE)’ ออก ทำให้คำสำคัญของหัวข้อย่อยที่ 5 เหลือเพียง ⟨‘sorting algorithm(KW)’⟩, รายการคำสำคัญคำสำคัญในหัวข้อย่อยที่ 11 คือ ⟨‘algorithm design(KW)’, ‘algorithm design laboratory(KW)’⟩ มีคำสำคัญที่ซ้ำซ้อนกันอยู่ คือ คำสำคัญ ‘algorithm design(KW)’ มีความซ้ำซ้อนกันกับคำสำคัญ ‘algorithm design laboratory(KW)’ ดังนั้นจึงต้องทำการลบคำสำคัญ ‘algorithm design(KW)’ ออก ทำให้คำสำคัญของหัวข้อย่อยที่ 11 เหลือเพียง ⟨‘algorithm design laboratory(KW)’⟩, รายการคำสำคัญคำสำคัญในหัวข้อย่อยที่ 12 คือ ⟨‘applying algorithm(KW)’, ‘applying algorithm case study(KW)’, ‘problem solving(TE)’⟩ มีคำสำคัญที่ซ้ำซ้อนกันอยู่ คือ คำสำคัญ ‘applying algorithm(KW)’ มีความซ้ำซ้อนกันกับคำสำคัญ ‘applying algorithm case study(KW)’ ดังนั้นจึงต้องทำการลบคำสำคัญ ‘applying algorithm(KW)’ ออก ทำให้คำสำคัญของหัวข้อย่อยที่ 12 เหลือเพียง ⟨‘applying algorithm case study(KW)’, ‘problem solving(TE)’⟩ และ สุดท้ายในภาพที่ 16 เป็นการลดทอนเนื้อหาที่ไม่สำคัญในส่วนของการลบส่วนของคำที่เป็นชื่อรายวิชาในแต่ละหัวข้อย่อย โดยเมื่อทำการตรวจสอบในแต่ละหัวข้อย่อยจะพบว่าในหัวข้อที่ 1 และหัวข้อย่อยที่ 11 มีคำที่เป็นชื่อรายวิชา (“Algorithm Design and Applications”) คือคำว่า ‘algorithm design’ ปรากฏอยู่ในคำสำคัญ จึงทำให้ต้องลบคำคำนี้ออกจากคำสำคัญ ดังนั้นผลลัพธ์ของคำสำคัญหลังจากการลบคำที่เป็นชื่อรายวิชาออกในหัวข้อย่อยที่ 1 คือ ⟨‘analysis(TE)’⟩ และหัวข้อย่อยที่ 11 คือ ⟨‘laboratory(KW)’⟩ นอกจากนี้ในหัวข้อย่อยที่ 5, 6, 7, 9 และ 12 มีคำที่เป็นชื่อรายวิชาคือคำว่า ‘algorithm’ ปรากฏอยู่ในคำสำคัญทำให้ต้องลบคำคำนี้ออกจากคำสำคัญ ดังนั้น

ผลลัพธ์ที่ได้จากการลบคำที่เป็นชื่อรายวิชาในหัวข้อย่อยที่ 5 คือ ⟨‘sorting(TE)’⟩, หัวข้อย่อยที่ 6 คือ ⟨‘searching pattern matching(KW)’, ‘string(TE)’, ‘text(KW)’⟩, หัวข้อย่อยที่ 7 คือ ⟨‘greedy(TE)’⟩, หัวข้อย่อยที่ 9 คือ ⟨‘graph(TE)’⟩, หัวข้อย่อยที่ 12 คือ ⟨‘applying case study(KW)’, ‘problem solving(TE)’⟩

หลังจากทำการลดทอนคำที่ไม่สำคัญในแต่ละหัวข้อย่อยแล้ว คำต่าง ๆ ในหัวข้อย่อยจะถูกจัดเก็บอยู่ในเวคเตอร์คำสำคัญ (KV_{tp_i}) ซึ่งจะทำให้โครงสร้างการจัดเก็บข้อมูลของคำอธิบายรายวิชาหนึ่ง ๆ จะเปลี่ยนจาก $< s, u, TP = < tp_1, tp_2, \dots, tp_n >>$ กลายเป็น $< s, u, KV = \{KV_{tp_1}, KV_{tp_2}, \dots, KV_{tp_n}\} >$ เมื่อแต่ละ KV_{tp_i} จะประกอบไปด้วยคำทั่วไป, คำศัพท์เฉพาะ หรือ คำสำคัญ ที่ปรากฏอยู่ในเนื้อหาของหัวข้อย่อยนั้น ๆ

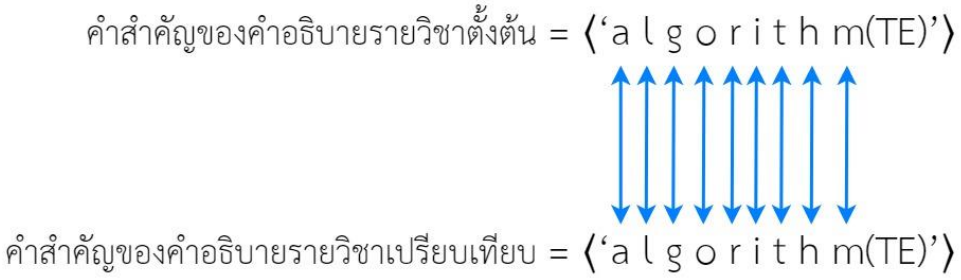
3.3 การเปรียบเทียบคำสำคัญระหว่างคำอธิบายรายวิชา

ขั้นตอนสุดท้ายของระบบ *CSCDA* คือ การเปรียบเทียบคำสำคัญที่สกัดได้จากแต่ละคำอธิบายรายวิชา ซึ่งจะเป็นการเปรียบเทียบระหว่าง 2 คำอธิบายรายวิชาที่เป็นรายวิชาเดียวกันหรือที่สอดคล้องกัน โดยการเปรียบเทียบคำอธิบายรายวิชาจะทำการกำหนดข้อมูลนำเข้า 2 ส่วน คือ 1) คำอธิบายรายวิชาตั้งต้น $(c_{s,x})$, และ 2) คำอธิบายรายวิชาเปรียบเทียบ $(c_{s,y})$ โดยผลลัพธ์ที่ได้จากการเปรียบเทียบระหว่าง 2 คำอธิบายรายวิชาจะแยกออกเป็น 2 ส่วน คือ เนื้อหาของคำอธิบายรายวิชาตั้งต้นที่เหมือนและแตกต่างกันกับคำอธิบายเปรียบเทียบ โดยขั้นตอนวิธีสำหรับทำการเปรียบเทียบจะเป็นการผสมสานกันระหว่าง 3 วิธี คือ 1) วิธีการเปรียบเทียบแบบตรงตัว (Exact Matching) 2) วิธีการเปรียบเทียบแบบเซตย่อย (Subset matching) และ 3) วิธีการเปรียบเทียบแบบซูเปอร์เซต (Superset matching) โดยวิธีการเปรียบเทียบทั้ง 3 วิธีจะมีรายละเอียดดังต่อไปนี้

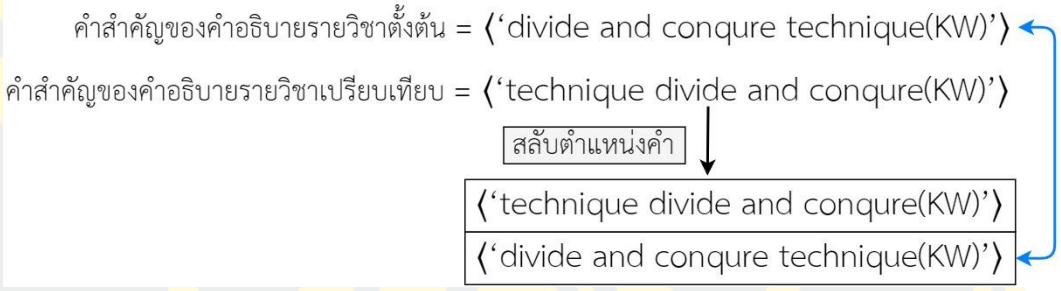
3.3.1 วิธีการเปรียบเทียบแบบตรงตัว (Exact Matching)

การเปรียบเทียบแบบตรงตัวจะทำการเปรียบเทียบความเหมือนกันของตัวอักษรและลำดับการเกิดขึ้นของตัวอักษร (S. Wang & Jiang, 2016) ระหว่างคำสำคัญทั้ง 2 คำจากคำอธิบายรายวิชาตั้งต้นและคำอธิบายเปรียบเทียบ นอกจากนี้ผู้วิจัยยังได้ผสมผสานวิธีการเปรียบเทียบแบบวลี (Paraphrase Matching) (Zhang, Baldridge, & He, 2019) ร่วมกับวิธีการเปรียบเทียบแบบตรงตัวด้วย กล่าวคือ เป็นวิธีการเปรียบเทียบแบบการสลับตำแหน่งคำที่อยู่ในคำสำคัญที่นำมาเปรียบเทียบ โดยจะทำการวนสลับตำแหน่งในทุก ๆ ตำแหน่งของคำที่อยู่ในคำสำคัญจนครบ แล้วนำรายการของคำสำคัญที่ได้จากการสลับตำแหน่ง มาดำเนินการเปรียบเทียบกับคำสำคัญของคำอธิบาย

รายวิชาตั้งต้น โดยตัวอย่างของวิธีการเปรียบเทียบแบบตรงตัวและแบบวลีจะแสดงดังภาพที่ 17 และภาพที่ 18 ตามลำดับ



ภาพที่ 17 ตัวอย่างวิธีการเปรียบเทียบแบบตรงตัวระหว่าง 2 คำสำคัญ



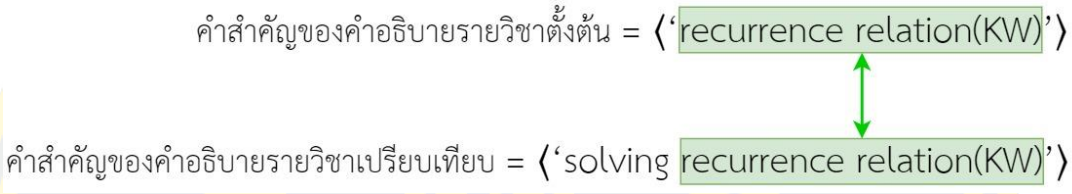
ภาพที่ 18 ตัวอย่างวิธีการเปรียบเทียบแบบวลีระหว่าง 2 คำสำคัญ

จากภาพที่ 17 แสดงให้เห็นถึงการดำเนินการเปรียบเทียบโดยวิธีการเปรียบเทียบแบบตรงตัวระหว่างคำสำคัญของคำอธิบายรายวิชาตั้งต้น คือ ⟨'algorithm (TE)'⟩ และคำสำคัญของคำอธิบายรายวิชาเปรียบเทียบ คือ ⟨'algorithm (TE)'⟩ เมื่อพิจารณาจะพบว่าตัวอักษรและลำดับการเกิดขึ้นของตัวอักษรของทั้ง 2 คำสำคัญ ที่นำมาเปรียบเทียบกันมีความเหมือนกัน ดังนั้นจากการเปรียบเทียบแบบตรงตัวจึงสามารถสรุปได้ว่าคำสำคัญทั้งสองเหมือนกัน และจากภาพที่ 18 แสดงให้เห็นถึงการดำเนินการเปรียบเทียบโดยวิธีการเปรียบเทียบแบบวลีระหว่างคำสำคัญของคำอธิบายรายวิชาตั้งต้น คือ ⟨'divide and conquer technique(KW)'⟩ และคำสำคัญของคำอธิบายรายวิชาเปรียบเทียบ คือ ⟨'technique divide and conquer(KW)'⟩ โดยเริ่มจากการสลับตำแหน่งคำในคำสำคัญของคำอธิบายรายวิชาเปรียบเทียบ ซึ่งเมื่อทำการสลับตำแหน่งครบทุกกรณีแล้วจะได้รายการของคำสำคัญจากการสลับตำแหน่ง ได้แก่ 1) ⟨'technique divide and conquer(KW)'⟩ และ 2) ⟨'divide and conquer technique(KW)'⟩ จากนั้นนำคำสำคัญแต่ละคำที่ได้จากการสลับตำแหน่งมาเปรียบเทียบกับคำสำคัญของคำอธิบายรายวิชาตั้งต้น จึงพบว่าคำสำคัญที่ได้จากการสลับตำแหน่งของคำอธิบายรายวิชาที่นำมาเปรียบเทียบ คือ ⟨'divide and conquer technique(KW)'⟩ เหมือนกันกับคำสำคัญ

ของคำอธิบายรายวิชาตั้งต้น ดังนั้นจากการดำเนินการเปรียบเทียบแบบวลีจึงสามารถสรุปได้ว่าคำสำคัญทั้งสองเหมือนกัน

3.3.2 วิธีการเปรียบเทียบแบบเซตย่อย (Subset matching)

เป็นการเปรียบเทียบโดยพิจารณาว่าคำสำคัญจากคำอธิบายรายวิชาตั้งต้นเป็นเซตย่อยของคำสำคัญจากคำอธิบายรายวิชาเปรียบเทียบหรือไม่ หากเงื่อนไขดังกล่าวเป็นจริงจะสามารถสรุปได้ว่าคำสำคัญทั้ง 2 คำเหมือนกัน โดยตัวอย่างของวิธีการเปรียบเทียบแบบเซตย่อยจะแสดงดังภาพที่ 19

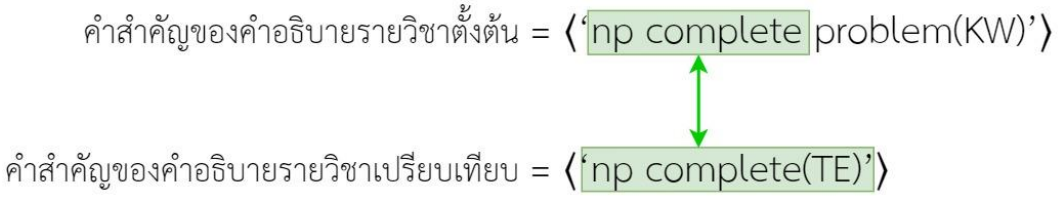


ภาพที่ 19 ตัวอย่างวิธีการเปรียบเทียบแบบเซตย่อยระหว่างคำสำคัญ

ภาพที่ 19 จะแสดงการเปรียบเทียบคำสำคัญจากคำอธิบายรายวิชาตั้งต้น คือ ⟨'recurrence relation(KW)'⟩ และสำคัญจากคำอธิบายรายวิชาเปรียบเทียบ คือ ⟨'solving recurrence relation(KW)'⟩ เมื่อพิจารณาจะพบว่าสำคัญจากคำอธิบายรายวิชาตั้งต้นเป็นเซตย่อยของคำสำคัญจากคำอธิบายรายวิชาเปรียบเทียบ ดังนั้น จึงสามารถสรุปได้ว่าสำคัญของคำอธิบายรายวิชาตั้งต้น คือ ⟨'recurrence relation(KW)'⟩ มีความเหมือนกันกับสำคัญของคำอธิบายรายวิชาเปรียบเทียบ คือ ⟨'solving recurrence relation(KW)'⟩

3.3.3 วิธีการเปรียบเทียบแบบซูเปอร์เซต (Superset matching)

เป็นการเปรียบเทียบโดยพิจารณาว่าคำสำคัญจากคำอธิบายรายวิชาตั้งต้นมีคำสำคัญจากคำอธิบายรายวิชาเปรียบเทียบเป็นสมาชิกหรือไม่ (หรือกล่าวอีกนัยหนึ่งคือ พิจารณาว่าคำสำคัญจากคำอธิบายรายวิชาเปรียบเทียบเป็นเซตย่อยของคำสำคัญจากคำอธิบายรายวิชาตั้งต้นหรือไม่) หากเงื่อนไขดังกล่าวเป็นจริง จะสามารถสรุปได้ว่าคำสำคัญที่นำมาเปรียบเทียบกันมีความเหมือนกัน โดยตัวอย่างของวิธีการเปรียบเทียบแบบซูเปอร์เซตจะแสดงดังภาพที่ 20



ภาพที่ 20 ตัวอย่างวิธีการเปรียบเทียบแบบซูเปอร์เซตระหว่างคำสำคัญ

จากภาพที่ 20 แสดงการเปรียบเทียบระหว่างคำสำคัญของคำอธิบายรายวิชาตั้งต้น คือ ⟨‘np complete problem(KW)’⟩ และสำคัญของคำอธิบายรายวิชาเปรียบเทียบ คือ ⟨‘np complete(TE)’⟩ เมื่อพิจารณาจะพบว่าสำคัญของคำอธิบายรายวิชาตั้งต้น มีคำสำคัญของคำอธิบายรายวิชาเปรียบเทียบเป็นสมาชิก ดังนั้นจึงสามารถสรุปได้ว่าสำคัญของคำอธิบายรายวิชาตั้งต้น คือ ⟨‘np complete problem(KW)’⟩ มีความเหมือนกันกับสำคัญของคำอธิบายรายวิชาเปรียบเทียบ คือ ⟨‘np complete(TE)’⟩

วิธีการเปรียบเทียบทั้ง 3 วิธีการที่กล่าวมาข้างต้น สามารถนิยามได้ดังต่อไปนี้

$$sim(w_i, w_j) = \begin{cases} 1 & , w_i = w_j, w_i \subset w_j \; or \; w_i \supset w_j \\ 0 & , otherwise \end{cases} \quad (1)$$

โดยที่ w_i คือ คำสำคัญในหัวข้อย่อยหนึ่ง ๆ ของคำอธิบายรายวิชาตั้งต้น และ w_j คือ คำสำคัญในหัวข้อย่อยหนึ่ง ๆ ของคำอธิบายรายวิชาเปรียบเทียบ หากคำสำคัญของคำอธิบายรายวิชาตั้งต้นเหมือนกันกับคำสำคัญของคำอธิบายรายวิชาเปรียบเทียบ สามารถสรุปได้ว่าหัวข้อย่อยของคำอธิบายรายวิชาตั้งต้นเหมือนกับหัวข้อย่อยของคำอธิบายรายวิชาเปรียบเทียบ และกำหนดให้มีค่าความเหมือนเท่ากับ 1 ในทางกลับกัน ถ้าคำสำคัญของคำอธิบายรายวิชาตั้งต้นไม่เหมือนกับคำสำคัญของคำอธิบายรายวิชาเปรียบเทียบ สามารถสรุปได้ว่าหัวข้อย่อยของคำอธิบายรายวิชาตั้งต้นไม่เหมือนกับหัวข้อย่อยของคำอธิบายรายวิชาเปรียบเทียบ และกำหนดให้มีค่าความเหมือนเท่ากับ 0 ซึ่งจากการนิยามการให้คะแนนข้างต้นสามารถนิยามได้ดังต่อไปนี้

$$match(tp_p) = \begin{cases} 1 & , \exists w_i \in KV_{tp_p} | sim(w_i, w_j) = 1 \\ & where \; w_j \in KV_{tp_q} \; of \; tp_q \wedge tp_q \in c_{s,y} \\ 0 & , otherwise \end{cases} \quad (2)$$

โดยที่ tp_p คือ หัวข้อย่อยของคำอธิบายรายวิชาตั้งต้น และ tp_q คือ หัวข้อย่อยของคำอธิบายรายวิชาเปรียบเทียบ หลังจากการกำหนดค่าความเหมือนให้กับทุกหัวข้อย่อยของรายวิชาตั้งต้นแล้ว สามารถนำค่าที่ได้มาคำนวณหาอัตราร้อยละของการเปรียบเทียบ ได้ดังสมการต่อไปนี้

$$sim(c_{s,x}, c_{s,y}) = \frac{\sum_{i=1}^{n} match(tp_i)}{n} \quad (3)$$

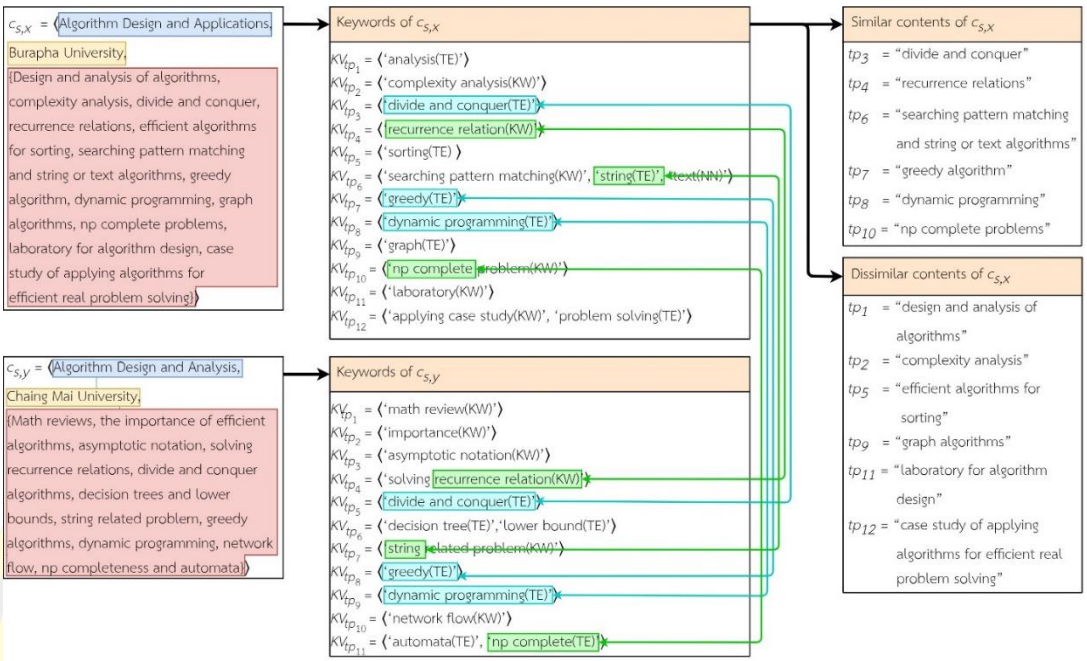
โดยที่ n คือ จำนวนหัวข้อย่อยทั้งหมดในคำอธิบายรายวิชาตั้งต้น

ภาพที่ 21 จะแสดงตัวอย่างการเปรียบเทียบคำอธิบายรายวิชาของวิชา “Algorithm design and applications” ระหว่างคำอธิบายรายวิชาของมหาวิทยาลัยบูรพา (กำหนดให้เป็นคำอธิบายรายวิชาตั้งต้น) และคำอธิบายรายวิชาของมหาวิทยาลัยเชียงใหม่ (กำหนดให้เป็นคำอธิบายรายวิชาเปรียบเทียบ) ของระบบ *CSCDA* จากภาพเมื่อพิจารณาผลลัพธ์จากการเปรียบเทียบจะพบว่าคำสำคัญจากหัวข้อย่อยที่ 3 ของคำอธิบายรายวิชาตั้งต้น คือ ⟨‘divide and conquer(TE)’⟩ เหมือนกับคำสำคัญของคำอธิบายรายวิชาเปรียบเทียบในหัวข้อย่อยที่ 5 คือ ⟨‘divide and conquer(TE)’⟩ ด้วยวิธีการเปรียบเทียบแบบตรงตัว แล้วยังพบว่าคำสำคัญจากหัวข้อย่อยที่ 7 คือ ⟨‘greedy(TE)’⟩ และในหัวข้อย่อยที่ 8 คือ ⟨‘dynamic programming(TE)’⟩ ของคำอธิบายรายวิชาตั้งต้น เหมือนกันกับคำสำคัญในหัวย่อยข้อที่ 8 และ 9 ของคำอธิบายรายวิชาเปรียบเทียบด้วยวิธีการเปรียบเทียบแบบตรงตัวเช่นเดียวกัน นอกจากนี้ ด้วยวิธีการเปรียบเทียบแบบเซตย่อยยังพบว่าคำสำคัญจากหัวข้อย่อยที่ 4 ของคำอธิบายรายวิชาตั้งต้น คือ ⟨‘recurrence relation(TE)’⟩ เหมือนกับคำสำคัญจากหัวข้อย่อยที่ 4 ของคำอธิบายรายวิชาเปรียบเทียบ คือ ⟨‘solving recurrence relation(KW)’⟩ และ คำสำคัญจากหัวข้อย่อยที่ 6 ของคำอธิบายรายวิชาตั้งต้น คือ ⟨‘string(TE)’⟩ เหมือนกับคำสำคัญจากหัวข้อย่อยที่ 4 ของคำอธิบายรายวิชาเปรียบเทียบ คือ ⟨‘string related problem(KW)’⟩ ท้ายสุด ด้วยวิธีการเปรียบเทียบแบบซูเปอร์เซตพบว่าคำสำคัญจากหัวข้อย่อยที่ 10 ของคำอธิบายรายวิชาตั้งต้น คือ ⟨‘np complete problem(KW)’⟩ เหมือนกับคำสำคัญจากหัวข้อย่อยที่ 11 ของคำอธิบายรายวิชาเปรียบเทียบ คือ ⟨‘np complete(TE)’⟩

จากการดำเนินการเปรียบเทียบระหว่างคำสำคัญในแต่ละหัวข้อย่อยของคำอธิบายรายวิชา “Algorithm design and applications” ของมหาวิทยาลัยบูรพาและมหาวิทยาลัยเชียงใหม่ ทำให้ได้มาซึ่งส่วนของเนื้อหาของคำอธิบายรายวิชาตั้งต้นที่เหมือนกับคำอธิบายรายวิชาเปรียบเทียบ ได้แก่

หัวข้อย่อยที่ 3 ของคำอธิบายรายวิชาตั้งต้น คือ “divide and conquer”, หัวข้อย่อยที่ 4 ของคำอธิบายรายวิชาตั้งต้น คือ “recurrence relations”, หัวข้อย่อยที่ 6 ของคำอธิบายรายวิชาตั้งต้น คือ “searching pattern matching and string or text algorithms”, หัวข้อย่อยที่ 7 ของคำอธิบายรายวิชาตั้งต้น คือ “greedy algorithm”, หัวข้อย่อยที่ 8 ของคำอธิบายรายวิชาตั้งต้น คือ “dynamic programming”, และ หัวข้อย่อยที่ 10 ของคำอธิบายรายวิชาตั้งต้น คือ “dynamic programming” โดยเมื่อทำการคำนวณอัตราร้อยละของความเหมือนกันของคำอธิบายรายวิชาตั้งต้นมีค่าเท่ากับ 50% นอกจากนี้ยังได้ส่วนของเนื้อหาที่แตกต่างกัน ได้แก่ หัวข้อย่อยที่ 1 ของคำอธิบายรายวิชาตั้งต้น คือ “design and analysis of algorithms”, หัวข้อย่อยที่ 2 ของคำอธิบายรายวิชาตั้งต้น คือ “complexity analysis”, หัวข้อย่อยที่ 5 ของคำอธิบายรายวิชาตั้งต้น คือ “efficient algorithms for sorting”, หัวข้อย่อยที่ 9 ของคำอธิบายรายวิชาตั้งต้น คือ “graph algorithms”, หัวข้อย่อยที่ 11 ของคำอธิบายรายวิชาตั้งต้น คือ “laboratory for algorithm design”, และ หัวข้อย่อยที่ 12 ของคำอธิบายรายวิชาตั้งต้น คือ “case study of applying algorithms for efficient real problem solving”

จากการดำเนินการในทุกขั้นตอนของระบบ *CSCDA* ทำให้ได้มาซึ่งผลลัพธ์ของเนื้อหาที่มีความเหมือนและแตกต่างกันของคำอธิบายรายวิชาตั้งต้นกับคำอธิบายรายวิชาเปรียบเทียบ จากผลลัพธ์ที่ได้สามารถเป็นส่วนช่วยในการตัดสินใจให้กับคณะกรรมการหรืออาจารย์ผู้สอนในรายวิชานั้น ๆ ในด้านของการแก้ไขและปรับปรุงคำเนื้อหาของอธิบายรายวิชาให้มีความเหมาะสมและทันสมัยมากยิ่งขึ้น เพื่อทำให้ผู้เรียนที่ได้รับการถ่ายทอดจากรายวิชานั้น ๆ ได้รับองค์ความรู้ที่เหมาะสมและมีประสิทธิภาพ

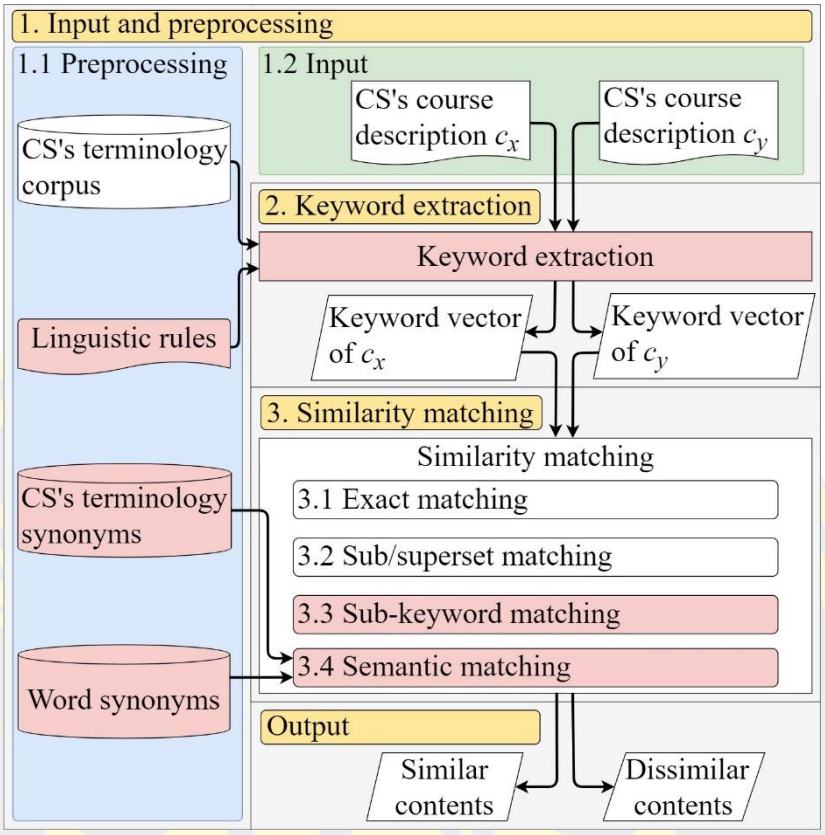


ภาพที่ 21 ตัวอย่างการเปรียบเทียบคำอธิบายรายวิชาของวิชา "Algorithm Design and Applications"

บทที่ 4
ระบบวิเคราะห์คำอธิบายรายวิชาในหลักสูตรวิทยาการคอมพิวเตอร์ที่มีประสิทธิภาพ

ระบบ *CSCDA* ที่นำเสนอในส่วนของบทที่ 3 สามารถวิเคราะห์ส่วนของเนื้อหาในคำอธิบายรายวิชาของคำอธิบายรายวิชาตั้งต้นที่เหมือนและแตกต่างกับเนื้อหาของคำอธิบายรายวิชาเปรียบเทียบ ซึ่งเมื่อทำการพิจารณาการดำเนินงานของระบบ *CSCDA* จะพบว่าในส่วนของวิธีการเปรียบเทียบทั้ง 3 วิธีการได้แก่ 1) วิธีการเปรียบเทียบแบบตรงตัว (Exact Matching), 2) วิธีการเปรียบเทียบแบบเซตย่อย (Subset matching), และ 3) วิธีการเปรียบเทียบแบบซูเปอร์เซต (Superset matching) เป็นวิธีการเปรียบเทียบจากความเหมือนกันระหว่างตัวอักษร และลำดับการเกิดขึ้นของตัวอักษรในคำสำคัญที่นำมาเปรียบเทียบกันเท่านั้น แต่อย่างไรก็ตาม คำสำคัญที่ปรากฏในคำอธิบายรายวิชาอาจมีการเขียนที่แตกต่างกัน แต่ให้ความหมายที่เหมือนกันหรือคล้ายคลึงกัน อาทิ เช่น คำสำคัญคำว่า ‘principle’ และ ‘theory’ เมื่อดำเนินการเปรียบเทียบโดยระบบ *CSCDA* จะพบว่าคำสำคัญทั้ง 2 คำแตกต่างกัน เนื่องจากทั้ง 2 คำมีชุดของตัวอักษรที่แตกต่างกันอย่างสิ้นเชิง แต่เมื่อทำการพิจารณาถึงความหมายของทั้ง 2 คำพบว่าคำสำคัญทั้ง 2 คำมีความหมายเหมือนกันและควรถูกพิจารณาว่าเหมือนกัน จากการที่ระบบ *CSCDA* ไม่สามารถดำเนินการเปรียบเทียบเชิงความหมายระหว่างคำสำคัญได้ ทำให้เนื้อหาในส่วนที่เหมือนกันยังคงมีความไม่ครอบคลุมอยู่

ดังนั้น จากปัญหาข้างต้นผู้วิจัยจึงทำการพัฒนาระบบ *CSCDA* ให้มีประสิทธิภาพมากยิ่ง โดยระบบที่ได้ทำการพัฒนาต่อยอดจากระบบ *CSCDA* คือ “ระบบวิเคราะห์คำอธิบายรายวิชาในหลักสูตรวิทยาการคอมพิวเตอร์ที่มีประสิทธิภาพ *(An efficient Computer Science Course Description Analysis system)*” หรือเรียกว่า “ระบบ *eCSCDA*” ซึ่งมีขั้นตอนการดำเนินงาน 3 ขั้นตอนเหมือนกับระบบ *CSCDA* ได้แก่ 1) การเตรียมข้อมูลนำเข้าและการประมวลผลข้อมูลเบื้องต้น (Input and preprocessing) 2) การสกัดคำสำคัญจากคำอธิบายรายวิชา (Keyword extraction) และ 3) การเปรียบเทียบคำสำคัญระหว่างคำอธิบายรายวิชา (Similarity Matching) โดยที่โครงสร้างการดำเนินงานของระบบ *eCSCDA* จะแสดงดังภาพที่ 22 และรายละเอียดของขั้นตอนการดำเนินงานในแต่ละขั้นตอนที่ถูกปรับปรุงและพัฒนาขึ้น จะแสดงภาพรวมในตารางที่ 3



ภาพที่ 22 โครงสร้างของระบบ eCSCDA

ตารางที่ 3 รายการการปรับปรุงและพัฒนาขั้นตอนการดำเนินงานของระบบ eCSCDA

ขั้นตอนการดำเนินงาน	**ระบบ *CSCDA***	**ระบบ *eCSCDA***
1. การเตรียมข้อมูลนำเข้า	1. การรวบรวมคำอธิบายรายวิชา 2. การรวบรวมคำศัพท์เฉพาะ 3. การรวบรวมกฎทางภาษาศาสตร์	1. การเตรียมข้อมูลคลังคำศัพท์เฉพาะ **2. การเตรียมข้อมูลกฎทางภาษาศาสตร์ฉบับปรับปรุง** **3. การเตรียมข้อมูลคลังคำพ้องความหมายของคำศัพท์เฉพาะ** **4. การเตรียมข้อมูลคลังคำพ้องความหมายของคำศัพท์ทั่วไป** 5. การเตรียมข้อมูลคำอธิบายรายวิชา

ตารางที่ 3 รายการการปรับปรุงและพัฒนาขั้นตอนการดำเนินงานของระบบ eCSCDA (ต่อ)

ขั้นตอนการดำเนินงาน	**ระบบ *CSCDA***	**ระบบ *eCSCDA***
2. การสกัดคำสำคัญจากคำอธิบายรายวิชา	1. การประมวลผลข้อความเบื้องต้น 1.1 การแปลงตัวอักษรพิมพ์ใหญ่ให้เป็นตัวพิมพ์เล็ก 1.2 การแก้ไขคำผิด 1.3 การกำจัดคำหยุด 1.4 การระบุหน้าที่ของคำ 2. การระบุคำศัพท์เฉพาะ 2.1 การระบุคำศัพท์เฉพาะ (ครั้งที่ 1) 2.2 การแปลงรูปคำให้อยู่ในรากศัพท์ 2.3 การระบุคำศัพท์เฉพาะ (ครั้งที่ 2) 3. การสกัดคำสำคัญ 3.1 การระบุคำสำคัญโดยกฎทางภาษาศาสตร์ 4. การลดทอนเนื้อหาที่ไม่สำคัญ	1. การประมวลผลข้อความเบื้องต้น **1.1 การแบ่งประโยค** **1.2 การแบ่งคำ** 1.3 การแปลงตัวอักษรพิมพ์ใหญ่ให้เป็นตัวพิมพ์เล็ก 1.4 การแก้ไขคำผิด 1.5 การกำจัดคำหยุด 1.6 การระบุหน้าที่ของคำ 2. การระบุคำศัพท์เฉพาะ 2.1 การระบุคำศัพท์เฉพาะ (ครั้งที่ 1) 2.2 การแปลงรูปคำให้อยู่ในรากศัพท์ 2.3 การระบุคำศัพท์เฉพาะ (ครั้งที่ 2) 3. การสกัดคำสำคัญ **3.1 การระบุคำสำคัญโดยกฎทางภาษาศาสตร์ฉบับปรับปรุง**

ตารางที่ 3 รายการการปรับปรุงและพัฒนาขั้นตอนการดำเนินงานของระบบ eCSCDA (ต่อ)

ขั้นตอนการดำเนินงาน	ระบบ *CSCDA*	ระบบ *eCSCDA*
3. การเปรียบเทียบคำสำคัญระหว่างคำอธิบายรายวิชา	1. วิธีการเปรียบเทียบแบบตรงตัว (Exact Matching) 2. วิธีการเปรียบเทียบแบบเซตย่อย (Subset matching) 3. วิธีการเปรียบเทียบแบบซูเปอร์เซต (Superset matching)	1. วิธีการเปรียบเทียบแบบตรงตัว (Exact Matching) **2. วิธีการเปรียบเทียบแบบเซตย่อย/ซูเปอร์เซต (Sub/superset matching)** **3. วิธีการเปรียบเทียบแบบองค์ประกอบร่วม (Sub-keyword matching)** **4. วิธีการเปรียบเทียบเชิงความหมาย (Semantic matching)**

4.1 การเตรียมข้อมูลนำเข้าและการประมวลผลข้อมูลเบื้องต้น

การเตรียมข้อมูลนำเข้าและการประมวลผลข้อมูลเบื้องต้น สามารถแบ่งการทำงานออกเป็น 5 ส่วน โดยมีรายละเอียดดังนี้

4.1.1 การเตรียมข้อมูลคลังคำศัพท์เฉพาะ

ระบบ *eCSCDA* ได้ทำการสร้างคลังคำศัพท์เฉพาะในศาสตร์ทางด้านวิทยาการคอมพิวเตอร์เช่นเดียวกับระบบ *CSCDA* โดยจะทำการรวบรวมคำศัพท์เฉพาะจาก 8 แหล่ง ได้แก่ 1) A Dictionary of Computer Science, 2) Computer dictionary, 3) Computer Science Glossary, 4) Glossary of Computer Related Terms, 5) Glossary of computer science, 6) Labautopedia, 7) PC Glossary, และ 8) Techtarget ซึ่งทำให้สามารถนำมาสร้างเป็นคลังคำศัพท์เฉพาะที่มีจำนวนคำศัพท์เฉพาะทั้งสิ้น 28,392 คำ โดยระบบ *eCSCDA* จะทำการประยุกต์ใช้คลังคำศัพท์เฉพาะนี้ เข้ามาช่วยในการระบุถึงคำศัพท์เฉพาะที่ปรากฏอยู่ในเนื้อหาของคำอธิบายรายวิชา เพื่อทำให้การสกัดคำสำคัญมีประสิทธิภาพ

4.1.2 การเตรียมข้อมูลกฎทางภาษาศาสตร์

ในส่วนนี้จะเป็นการเตรียมข้อมูลกฎทางภาษาที่ถูกปรังปรุงและพัฒนาขึ้นมาใหม่โดยระบบ SBS+ *(An efficient Supplementary Book Suggestion system)* (Chaisoongnoen & Amphawan, 2020) เพื่อนำเข้ามาช่วยในการสกัดคำสำคัญจากเนื้อหาในแต่ละหัวข้อย่อยของคำอธิบายรายวิชาหนึ่ง ๆ ซึ่งกฎทางภาษาศาสตร์ที่ถูกปรับปรุงและพัฒนาขึ้นมาใหม่จะถูกเรียกว่า “กฎทางภาษาศาสตร์ฉบับปรับปรุง” ซึ่งเป็นกฎที่จะทำการพิจารณาถึงคำบุพบท (Preposition) อาทิเช่น คำว่า ‘for’ ‘in’ ‘of’ ‘on’ เป็นต้น หรือคำสันธาน (Conjunction) อาทิเช่น คำว่า ‘and’ ‘or’ เป็นต้น ที่ปรากฏอยู่ในเนื้อหาของแต่ละหัวข้อย่อยเป็นอันดับแรก จากนั้นทำการแยกกลุ่มคำที่ปรากฏขึ้นข้างหน้าและข้างหลังคำสันธานหรือคำบุพบทออกจากกัน แล้วทำการตรวจสอบคำสันธานหรือคำบุพบทในแต่ละกลุ่มอีกรอบ ถ้าพบว่ามีคำสันธานหรือคำบุพบทปรากฏอยู่ในกลุ่มที่แยก จะทำการแยกกลุ่มย่อยของกลุ่มนั้น ๆ ไปเรื่อย ๆ แต่ถ้าเมื่อไม่พบคำสันธานหรือคำบุพบทในกลุ่มที่แยกแล้ว จะทำการประยุกต์ใช้กฎทางภาษาศาสตร์ฉบับปรับปรุงในแต่ละกลุ่มที่ถูกแยก สุดท้ายทำการลบคำคุณศัพท์บางคำที่ไม่ได้บ่งบอกถึงใจความสำคัญออกจากคำสำคัญที่สกัดได้ โดยมีตัวอย่างของการสกัดคำสำคัญด้วยกฎทางภาษาศาสตร์ฉบับปรับปรุง ดังนี้ การสกัดคำสำคัญจากหัวข้อย่อย “(‘experiment’, ‘JJ’), (‘design’, ‘TE’), (‘and’, ‘CC’), (‘hypothesis’, ‘TE’), (‘testing’, ‘TE’)” เมื่อตรวจสอบจะพบคำว่า (‘and’, ‘CC’) เป็นคำบุพบทที่ปรากฏขึ้น ดังนั้นจึงทำการแบ่งกลุ่มคำที่อยู่ข้างหน้าและหลังคำบุพบทออกเป็น 2 กลุ่ม คือ 1) “(‘experiment’, ‘JJ’), (‘design’, ‘TE’)” และ 2) “(‘hypothesis’, ‘TE’), (‘testing’, ‘TE’)” จากการแบ่งกลุ่มคำทั้ง 2 ทำให้โครงสร้างของหัวข้อย่อยนี้อยู่ในรูปของ (Phrase1) + CC + (Phrase2) ซึ่งมีโครงสร้างตรงกับกฎฉบับปรับปรุง คือ กฎ “(Phrase1) + CC + (Phrase2)” $\longrightarrow$ ⟨(Phrase1), (Phrase2)⟩ จากการสกัดคำสำคัญด้วยกฎข้างต้น ทำให้ได้มาซึ่งคำสำคัญ คือ ⟨‘experiment(NN) design(TE)’ และ ‘hypothesis(TE) testing(TE)’⟩ จากนั้นนำคำสำคัญที่สกัดได้แต่ละคำมาตรวจสอบถึงคำคุณศัพท์ที่ไม่ได้บ่งบอกถึงใจความสำคัญ ซึ่งจากการตรวจสอบไม่พบคำคุณศัพท์ปรากฏอยู่ในคำสำคัญทั้ง 2 ทำให้คำสำคัญทั้ง 2 ยังคงเดิม

จากการที่กฎทางภาษาศาสตร์ฉบับปรับปรุงได้ทำการพัฒนาและปรับปรุงขึ้นมาใหม่ ทำให้มีความแตกต่างจากกฎทางภาษาศาสตร์แบบเดิม ในแง่มุมของ

1) กฎทางภาษาศาสตร์ฉบับปรับปรุงจะไม่มีการแบ่งกฎที่ใช้สำหรับการพิจารณร่วมกับคำศัพท์เฉพาะ และกฎที่ไม่ได้พิจารณาร่วมกับคำศัพท์เฉพาะ เนื่องจากกฎฉบับปรับปรุงจะทำการพิจารณาถึงคำสันธานหรือคำบุพบทในหัวข้อย่อยก่อนเป็นอันดับแรก เมื่อพบว่ามี

คำสันธานหรือคำบุพบทปรากฏอยู่จะทำการแยกกลุ่มคำที่ปรากฏขึ้นข้างหน้าและข้างหลังคำสันธานหรือคำบุพบทออกจากการ จากนั้นจึงทำการประยุกต์ใช้กฎฉบับปรับปรุง เพื่อทำการรวมกลุ่มคำทั้ง 2 กลุ่มเข้าด้วยกัน ทำให้ได้มาซึ่งคำสำคัญของหัวข้อย่อยนั้น ๆ

2) คำสำคัญที่ได้จากกฎทางภาษาศาสตร์ฉบับปรับปรุงจะไม่มีความซ้ำซ้อนกันเกิดขึ้น เนื่องจากการสกัดคำสำคัญด้วยกฎเดิมในกฎประเภทที่ 1 คือ กฎที่พิจารณาร่วมกับคำศัพท์เฉพาะ เมื่อพบคำสันธานหรือคำบุพบทจะทำการสกัดกลุ่มคำที่ปรากฏอยู่ข้างหน้าและข้างหลังออกมาเป็นคำสำคัญที่สกัดได้ อีกทั้งยังได้คำสำคัญที่เกิดจากการรวมกลุ่มของกลุ่มคำทั้ง 2 กลุ่มเข้าด้วยกันอีกด้วย ในทางกลับกัน กฎฉบับปรับปรุงจะไม่ทำการสกัดกลุ่มคำที่ถูกแยกออกจากกันเมื่อตรวจสอบพบคำสันธานหรือคำบุพบท แต่จะทำการรวมกลุ่มคำทั้งสองกลุ่มโดยการใช้กฎฉบับปรับปรุงแล้วจึงสกัดคำสำคัญออกมา อาทิเช่น การสกัดคำสำคัญจากหัวข้อย่อย “(‘data analysis’, ‘TE’), (‘for’, ‘IN’), (‘decision’, ‘NN’), (‘support’, ‘NN’)” เมื่อใช้กฎทางภาษาศาสตร์เดิมในการสกัดคำสำคัญ ทำให้ได้คำสำคัญคือ ⟨‘data analysis(TE)’, ‘decision support(KW)’ และ ‘decision support data analysis(KW)’⟩ จะเห็นว่ามีคำสำคัญที่ซ้ำซ้อนกันเกิดขึ้น คือ คำว่า ‘data analysis(TE)’ และ ‘decision support(KW)’ มีความซ้ำซ้ำกับ ‘decision support data analysis(KW)’ ในทางกลับกันเมื่อใช้กฎทางภาษาศาสตร์ฉบับปรับปรุงจะทำให้ได้มาซึ่งคำสำคัญที่สกัดได้ คือ ⟨‘decision support data analysis(KW)’⟩ และเมื่อตรวจสอบจะไม่พบว่ามีการซ้ำซ้อนกันของคำสำคัญเกิดขึ้น

3) คำสำคัญที่สกัดได้จากกฎทางภาษาศาสตร์ฉบับปรับปรุงจะไม่ปรากฏคำคุณศัพท์ที่ไม่ได้บ่งบอกถึงใจความสำคัญ เนื่องจากคำสำคัญที่สกัดได้ทุกคำจะถูกนำไปพิจารณาคำคุณศัพท์ที่ปรากฏขึ้น โดยการตรวจสอบคำคุณศัพท์ที่ปรากฏขึ้นในแต่ละคำสำคัญกับคำคุณศัพท์ในคลังคำคุณศัพท์ที่ไม่ได้บ่งบอกถึงใจความสำคัญ ซึ่งถ้าพบว่าเป็นคำคุณศัพท์ที่ปรากฏอยู่ในคลัง จะทำการลบคำคุณศัพท์คำนั้นออกจากคำสำคัญ แต่ถ้าเป็นคำคุณศัพท์ที่ไม่ได้ปรากฏอยู่ในคลังจะยังคงคำคุณศัพท์ในคำสำคัญนั้นเช่นเดิม

4.1.3 การเตรียมข้อมูลคลังคำพ้องความหมายของคำศัพท์เฉพาะ

หลังจากทำการรวบรวมคำศัพท์เฉพาะจาก 8 แหล่งข้อมูลคำศัพท์เฉพาะ ในหัวข้อที่ 4.1.1 แล้ว จะทำการนำแต่ละคำศัพท์เฉพาะมาค้นหาคำพ้องความหมายจาก 8 แหล่งข้อมูลคลังคำศัพท์เฉพาะ ซึ่งจากการดำเนินการดังกล่าวจะทำให้คำศัพท์เฉพาะหนึ่ง ๆ อาจมีหลายคำพ้องความหมาย โดยอาจมีบางคำที่มีความพ้องความหมายน้อยปะปนอยู่ด้วย ดังนั้น เพื่อทำให้การพิจารณาคำพ้องความหมายมีความถูกต้องและมีประสิทธิภาพ ผู้วิจัยจึงนำแต่ละคำศัพท์เฉพาะ พร้อมกับคำพ้องความหมายหนึ่ง ๆ มาทำการตรวจสอบกับ 3 พจนานุกรมภาษาอังกฤษออนไลน์ ได้แก่ 1) Longdo Dictionary[10], 2) google translation corpus[11] และ 3) Cambridge Dictionary[12] ตามลำดับ โดยหากมีอย่างน้อย 2 จาก 3 เว็บไซต์ข้างต้น แสดงให้เห็นว่าคำศัพท์เฉพาะมีการพ้องความหมายกับคำพ้องความหมายจริง คำศัพท์เฉพาะและคำพ้องความหมายที่พิจารณาจะถูกเก็บรวบรวมในคลังพ้องความหมายของคำศัพท์เฉพาะ (CS's terminology synonyms)

ภาพที่ 23 แสดงการรวบรวมคำพ้องความหมายของคำศัพท์เฉพาะ “Average” จากคลังคำศัพท์ของ Computer dictionary[13] โดยมีคำพ้องความหมาย ได้แก่ ‘Avg’, ‘Median’, ‘Mode’ และ ‘Number’ จากนั้นนำคำพ้องความหมายที่ได้มาทำการตรวจสอบถึงความเหมาะสมทางการพ้องความหมาย ดังตัวอย่างในภาพที่ 24 ซึ่งจะเป็นการนำคำศัพท์เฉพาะคำว่า “Average” ไปทำการค้นหาคำพ้องความหมายกับพจนานุกรมออนไลน์ทั้ง 3 เว็บไซต์ ได้แก่ 1) Longdo Dictionary, 2) google translation corpus และ 3) Cambridge Dictionary จากนั้นทำการตรวจสอบคำพ้องความหมายทุกคำที่รวบรวมมาได้กับคำพ้องความหมายที่ได้จากพจนานุกรมมออนไลน์ทั้ง 3 เว็บไซต์ เมื่อพิจารณาจะพบคำพ้องความหมายคำว่า ‘Median’ เป็นคำพ้องความหมายที่มีเหมือนกันในทั้ง 3 เว็บไซต์ จึงสามารถสรุปได้ว่าคำว่า ‘Median’ เหมาะสมที่จะเป็นคำพ้องความหมายของคำศัพท์เฉพาะคำว่า “Average” สุดท้ายในภาพที่ 25 เป็นการนำคำศัพท์เฉพาะและคำพ้องความหมายที่ผ่านการตรวจสอบความเหมาะสมแล้วมาสร้างเป็นคลังคำพ้องความหมายของคำศัพท์เฉพาะ เพื่อนำไปใช้งานในขั้นตอนต่อไป

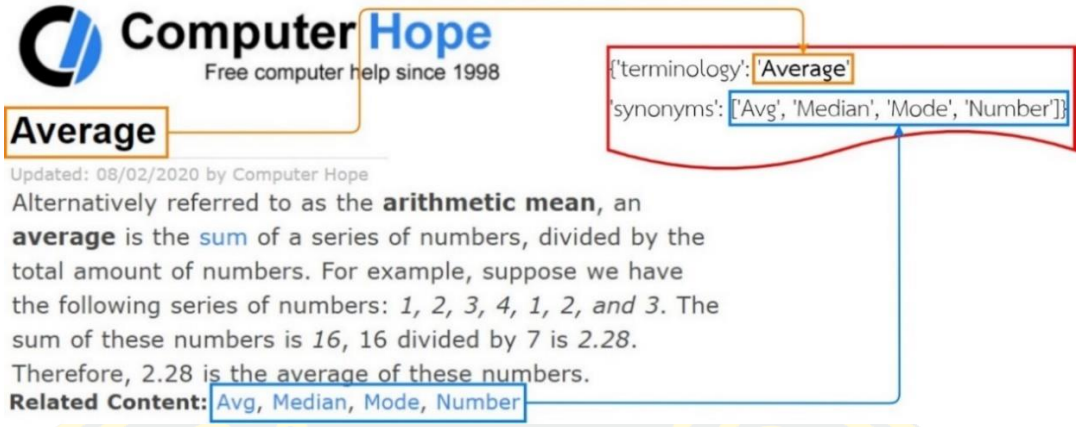
---

[10] https://dict.longdo.com

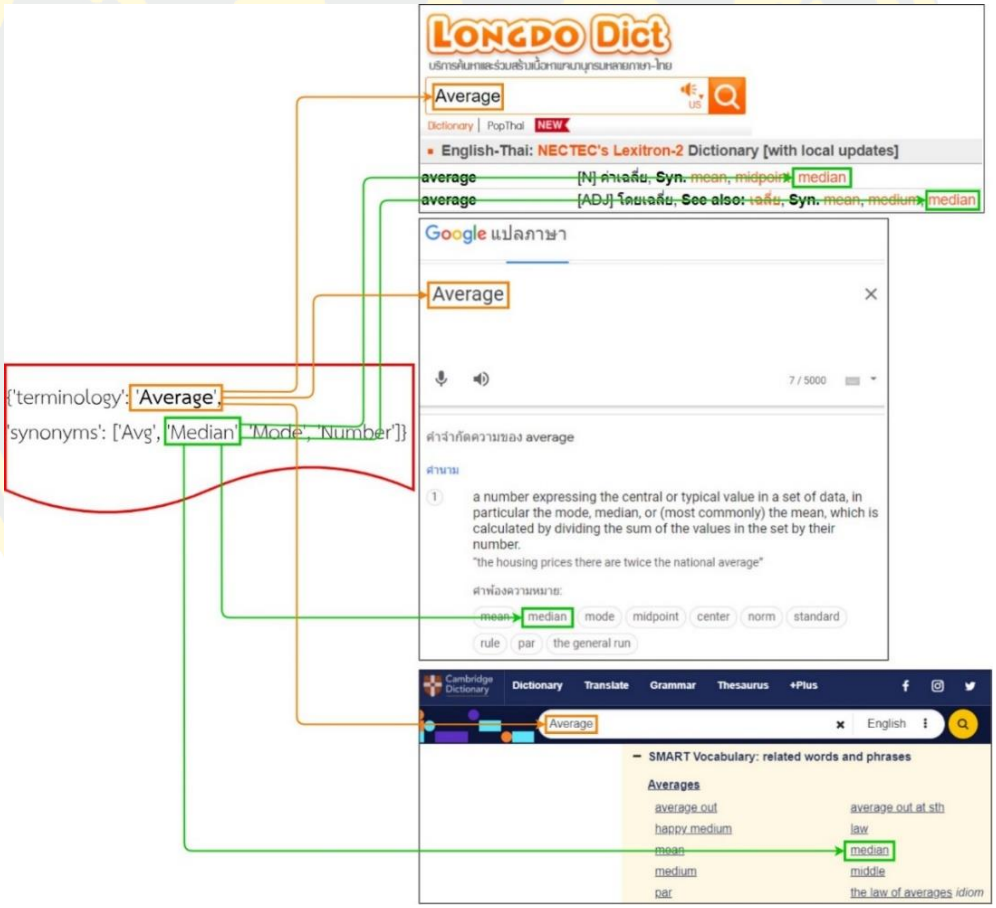
[11] https://translate.google.com

[12] https://dictionary.cambridge.org

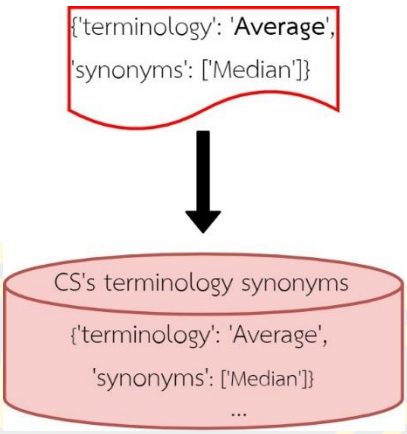
[13] https://www.computerhope.com/jargon.htm



ภาพที่ 23 ตัวอย่างการรวบรวมคำพ้องความหมายของคำศัพท์เฉพาะคำว่า "Average"



ภาพที่ 24 ตัวอย่างการตรวจสอบความเหมาะสมทางการพ้องความหมายของคำพ้องความหมายที่รวบรวมมาได้

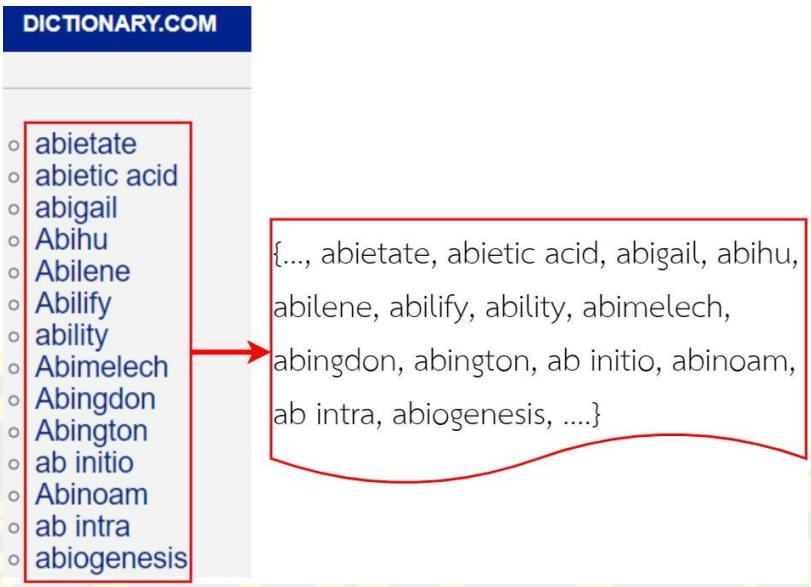


ภาพที่ 25 ตัวอย่างการสร้างคลังคำพ้องความหมายของคำศัพท์เฉพาะ

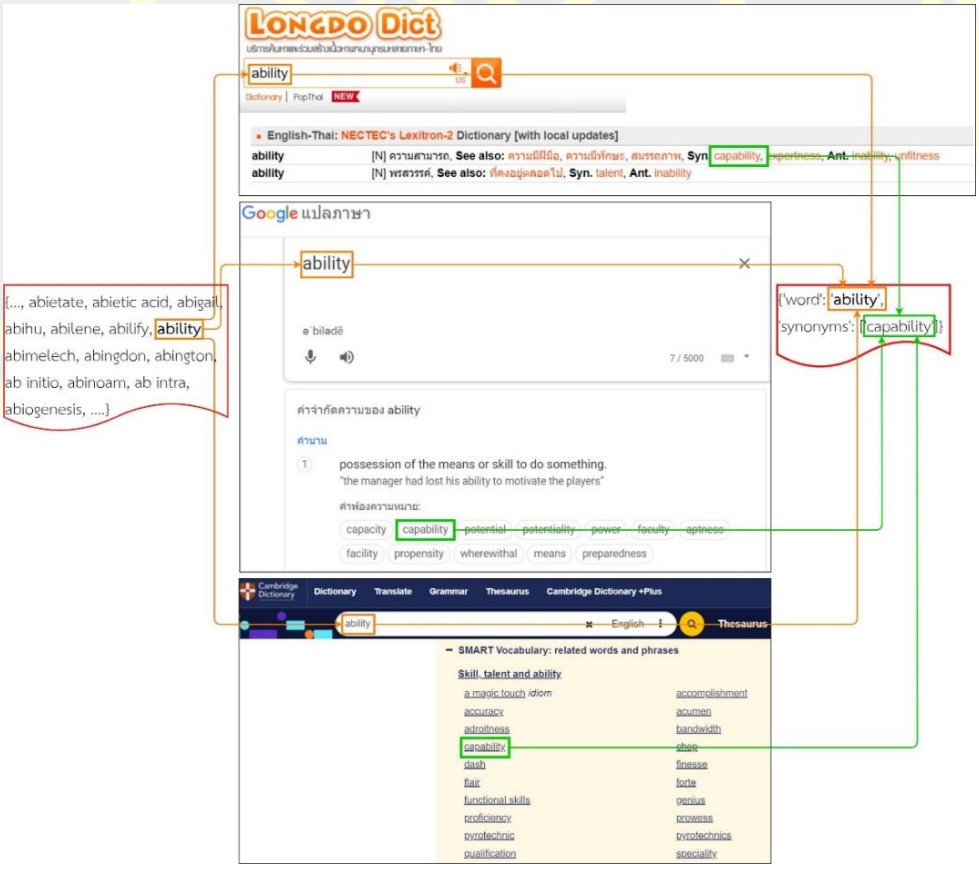
4.1.4 การเตรียมข้อมูลคลังคำพ้องความหมายของคำศัพท์ทั่วไป

การสร้างคลังคำพ้องความหมายของคำศัพท์ทั่วไป จะทำการพิจารณาทุกคำที่ไม่ได้ถูกระบุว่าเป็นคำศัพท์เฉพาะในศาสตร์ทางด้านวิทยาการคอมพิวเตอร์ โดยทำการรวบรวมคำศัพท์ทั่วไปจากเว็บไซต์ www.dictionary.com เป็นจำนวน 133,291 คำ จากนั้นทำการพิจารณาแต่ละคำศัพท์ทั่วไปแล้วทำการค้นหาคำพ้องความหมายจากเว็บไซต์พจนานุกรมภาษาอังกฤษออนไลน์ ได้แก่ 1) Longdo Dictionary, 2) google translation corpus และ 3) Cambridge Dictionary โดยคำพ้องความหมายที่จะถูกพิจารณาว่าเหมาะสมทางการพ้องความหมายกับคำศัพท์คำนั้น ๆ จะต้องเป็นคำพ้องความหมายที่ปรากฏขึ้นอย่างน้อย 2 ใน 3 เว็บไซต์ข้างต้น คำศัพท์ทั่วไปและคำพ้องความหมายที่พิจารณาจะถูกเก็บรวบรวมในคลังพ้องความหมายของคำศัพท์ทั่วไป (Word synonyms)

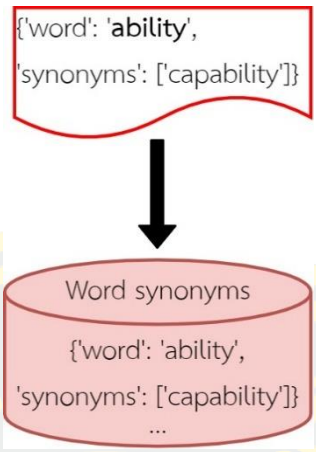
ภาพที่ 26 แสดงการรวบรวมคำศัพท์ทั่วไปจาก www.dictionary.com จากนั้นนำแต่ละคำศัพท์ทั่วไปมาทำการค้นหาคำพ้องความหมายจากพจนานุกรมภาษาอังกฤษออนไลน์ ซึ่งแสดงดังภาพที่ 27 เป็นการค้นหาคำพ้องความหมายของคำศัพท์ทั่วไปคำว่า “ability” กับพจนานุกรมภาษาอังกฤษออนไลน์ที่กล่าวข้างต้น จากนั้นทำการพิจารณาการพ้องความหมายของคำพ้องความหมายที่ปรากฏขึ้นในทั้ง 3 เว็บไซต์ พบว่า ‘capability’ เป็นคำพ้องความหมายที่ปรากฏใน 3 เว็บไซต์ ด้วยเหตุนี้ จึงสามารถสรุปได้ว่าคำศัพท์คำว่า “ability” พ้องความหมายกับคำว่า ‘capability’ สุดท้ายในภาพที่ 28 เป็นการนำคำศัพท์ทั่วไปและคำพ้องความหมายที่ผ่านการตรวจสอบความเหมาะสมแล้วมาสร้างเป็นคลังคำพ้องความหมายของคำศัพท์ทั่วไป เพื่อนำไปใช้งานในขั้นตอนต่อไป



ภาพที่ 26 ตัวอย่างการรวบรวมคำศัพท์ทั่วไปจาก www.dictionary.com



ภาพที่ 27 ตัวอย่างการค้นหาคำพ้องควาหมายและพิจารณาความเหมาะสมของความพ้องความหมาย



ภาพที่ 28 ตัวอย่างการสร้างคลังคำพ้องความหมายของคำศัพท์ทั่วไป

4.1.5 การเตรียมข้อมูลคำอธิบายรายวิชา

ในส่วนนี้เป็นการเตรียมข้อมูลคำอธิบายรายวิชาที่ใช้สำหรับเป็นข้อมูลนำเข้าของระบบ *eCSCDA* จะเป็นการนำข้อมูลคำอธิบายรายวิชาที่ได้จากขั้นตอนการรวบรวมคำอธิบายรายวิชาของระบบ *CSCDA* ในส่วนของหัวข้อที่ 3.1.1 มาเป็นข้อมูลนำเข้า โดยที่คำอธิบายรายวิชาหนึ่ง ๆ จะประกอบไปด้วยข้อมูล 3 ส่วน ซึ่งสามารถจัดเก็บให้อยู่ในรูปแบบ 3-tuple : $< s, u, cc >$ เมื่อ s หมายถึง ชื่อรายวิชา (ภาษาอังกฤษ) u หมายถึง ชื่อมหาวิทยาลัย (ภาษาอังกฤษ) และ cc หมายถึง เนื้อหาของคำอธิบายรายวิชา (ภาษาอังกฤษ) ตามลำดับ ในการเปรียบเทียบระหว่างคำอธิบายรายวิชาตั้งต้น (c_x) และ คำอธิบายรายวิชาเปรียบเทียบ (c_y) ควรที่จะต้องเป็นคำอธิบายที่มาจากรายวิชาเดียวกัน หรือเป็นรายวิชาที่สอดคล้องกันเท่านั้น

4.2 การสกัดคำสำคัญจากคำอธิบายรายวิชา

การสกัดคำสำคัญจากเนื้อหาในคำอธิบายรายวิชาหนึ่ง ๆ ของระบบ *eCSCDA* มีการดำเนินงาน 3 ส่วน ได้แก่ 1) การประมวลผลข้อความเบื้องต้น 2) การระบุถึงคำคำศัพท์เฉพาะ และ 3) การสกัดคำสำคัญ ตามลำดับ โดยในแต่ละขั้นตอนจะมีรายละเอียดดังต่อไปนี้

4.2.1 การประมวลผลข้อความเบื้องต้น (Text preprocessing)

ในส่วนนี้ผู้วิจัยได้ทำการประยุกต์ใช้การประมวลผลข้อความเบื้องต้นเข้ามาดำเนินงาน เพื่อช่วยในการประมวลผลเนื้อหาของคำอธิบายรายวิชาในทุก ๆ มหาวิยาลัย ให้มีโครงสร้างและรูปแบบของเนื้อหาที่เหมือนกัน ซึ่งจะช่วยให้สามารถดำเนินการสกัดคำสำคัญจากเนื้อหาของแต่ละคำอธิบายรายวิชาได้อย่างมีประสิทธิภาพมากยิ่งขึ้น โดยเทคนิคการประมวลข้อความเบื้องต้นที่ได้นำมาประยุกต์ใช้ในระบบ *eCSCDA* มีดังต่อไปนี้

4.2.1.1 การแบ่งประโยค (Sentence tokenization)

คือ การแบ่งประโยคแต่ละประโยคในเนื้อหาของคำอธิบายรายวิชาออกจากกัน โดยการพิจารณาถึงสัญลักษณ์ที่ปรากฏอยู่ท้ายประโยคในแต่ละประโยค เช่น เครื่องหมายมหัพภาค (‘.’), เครื่องหมายจุลภาคหรือเครื่องหมายลูกน้ำ (‘,’) หรือ เครื่องหมายอัฒภาค (‘;’) เป็นต้น เมื่อดำเนินการแบ่งประโยคในเนื้อหาของคำอธิบายรายวิชาเสร็จเรียบร้อยแล้ว จะสามารถดำเนินการจัดเก็บประโยคที่ถูกแบ่งให้อยู่ในรูปแบบของ $cc_x = \{tp_{1,x}, tp_{2,x}, \dots, tp_{n,x}\}$

4.2.1.2 การแบ่งคำ (Word tokenization)

ขั้นตอนต่อมาจะเป็นการแบ่งคำในประโยคแต่ละโยคของคำอธิบายรายวิชาหนึ่ง ๆ ซึ่งในการแบ่งคำจะทำการพิจารณาถึงช่องว่างระหว่างคำ (White space) จากนั้นดำเนินการแบ่งคำแต่ละคำออกจากกัน

4.2.1.3 การแปลงตัวอักษรพิมพ์ใหญ่ให้เป็นตัวอักษรพิมพ์เล็ก (Lowercase conversion)

ในส่วนนี้จะเป็นการพิจารณาถึงตัวอักษรตัวพิมพ์ใหญ่ในภาษาอังกฤษที่ปรากฏอยู่ในเนื้อหาของคำอธิบายรายวิชาแต่ละคำอธิบาย ซึ่งเมื่อตรวจพบตัวอักษรตัวพิมพ์ใหญ่จะดำเนินการแทนที่ตัวอักษรตัวนั้นด้วยตัวอักษรตัวเดียวกันที่เป็นตัวพิมพ์เล็ก

4.2.1.4 การแก้ไขคำผิด (Word error correction)

การแก้ไขคำผิดเป็นเทคนิคที่ผู้วิจัยได้ทำการประยุกใช้จากวิธีการ SMC (Spelling Mistake Correction) (Gupta, 2015) ซึ่งจะเป็นการตรวจการสะกดคำของคำศัพท์แต่ละคำในประโยค โดยการนำคำศัพท์แต่ละคำไปดำเนินการตรวจสอบกับคำในพจนานุกรมภาษาอังกฤษ ซึ่งถ้าพบว่าคำศัพท์

คำนั้น ๆ มีการสะกดคำที่ไม่ตรงกับคำในพจนานุกรมจะถือว่าคำศัพท์คำนั้นมีการสะกดคำที่ผิด และจะทำการแทนที่คำศัพท์คำนั้นด้วยคำที่ถูกต้อง

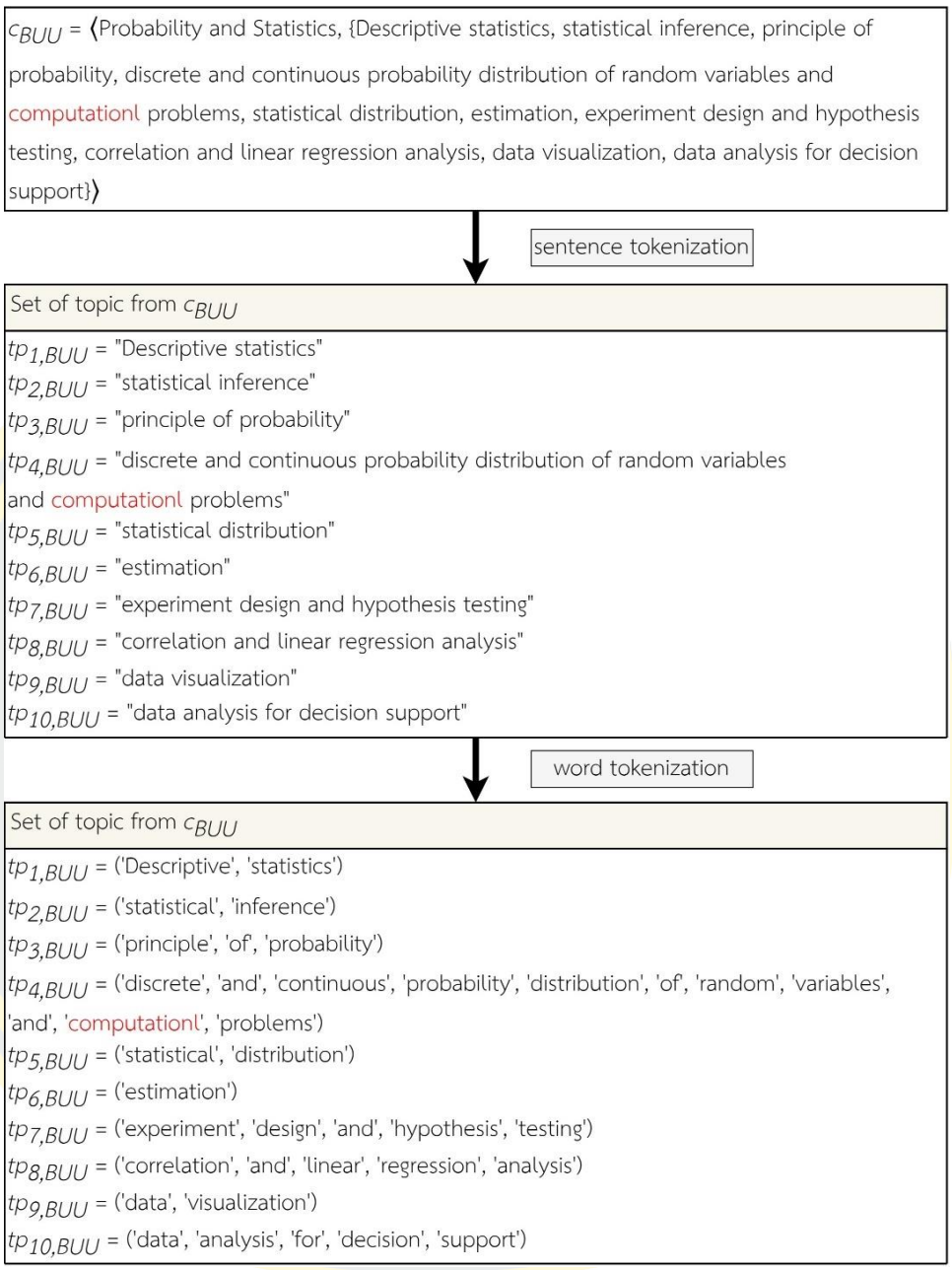
4.2.1.5 การกำจัดคำหยุด (Stopword removal)

การกำจัดคำหยุดของระบบ *eCSCDA* จะเป็นการพิจารณากำจัดคำหยุดบางคำที่ปรากฏอยู่ในเนื้อหาของคำอธิบายรายวิชา โดยคำหยุดที่สามารถกำจัดออกไปจะเป็นคำที่เมื่อกำจัดออกไปแล้วจะความหมายของประโยคนั้น ๆ จะไม่เกิดการเปลี่ยนแปลงไปจากเดิม เช่น คำหยุดคำว่า “a”, “an” และ “the” เป็นต้น ในงานวิจัยได้มีกำหนดเซตของคำหยุดที่ต้องลบออกจากการพิจารณา ได้แก่ คำหยุดประเภทคำบุพบท (Preposition) และ คำหยุดประเภทคำสันธาน (Conjunction) ซึ่งคำหยุดทั้ง 2 ประเภทนี้จะไม่ถูกกำจัดออกไปจากเนื้อหาในคำอธิบายรายวิชา

4.2.1.6 การระบุหน้าที่ของคำ (Part-of-speech tagging)

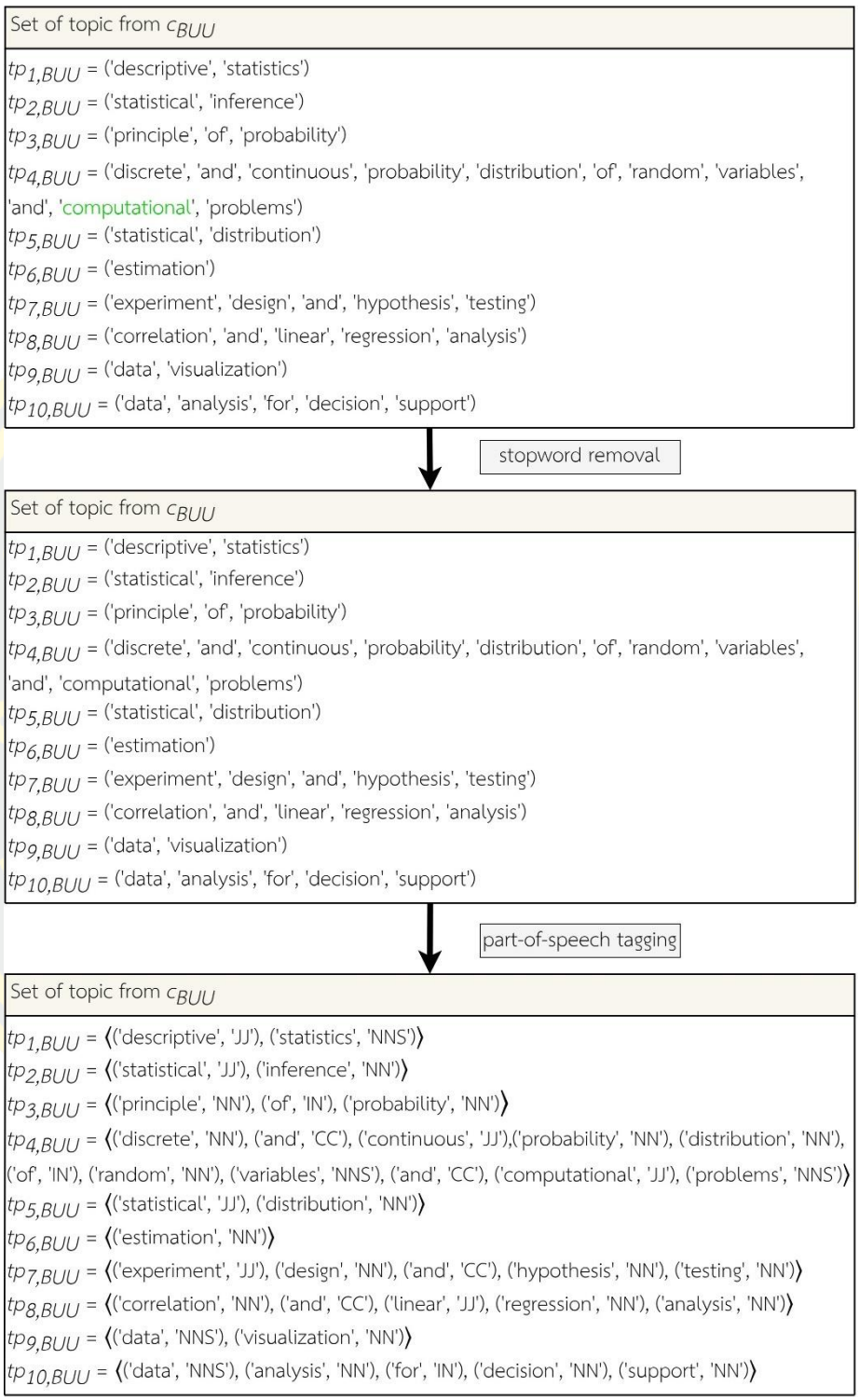
ในการระบุถึงหน้าที่ของคำแต่ละคำในเนื้อหาของคำอธิบายรายวิชาหนึ่ง ๆ ผู้วิจัยได้ทำการประยุกต์ใช้วิธีการของ Stanford part of speech tagger (Toutanova et al., 2003) เข้ามาช่วยดำเนินงาน ซึ่งเมื่อทำการระบุหน้าที่ของคำครบทุกคำในทุก ๆ ประโยคเรียบร้อยแล้ว จะสามารถจัดเก็บให้อยู่ในรูปแบบของ $tp_{i,x} = < \left(w_1^{tp_{i,x}}, tag\right), \left(w_2^{tp_{i,x}}, tag\right), \ldots, (w_n^{tp_{i,x}}, tag) >$

โดยมีตัวอย่างการดำเนินการประมวลผลข้อความเบื้องต้นกับเนื้อหาของคำอธิบายรายวิชาในรายวิชา “Probability and Statistics” ดังภาพที่ 29 ซึ่งเริ่มจากการแบ่งประโยคแต่ละประโยคในเนื้อหาของคำอธิบายรายวิชาออกจากกัน เมื่อดำเนินการแบ่งประโยคทำให้ได้มาซึ่งประโยคทั้งสิ้น 10 หัวข้อย่อย ต่อมาดำเนินการแบ่งคำแต่ละคำออกจากกันในทุก ๆ หัวข้อย่อย จากนั้นเมื่อดำเนินการแบ่งคำเสร็จแล้วในภาพที่ 30 จะเป็นการแปลงตัวอักษรตัวพิมพ์ใหญ่ให้กลายเป็นตัวพิมพ์เล็ก ซึ่งเมื่อพิจารณาจะพบว่าในหัวข้อย่อยที่ 1 คือ (‘Descriptive’, ‘statistics’) มีตัวอักษรตัวพิมพ์ใหญ่ปรากฏอยู่ ดังนั้นจึงต้องทำการแปลงให้กลายเป็นตัวเล็ก ได้แก่ (‘descriptive’, ‘statistics’) ต่อมาดำเนินการตรวจสอบและแก้ไขคำผิดจากการพิจารณาจะพบว่าคำว่า ‘computationl’ ในหัวข้อย่อยที่ 4 เป็นคำที่มีการสะกดผิดเกิดขึ้น (คำที่เป็นสีแดง) ดังนั้นจึงทำการแก้ไขให้มีการสะกดคำที่ถูกต้อง ได้แก่คำว่า ‘computational’ (คำที่เป็นสีเขียว) จากนั้นในภาพที่ 31 จะทำการกำจัดคำหยุดที่ปรากฏขึ้นในแต่ละหัวข้อย่อย ซึ่งเมื่อทำการพิจารณาในแต่ละหัวข้อย่อยไม่พบว่ามีคำหยุดที่ควรถูกกำจัดออกเลย ทำให้เนื้อหาในทุกหัวข้อย่อยยังคงเดิม สุดท้ายทำการระบุถึงหน้าที่ของคำแต่ละคำในทุก ๆ หัวข้อย่อย



ภาพที่ 29 ตัวอย่างการดำเนินการประมวลผลข้อความเบื้องต้นกับเนื้อของคำอธิบายรายวิชาในรายวิชา “Probability and Statistics” ของระบบ eCSCDA

ภาพที่ 30 ตัวอย่างการดำเนินการประมวลผลข้อความเบื้องต้นกับเนื้อของคำอธิบายรายวิชาในรายวิชา “Probability and Statistics” ของระบบ eCSCDA (ต่อ)



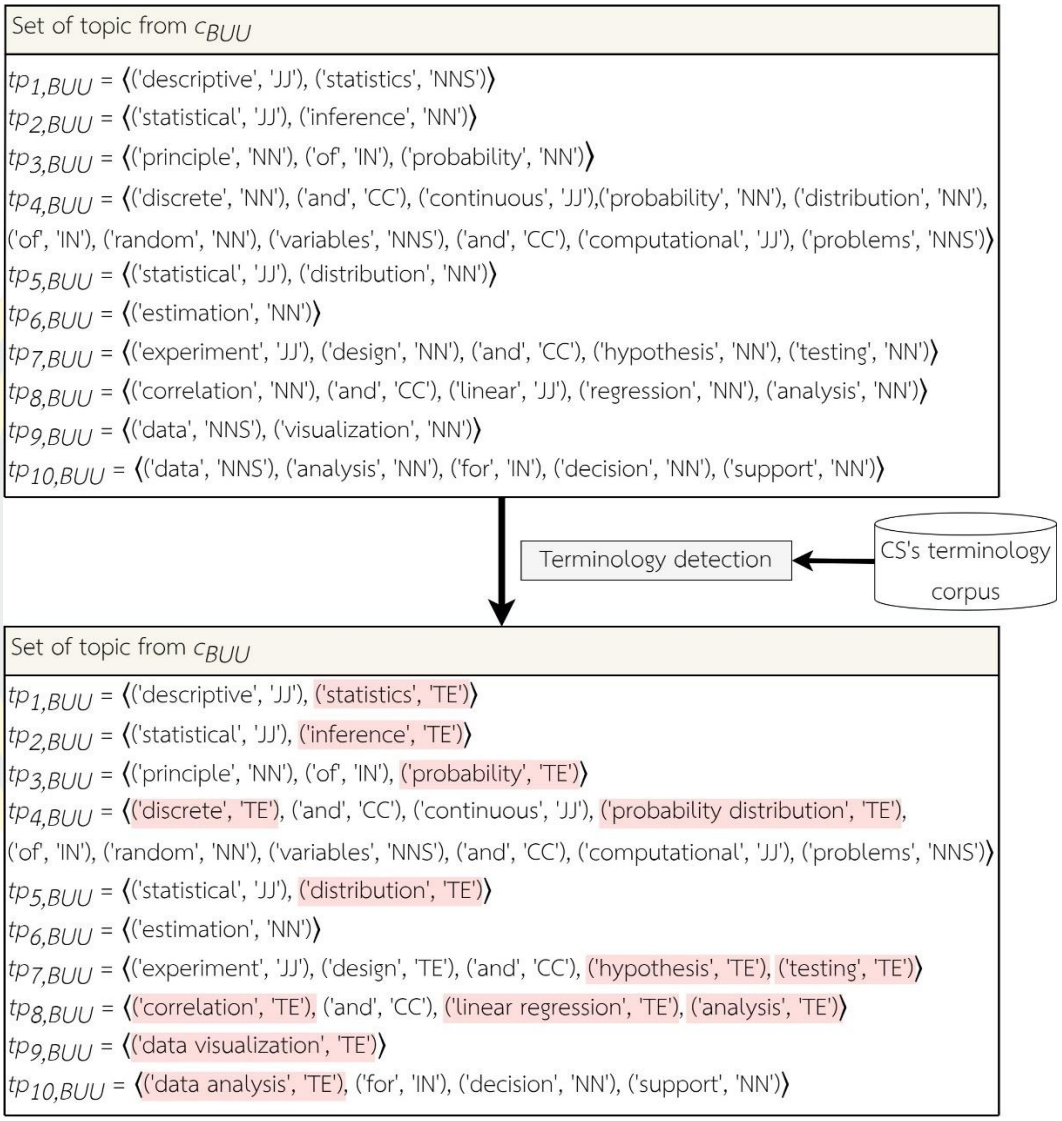
ภาพที่ 31 ตัวอย่างการดำเนินการประมวลผลข้อความเบื้องต้นกับเนื้อของคำอธิบายรายวิชาในรายวิชา “Probability and Statistics” ของระบบ eCSCDA (ต่อ)

4.2.2 การระบุคำศัพท์เฉพาะ

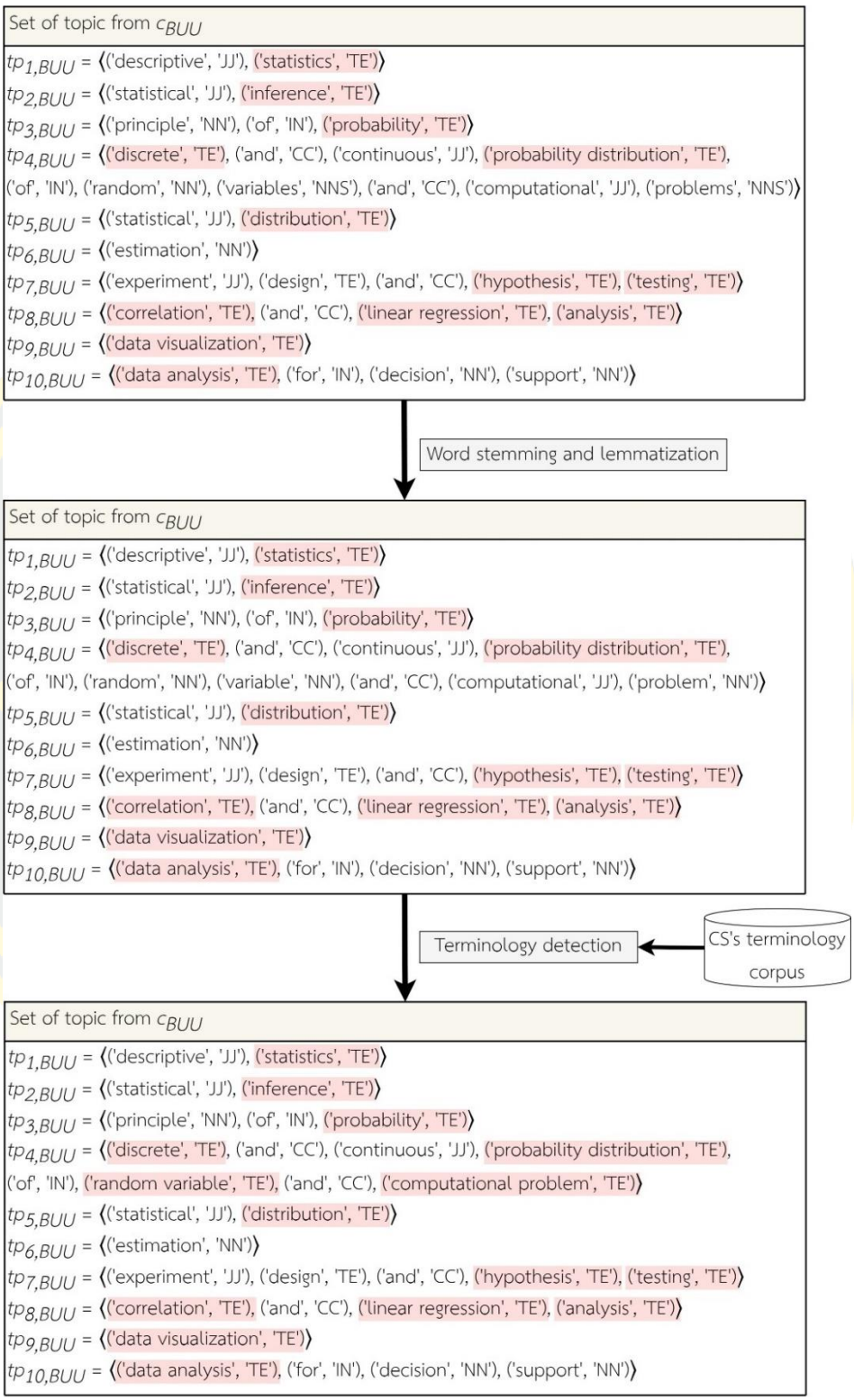
ขั้นตอนต่อมาจะเป็นการระบุถึงคำศัพท์เฉพาะในศาสตร์ทางด้านวิทยาการคอมพิวเตอร์ที่อยู่ในแต่ละหัวข้อย่อยของคำอธิบายรายวิชาหนึ่ง ๆ โดยเริ่มจากการประยุกต์ใช้วิธีการ N-gram เข้ามาเพื่อดำเนินการเปรียบเทียบระหว่างคำศัพท์คำหนึ่ง ๆ หรือกลุ่มของคำหนึ่ง ๆ กับคำศัพท์เฉพาะแต่ละคำในคลังคำศัพท์เฉพาะที่ได้จัดเตรียมไว้ในส่วนของหัวข้อที่ 4.1.1 โดยถ้าพบว่าคำศัพท์คำนั้น ๆ หรือกลุ่มคำนั้น ๆ เป็นคำศัพท์เฉพาะจะทำการเปลี่ยนหน้าที่ของคำนั้นหรือกลุ่มคำนั้นให้กลายเป็น ‘TE’ (Terminology) จากนั้นดำเนินการประยุกต์ใช้เทคนิคการแปลงรูปคำให้อยู่ในรากศัพท์ (Word stemming and lemmatization) เพื่อดำเนินการแปลงรูปของคำที่ยังไม่ถูกระบุว่าเป็นคำศัพท์เฉพาะในทุก ๆ หัวข้อย่อยให้อยู่ในรูปของรากศัพท์ของคำคำนั้น สุดท้ายทำการระบุถึงคำศัพท์เฉพาะอีกครั้งหนึ่ง ซึ่งจากการระบุถึงคำศัพท์เฉพาะในแต่ละหัวข้อย่อยจะช่วยให้การสกัดคำสำคัญสามารถดำเนินการได้อย่างมีประสิทธิภาพ และได้มาซึ่งคำสำคัญที่มีความครอบคลุมและมีประสิทธิภาพ

โดยมีตัวอย่างขั้นตอนการระบุถึงคำศัพท์เฉพาะดังภาพที่ 32 จะเป็นการดำเนินงานโดยการใช้ผลลัพธ์ที่ได้จากการประมวลผลข้อความเบื้องต้นในส่วนของขั้นตอนที่ 4.2.1 ซึ่งเริ่มจากการระบุถึงคำศัพท์เฉพาะในแต่ละหัวข้อย่อย เมื่อทำการระบุถึงคำศัพท์เฉพาะในแต่ละหัวข้อย่อยเสร็จเรียบร้อยแล้ว จะพบว่าหน้าที่ของคำที่ถูกระบุไว้โดยเทคนิคการระบุหน้าที่ของคำในส่วนที่ 4.2.1.6 จะถูกเปลี่ยนหน้าที่ของคำคำนั้นเป็น ‘TE’ (คำที่มีพื้นหลังสีแดง) อาทิเช่น หัวข้อย่อยที่ 1 มีเนื้อหาคือ (‘descriptive’, ‘JJ’), (‘statistics’, ‘NNS’) เมื่อทำการตรวจสอบและระบุถึงคำศัพท์เฉพาะจะพบคำว่า (‘statistics’, ‘NNS’) เป็นคำศัพท์เฉพาะ จึงทำการเปลี่ยนหน้าที่คำจากเดิมคือ ‘NNS’ ให้กลายเป็น ‘TE’ ดังนั้นในส่วนของหัวข้อย่อยที่ 1 จะมีเนื้อหาคือ (‘descriptive’, ‘JJ’), (‘statistics’, ‘TE’) ต่อมาในภาพที่ 33 จะเป็นการประยุกต์ใช้เทคนิคการแปลงรูปของคำให้อยู่ในรากศัพท์ โดยเป็นการแปลงรูปคำที่ยังไม่ถูกระบุว่าเป็นคำศัพท์เฉพาะในแต่หัวข้อย่อย อาทิเช่น หัวข้อย่อยที่ 4 มีเนื้อหาคือ (‘discrete’, ‘TE’), (‘and’, ‘CC’), (‘continuous’, ‘JJ’), (‘probability distribution’, ‘TE’), (‘of’, ‘IN’), (‘random’, ‘NN’), (‘variables’, ‘NNS’), (‘and’, ‘CC’), (‘computational’, ‘JJ’), (‘problems’, ‘NNS’) เมื่อใช้เทคนิคการแปลงรูปของคำให้อยู่ในรากศัพท์ ทำให้คำว่า (‘variables’, ‘NNS’) และ (‘problems’, ‘NNS’) กลายเป็น (‘variable’, ‘NN’) และ (‘problem’, ‘NN’) สุดท้ายดำเนินการระบุถึงคำศัพท์เฉพาะในทุก ๆ หัวข้อย่อยอีกครั้งหนึ่ง เช่น หัวข้อย่อยที่ 4 เมื่อถูกระบุถึงคำศัพท์เฉพาะอีกครั้งหนึ่ง พบว่ามีกลุ่มคำที่เป็นคำศัพท์เฉพาะ คือ ‘random variable’ และ ‘computational problem’ ทำให้ต้องระบุหน้าที่ของกลุ่มคำนี้เป็น ‘TE’ ดังนั้นในประโยคที่ 4 จะมี

เนื้อหาคือ (‘discrete’, ‘TE’), (‘and’, ‘CC’), (‘continuous’, ‘JJ’), (‘probability distribution’, ‘TE’), (‘ of’, ‘IN’), (‘random variable’, ‘TE’), (‘and’, ‘CC’), (‘computational problem’, ‘TE’)



ภาพที่ 32 ตัวอย่างการระบุถึงคำศัพท์เฉพาะในแต่ละหัวข้อย่อยของระบบ eCSCDA



ภาพที่ 33 ตัวอย่างการระบุถึงคำศัพท์เฉพาะในแต่ละหัวข้อย่อยของระบบ eCSCDA (ต่อ)

4.2.3 การสกัดคำสำคัญ

ในส่วนของขั้นตอนการสกัดคำสำคัญนี้จะเป็นการประยุกต์ใช้กฎทางภาษาศาสตร์ฉบับปรับปรุง ที่ได้ถูกจัดเตรียมไว้ในส่วนของหัวข้อที่ 4.1.2 เข้ามาสกัดคำสำคัญในแต่ละหัวข้อย่อย โดยคำสำคัญที่สกัดได้อาจอยู่ในรูปของ 1) คำศัพท์เฉพาะคำเดียวหรือกลุ่มของคำศัพท์เฉพาะกลุ่มเดียว, 2) คำนามคำเดียวหรือกลุ่มของคำนามกลุ่มเดียว, 3) คำคุณศัพท์ + คำศัพท์เฉพาะ, 4) คำคุณศัพท์ + คำนาม, 5) คำนาม + คำนาม, 6) คำนาม + คำศัพท์เฉพาะ, 7) คำศัพท์เฉพาะ + คำศัพท์เฉพาะ, 8) คำศัพท์เฉพาะ + คำนาม, 9) คำคุณศัพท์ + คำนาม + คำศัพท์เฉพาะ, 10) คำนาม + คำศัพท์เฉพาะ + คำศัพท์เฉพาะ เป็นต้น จากคำสำคัญที่สกัดได้ในแต่ละหัวข้อย่อยสามารถจัดเก็บให้อยู่ในรูปแบบของ $K^{tp_{i,x}} = \{k_1^{tp_{i,x}}, k_n^{tp_{i,x}}, \dots, k_n^{tp_{i,x}}\}$ โดยมีตัวอย่างการสกัดคำสำคัญในแต่ละหัวข้อย่อยดังต่อไปนี้

หัวข้อย่อยที่ 1 คือ “(‘descriptive’, ‘JJ’), (‘statistics’, ‘TE’)” เมื่อทำการพิจารณาแต่ละคำในหัวข้อย่อยที่ 1 ไม่พบว่ามีคำสันธานหรือคำบุพบทปรากฏอยู่ในเนื้อหา ทำให้ไม่ต้องแบ่งกลุ่มคำออกจากกัน โดยหัวข้อย่อยที่ 1 มีโครงสร้าง คือ “Adjective (JJ) + Terminology (TE)” ซึ่งตรงกับกฎทางภาษาศาสตร์ฉบับปรับปรุง ได้แก่ กฎ “Adjective (JJ) + Terminology (TE)” ⟶ ⟨‘Adjective (JJ) + Terminology (TE)’⟩ ทำให้ได้มาซึ่งคำสำคัญที่สกัดได้จากกฎทางภาษาศาสตร์ฉบับปรับปรุงข้างต้น คือ ⟨‘descriptive(JJ) statistics(TE)’⟩ จากนั้นนำคำสำคัญที่สกัดได้มาตรวจสอบถึงคำคุณศัพท์ที่ไม่ได้บ่งบอกถึงใจความสำคัญ เมื่อตรวจสอบพบคำว่า ‘descriptive(JJ)’ เป็นคำคุณศัพท์ที่ปรากฏอยู่ในคำสำคัญ จึงดำเนินการตรวจสอบคำคุณศัพท์ที่ที่ปรากฏขึ้นกับคลังคำคุณศัพท์ที่ไม่ได้บ่งบอกถึงใจความสำคัญ จากการตรวจสอบไม่พบคำว่า ‘descriptive(JJ)’ อยู่ในคำคุณศัพท์ที่ไม่ได้บ่งบอกถึงใจความสำคัญ จึงไม่ทำการลบคำว่า ‘descriptive(JJ)’ ออก ทำให้คำสำคัญที่สกัดได้ยังคงเดิม ดังนั้นการสกัดคำสำคัญด้วยกฎทางภาษาศาสตร์ฉบับปรับปรุงของหัวข้อย่อยที่ 1 ได้คำสำคัญ คือ ⟨‘descriptive(JJ) statistics(TE)’⟩

หัวข้อย่อยที่ 4 คือ “(‘discrete’, ‘TE’), (‘and’, ‘CC’), (‘continuous’, ‘JJ’), (‘probability distribution’, ‘TE’), (‘of’, ‘IN’), (‘random variable’, ‘TE’), ('and', 'CC'), (‘computational problem’, ‘TE’)” เมื่อทำการพิจารณาแต่ละคำจะพบคำว่า (‘of’, ‘IN’) เป็นคำบุพบทที่ปรากฏขึ้น จึงทำการแบ่งกลุ่มคำออกเป็น 2 กลุ่ม คือ 1) “(‘discrete’, ‘TE’), (‘and’, ‘CC’), (‘continuous’, ‘JJ’), (‘probability distribution’, ‘TE’)” และ 2) “(‘random variable’, ‘TE’), ('and', 'CC'), (‘computational problem’, ‘TE’)” จากนั้นทำการพิจารณาใน

แต่ละกลุ่ม โดยเมื่อพิจารณากลุ่มที่ 1 พบคำว่า (‘and’, ‘CC’) เป็นคำสันธานที่ปรากฏขึ้น จึงทำการแบ่งกลุ่มออกเป็น 2 กลุ่มย่อย ได้แก่ 1.1) “(‘discrete’, ‘TE’)” และ 1.2) “(‘continuous’, ‘JJ’), (‘probability distribution’, ‘TE’)” ทำให้กลุ่มที่ 1 มีโครงสร้าง คือ “(Phrase1) + CC + (Phrase2)” ซึ่งมีโครงสร้างตรงกับกฎฉบับปรับปรุง ได้แก่ กฎ “(Phrase1) + CC + (Phrase2) *ถ้า (Phrase1) มีแค่ 1 คำ” ⟶ ⟨‘Phrase1 + Phrase2 (ที่ไม่มีคำแรก)’ และ ‘Phrase2’⟩ ทำให้ในกลุ่มที่ 1 ได้มาซึ่งคำสำคัญที่สกัดได้จากกฎทางภาษาศาสตร์ฉบับปรับปรุงข้างต้น คือ ⟨‘discrete(TE) probability distribution(TE)’ และ ‘continuous(JJ) probability distribution(TE)’⟩ ต่อมาในกลุ่มที่ 2 เมื่อตรวจสอบพบคำว่า (‘and’, ‘CC’) เป็นคำสันธานที่ปรากฏขึ้น จึงทำการแบ่งกลุ่มออกเป็น 2 กลุ่มย่อย ได้แก่ 2.2) “(‘random variable’, ‘TE’)” และ 2.2) “(‘computational problem’, ‘TE’)” ทำให้กลุ่มที่ 2 มีโครงสร้าง คือ “(Phrase1) + CC + (Phrase2)” ซึ่งมีโครงสร้างตรงกับกฎฉบับปรับปรุง ได้แก่ กฎ “(Phrase1) + CC + (Phrase2)” ⟶ ⟨‘(Phrase1)’, ‘(Phrase2)’⟩ ทำให้ในกลุ่มที่ 2 ได้มาซึ่งคำสำคัญที่สกัดได้จากกฎทางภาษาศาสตร์ฉบับปรับปรุงข้างต้น คือ ⟨‘random variable (TE)’ และ ‘computational problem(TE)’⟩ จากนั้นนำคำสำคัญที่สกัดได้แต่ละคำมาตรวจสอบถึงคำคุณศัพท์ที่ไม่ได้บ่งบอกถึงใจความสำคัญ เมื่อตรวจสอบพบคำว่า ‘continuous(JJ)’ เป็นคำคุณศัพท์ที่ปรากฏอยู่ในคำสำคัญ ‘continuous(JJ) probability distribution(TE)’ จึงดำเนินการตรวจสอบคำคุณศัพท์ที่ปรากฏขึ้นกับคลังคำคุณศัพท์ที่ไม่ได้บ่งบอกถึงใจความสำคัญ จากการตรวจสอบไม่พบคำว่า ‘continuous(JJ)’ อยู่ในคลังคำคุณศัพท์ที่ไม่ได้บ่งบอกถึงใจความสำคัญ จึงไม่ทำการลบคำว่า ‘continuous(JJ)’ ออก ทำให้คำสำคัญ ‘continuous(JJ) probability distribution(TE)’ ยังคงเดิม ดังนั้นการสกัดคำสำคัญด้วยกฎทางภาษาศาสตร์ฉบับปรับปรุงของหัวข้อย่อยที่ 4 ได้คำสำคัญ คือ ⟨‘discrete(TE) probability distribution(TE)’, ‘continuous(JJ) probability distribution(TE)’, ‘random variable (TE)’, ‘computational problem(TE)’⟩

โดยจะแสดงให้เห็นถึงการสกัดคำสำคัญในแต่ละหัวข้อย่อย ดังภาพที่ 34

Keyword extraction from c_{BUU}
$tp_{1,BUU}$ = ⟨('descriptive', 'JJ'), ('statistics', 'TE')⟩ Linguistic rule : (JJ) + (TE) ➞ ⟨'(JJ) + (TE)'⟩ $k_1^{tp_{1,BUU}}$ = ⟨'descriptive(JJ) statistics(TE)'⟩
$tp_{2,BUU}$ = ⟨('statistical', 'JJ'), ('inference', 'TE')⟩ Linguistic rule : (JJ) + (TE) ➞ ⟨'(JJ) + (TE)'⟩ $k_2^{tp_{2,BUU}}$ = ⟨'statistical(JJ) inference(TE)'⟩
$tp_{3,BUU}$ = ⟨('principle', 'NN'), ('of', 'IN'), ('probability', 'TE')⟩ Linguistic rule : (Phrase1) + IN + (Phrase2) ➞ ⟨'(Phrase2) + (Phrase1)'⟩ $k_3^{tp_{3,BUU}}$ = ⟨'probability(TE) principle(NN)'⟩
$tp_{4,BUU}$ = ⟨('discrete', 'TE'), ('and', 'CC'), ('continuous', 'JJ'), ('probability distribution', 'TE'), ('of', 'IN'), ('random variable', 'TE'), ('and', 'CC'), ('computational problem', 'TE')⟩ แบ่งกลุ่มคำจากการพิจารณาคำสันธานหรือคำบุพบท จะได้กลุ่มที่ 1 คือ ⟨('discrete', 'TE'), ('and', 'CC'), ('continuous', 'JJ'), ('probability distribution', 'TE')⟩ Linguistic rule : (Phrase1) + CC + (Phrase2) *if (Phrase1) has one word ➞ ⟨'(Phrase1) + (Phrase2 *without first word)', '(Phrase2)'⟩ จะได้กลุ่มที่ 2 คือ ⟨('random variable', 'TE'), ('and', 'CC'), ('computational problem', 'TE')⟩ Linguistic rule : (Phrase1) + CC + (Phrase2) ➞ ⟨'(Phrase1)', '(Phrase2)'⟩ $k_4^{tp_{4,BUU}}$ = ⟨'discrete(TE) probability distribution(TE)', 'continuous(JJ) probability distribution(TE)', 'random variable(TE)', 'computational problem(TE)'⟩
$tp_{5,BUU}$ = ⟨('statistical', 'JJ'), ('distribution', 'TE')⟩ Linguistic rule : (JJ) + (TE) ➞ ⟨'(JJ) + (TE)'⟩ $k_5^{tp_{5,BUU}}$ = ⟨'statistical(JJ) distribution(TE)'⟩
$tp_{6,BUU}$ = ⟨('estimation', 'NN')⟩ Linguistic rule : (NN) ➞ ⟨'(NN)'⟩ $k_6^{tp_{6,BUU}}$ = ⟨'estimation(NN)'⟩
$tp_{7,BUU}$ = ⟨('experiment', 'JJ'), ('design', 'TE'), ('and', 'CC'), ('hypothesis', 'TE'), ('testing', 'TE')⟩ Linguistic rule : (Phrase1) + CC + (Phrase2) ➞ ⟨'(Phrase1)', '(Phrase2)'⟩ $k_7^{tp_{7,BUU}}$ = ⟨'experiment(NN) design(TE)', 'hypothesis(TE) testing(TE)'⟩
$tp_{8,BUU}$ = ⟨('correlation', 'TE'), ('and', 'CC'), ('linear regression', 'TE'), ('analysis', 'TE')⟩ Linguistic rule : (Phrase1) + CC + (Phrase2) ➞ ⟨'(Phrase1) + (Phrase2)', '(Phrase2)'⟩ $k_8^{tp_{8,BUU}}$ = ⟨'correlation(TE) analysis(TE)', 'linear regression(TE) analysis(TE)'⟩
$tp_{9,BUU}$ = ⟨('data visualization', 'TE')⟩ Linguistic rule : (TE) ➞ ⟨'(TE)'⟩ $k_9^{tp_{9,BUU}}$ = ⟨'data visualization(TE)'⟩
$tp_{10,BUU}$ = ⟨('data analysis', 'TE'), ('for', 'IN'), ('decision', 'NN'), ('support', 'NN')⟩ Linguistic rule : (Phrase1) + IN + (Phrase2) ➞ ⟨'(Phrase2) + (Phrase1)'⟩ $k_{10}^{tp_{10,BUU}}$ = ⟨'decision(NN) support(NN) data analysis(TE)'⟩

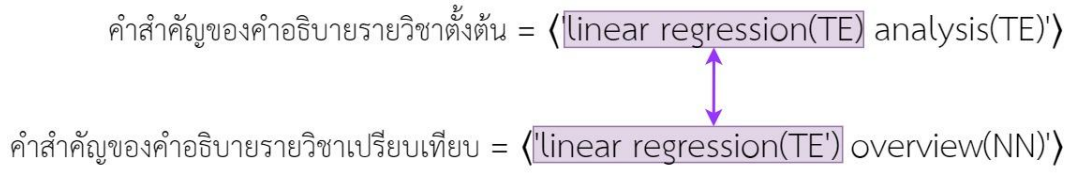
ภาพที่ 34 ตัวอย่างการสกัดคำสำคัญในแต่ละหัวข้อย่อยของระบบ eCSCDA

4.3 การเปรียบเทียบคำสำคัญระหว่างคำอธิบายรายวิชา

การเปรียบเทียบคำสำคัญระหว่าง 2 คำอธิบายรายวิชาในรายวิชาหนึ่ง ๆ จะเป็นการนำคำสำคัญที่สกัดได้จากเนื้อหาของคำอธิบายรายวิชาทั้ง 2 คำอธิบายมาดำเนินการเปรียบเทียบกัน โดยวิธีการเปรียบเทียบผู้วิจัยได้นำวิธีการเปรียบเทียบแบบตรงตัว (Exact matching), วิธีการเปรียบเทียบแบบเซตย่อย (Subset matching) และ วิธีการเปรียบเทียบแบบซูเปอร์เซต (Superset matching) ของระบบ *CSCDA* เข้ามาประยุกต์ใช้งาน โดยในระบบ *eCSCDA* จะทำการรวมวิธีการเปรียบเทียบแบบเซตย่อยและวิธีการเปรียบเทียบแบบเซตซูเปอร์เซตเข้าด้วยกัน จึงทำให้กลายเป็นวิธีการเปรียบเทียบแบบเซตย่อย/ซูเปอร์เซต (Sub/superset matching) นอกจากนี้ในระบบ *eCSCDA* ผู้วิจัยยังได้ทำการเพิ่มวิธีการเปรียบเทียบอีก 2 วิธีการ คือ 1) วิธีการเปรียบเทียบแบบองค์ประกอบร่วม (Sub-keyword matching) และ 2) วิธีการเปรียบเทียบเชิงความหมาย (Semantic matching) ตามลำดับ ซึ่งจากการดำเนินการดังกล่าวจะทำให้ระบบ *eCSCDA* ประยุกต์ใช้วิธีการเปรียบเทียบทั้งสิ้น 5 วิธี แต่ถูกจัดกลุ่มยุบรวมเหลือเพียง 4 วิธี โดย 2 วิธีใหม่ที่มีการประยุกต์ใช้มีรายละเอียดดังนี้

4.3.1 วิธีการเปรียบเทียบแบบองค์ประกอบร่วม (Sub-keyword matching)

วิธีการเปรียบเทียบแบบองค์ประกอบร่วม จะเริ่มจากการตรวจสอบถึงความเท่ากันของจำนวนหน้าที่ของคำ (tag) ของทั้ง 2 คำสำคัญ จากนั้นทำการพิจารณาถึงองค์ประกอบร่วมระหว่าง 2 คำสำคัญว่าทั้ง 2 คำสำคัญมีองค์ประกอบร่วมที่เหมือนกันหรือไม่ ซึ่งถ้าพบว่าคำสำคัญทั้ง 2 คำที่นำมาเปรียบเทียบกันมีจำนวนของหน้าที่ของคำเท่ากันและมีองค์ประกอบร่วมที่เหมือนกัน จะสามารถสรุปได้ว่าคำสำคัญทั้ง 2 คำที่นำมาเปรียบเทียบกันมีความคล้ายกัน โดยมีตัวอย่างการเปรียบเทียบแบบองค์ประกอบร่วม ในภาพที่ 35



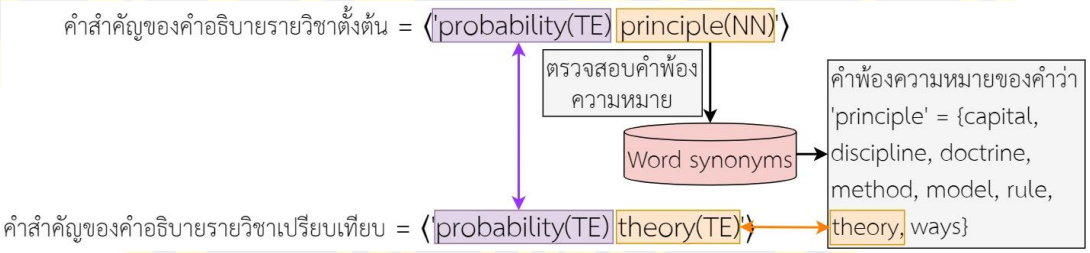
ภาพที่ 35 ตัวอย่างวิธีการเปรียบเทียบแบบองค์ประกอบร่วม

จากภาพที่ 35 แสดงให้เห็นถึงการเปรียบเทียบระหว่างคำสำคัญของคำอธิบายรายวิชาตั้งต้น คือ ⟨‘linear regression(TE) analysis(TE)’⟩ และคำสำคัญของคำอธิบายรายวิชาเปรียบเทียบ คือ ⟨‘linear regression(TE') overview(NN)’⟩ ด้วยวิธีการเปรียบเทียบแบบองค์ประกอบร่วม ซึ่งเมื่อทำ

การพิจารณาถึงจำนวนหน้าที่ของคำในแต่ละคำสำคัญพบว่าทั้ง 2 คำสำคัญมีจำนวนหน้าที่ของคำเท่ากัน จากนั้นทำการตรวจสอบถึงองค์ประกอบร่วมของทั้ง 2 คำสำคัญ จะพบว่าทั้ง 2 คำสำคัญมีคำว่า ‘linear regression(TE)’ ที่เป็นองค์ประกอบร่วมเหมือนกันใน ดังนั้นจึงสามารถสรุปได้ว่าคำสำคัญทั้ง 2 คำมีความคล้ายกัน

4.3.2 วิธีการเปรียบเทียบเชิงความหมาย (Semantic matching)

วิธีการเปรียบเทียบเชิงความหมาย จะทำการนำคลังคำพ้องความหมายของคำศัพท์เฉพาะที่ได้จัดเตรียมไว้ในส่วนที่ 4.1.3 และคลังคำพ้องความหมายของคำศัพท์ทั่วไปที่ได้จัดเตรียมไว้ในส่วนที่ 4.1.4 เข้ามาร่วมดำเนินการในการเปรียบเทียบระหว่างคำสำคัญ โดยในการเปรียบเทียบเชิงความหมายจะทำการพิจารณาถึงองค์ประกอบร่วมระหว่างสำคัญ ถ้าพบว่าทั้ง 2 คำสำคัญมีองค์ประกอบร่วมที่เหมือนกันจากนั้นจะทำการเปรียบเทียบเชิงความหมายในส่วนของคำที่เหลือของทั้ง 2 คำสำคัญ โดยการนำคลังคำพ้องความหมายของคำศัพท์เฉพาะ และคลังคำพ้องความหมายของคำศัพท์ทั่วไปเข้ามาดำเนินการ ซึ่งถ้าพบว่าคำที่เหลือของทั้ง 2 คำสำคัญเป็นคำพ้องความหมายกัน จะสามารถสรุปได้ว่าคำสำคัญทั้ง 2 คำที่นำมาเปรียบเทียบกันมีความคล้ายกัน โดยมีตัวอย่างการเปรียบเทียบเชิงความหมาย ดังภาพที่ 36



ภาพที่ 36 ตัวอย่างวิธีการเปรียบเทียบเชิงความหมาย

ภาพที่ 36 แสดงการเปรียบเทียบเชิงความหมายระหว่างคำสำคัญของคำอธิบายรายวิชาตั้งต้น คือ ⟨‘probability(TE) principle(NN)’⟩ และคำสำคัญของคำอธิบายรายวิชาเปรียบเทียบ คือ ⟨‘probability(TE) theory(TE)’⟩ โดยเริ่มจากการพิจารณาถึงองค์ประกอบร่วมของทั้ง 2 คำสำคัญ ซึ่งจะพบว่ามีคำที่เป็นองค์ประกอบร่วมคือ ‘probability(TE)’ จากนั้นนำคำที่เหลือมาทำการเปรียบเทียบเชิงความหมาย ในที่นี้คำที่เหลือในคำสำคัญของคำอธิบายรายวิชาตั้งต้นคือ ‘principle(NN)’ ซึ่งเป็นคำศัพท์ทั่วไป (เนื่องจากเป็นคำที่ไม่ได้ถูกระบุหน้าที่ของคำเป็น ‘TE’) ดังนั้นจึงนำคลังคำพ้องความหมายของคำศัพท์ทั่วไปเข้ามาดำเนินการเปรียบเทียบเชิงความหมาย จากการตรวจสอบพบว่าคำพ้องความหมายของคำว่า ‘principle(NN)’ คือ ‘theory’ ตรงกับคำที่เหลือในคำ

สำคัญของคำอธิบายรายวิชาเปรียบเทียบ ดังนั้นจากการเปรียบเทียบเชิงความหมายระหว่างคำสำคัญทั้ง 2 คำ จึงสามารถสรุปได้ว่าคำสำคัญของคำอธิบายรายวิชาตั้งต้นมีความคล้ายกับคำสำคัญของคำอธิบายรายวิชาเปรียบเทียบ

หลังจากประยุกต์ใช้วิธีการเปรียบเทียบทั้ง 4 วิธีการของระบบ *eCSCDA* ได้แก่ 1) วิธีการเปรียบเทียบแบบตรงตัว (Exact matching), 2) วิธีการเปรียบเทียบแบบเซตย่อย/ซูเปอร์เซต (Sub/superset matching), 3) วิธีการเปรียบเทียบแบบองค์ประกอบร่วม (Sub-keyword matching) และ 4) วิธีการเปรียบเทียบเชิงความหมาย (Semantic matching) กับหัวข้อย่อยหนึ่ง ๆ ของคำอธิบายรายวิชาตั้งต้น โดยหากหัวข้อย่อยของคำอธิบายรายวิชาตั้งต้นมีความเหมือนหรือคล้ายกับหัวข้อย่อยของคำอธิบายรายวิชาเปรียบเทียบ จะทำการกำหนดให้ค่าความเหมือนของหัวข้อย่อยของคำอธิบายรายวิชาตั้งต้นมีค่าเท่ากับ 1 ในทางกลับกันหากหัวข้อย่อยของคำอธิบายรายวิชาตั้งต้นไม่เหมือนหรือไม่คล้ายกับหัวข้อย่อยใด ๆ ของคำอธิบายรายวิชาเปรียบเทียบ จะทำการกำหนดให้ค่าความเหมือนของหัวข้อย่อยของคำอธิบายรายวิชาตั้งต้นมีค่าเท่ากับ 0 จากการนิยามข้างต้นสามารถนำมาสร้างให้อยู่ในรูปของสมการได้ดังต่อไปนี้

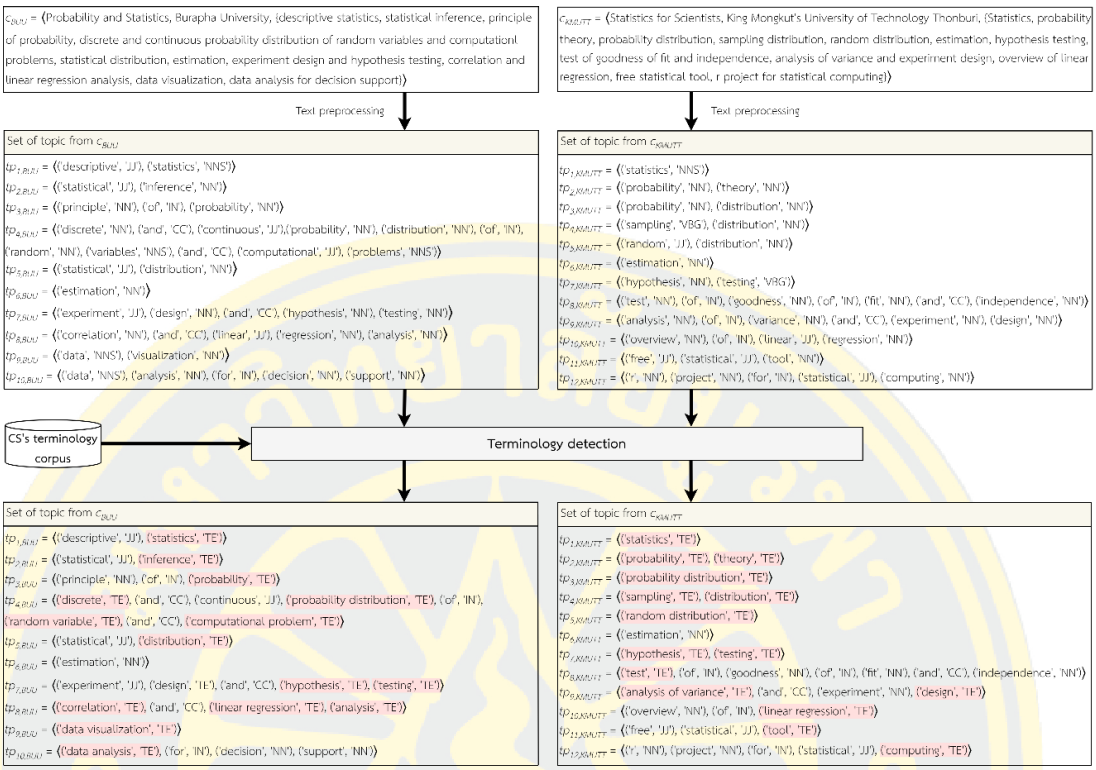
$$match(tp_{i,x}) = \begin{cases} 1, \ \{\exists tp_{u,y} \in c_y | K^{tp_{i,x}} \ match \ with \ K^{tp_{u,y}}, \\ \quad K^{tp_{i,x}} \not\subset s_x, K^{tp_{i,x}} \not\subset s_y\} \\ 0, \ otherwise \end{cases} \quad (4)$$

จากการที่ได้มาซึ่งคะแนนของการเปรียบเทียบในแต่ละหัวข้อย่อยของคำอธิบายรายวิชาตั้งต้น ต่อมาจะนำคะแนนที่ได้ทั้งหมดมาคำนวณหาอัตราร้อยละของความเหมือนของคำอธิบายรายวิชาตั้งต้น (ดำเนินการเช่นเดียวกับระบบ *CSCDA*) โดยการคำนวณหาอัตราร้อยละสามารถคำนวณได้จากสมการดังต่อไปนี้

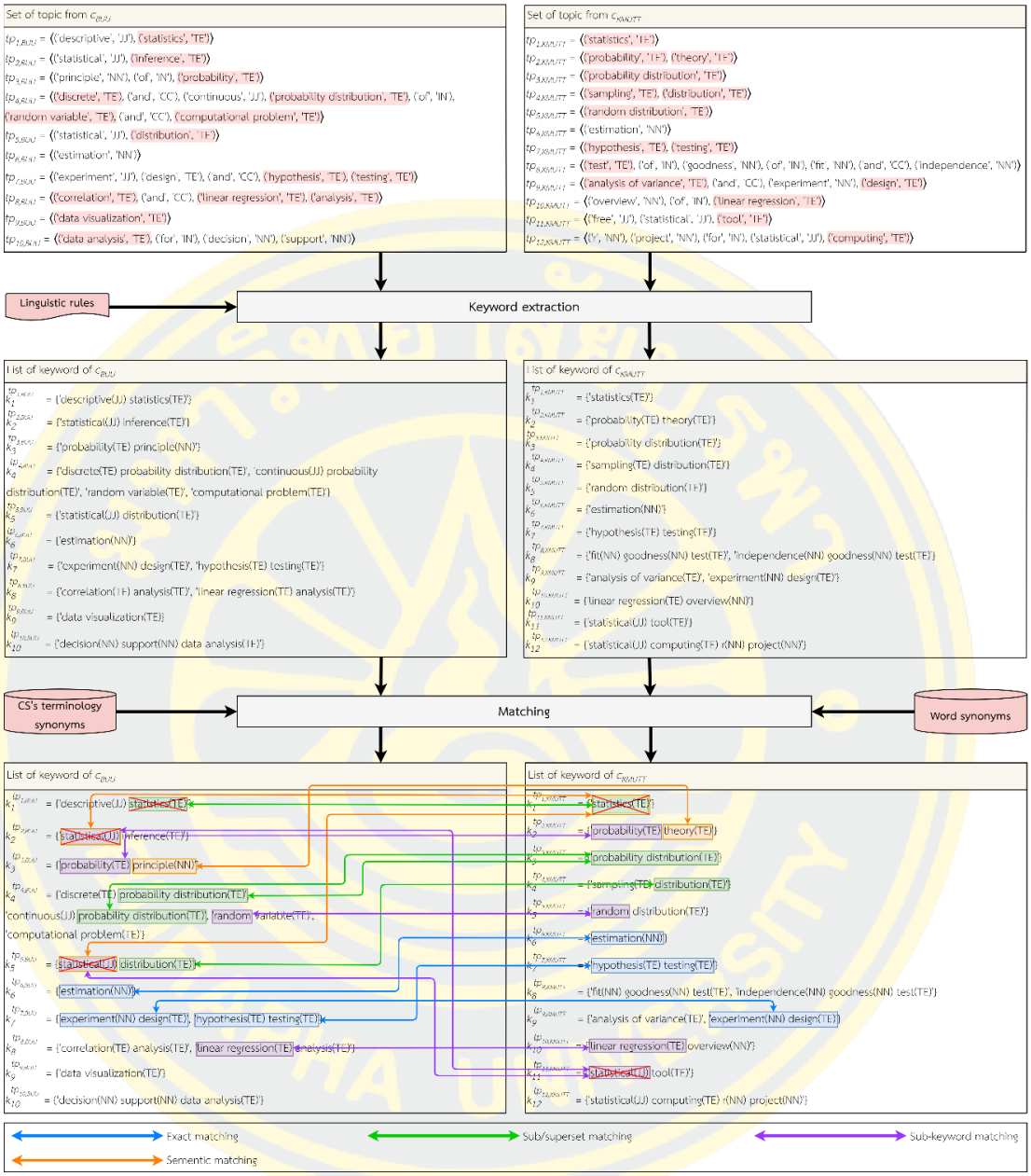
$$per\_sim(c_x, c_y) = \frac{\sum_{i=1}^{n} match(tp_{i,x})}{n} \quad (5)$$

โดยที่ n คือจำนวนหัวข้อย่อยทั้งหมดของคำอธิบายรายวิชาตั้งต้น

โดยมีตัวอย่างการเปรียบเทียบคำอธิบายรายวิชาของระบบ *eCSCDA* ในทุกขั้นตอนที่กล่าวมาข้างต้น จะถูกแสดงดังภาพที่ 37 และ 38



ภาพที่ 37 ตัวอย่างการเปรียบเทียบระหว่างคำอธิบายรายวิชาของระบบ eCSCDA



ภาพที่ 38 ตัวอย่างการเปรียบเทียบระหว่างคำอธิบายรายวิชาของระบบ eCSCDA

ภาพที่ 37 แสดงตัวอย่างการดำเนินงานของระบบ *eCSCDA* ในเปรียบเทียบระหว่างคำอธิบายรายวิชาของมหาวิทยาลัยบูรพา (กำหนดให้เป็นอธิบายรายวิชาตั้งต้น) และคำอธิบายรายวิชาของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี (กำหนดให้เป็นคำอธิบายรายวิชาเปรียบเทียบ) ในรายวิชา “Probability and statistics” โดยเริ่มการประยุกต์ใช้เทคนิคการประมวลข้อความเบื้องต้นเข้ามาประมวลผลทั้ง 2 คำอธิบายรายวิชา จากการประมวลผลทำให้สามารถแบ่งหัวข้อย่อยในคำอธิบายรายวิชาตั้งต้นได้ 10 หัวข้อย่อย และหัวข้อย่อยในคำอธิบายรายวิชา

เปรียบเทียบได้ 12 หัวข้อย่อย จากนั้นทำทุกหัวข้อย่อยที่ได้จากทั้ง 2 คำอธิบายรายวิชามาดำเนินการระบุถึงคำศัพท์เฉพาะที่ปรากฏขึ้นในแต่ละหัวข้อย่อย โดยการนำคลังคำศัพท์เฉพาะเข้ามาช่วยในการระบุ เมื่อทำการระบุถึงคำศัพท์เฉพาะเสร็จเรียบร้อยแล้ว ต่อมาในภาพที่ 38 จะทำการประยุกต์ใช้กฎทางภาษาศาสตร์ฉบับปรับปรุง เข้ามาใช้ในการสกัดคำสำคัญในแต่ละหัวข้อย่อยของทั้ง 2 คำอธิบายรายวิชา สุดท้ายนำคำสำคัญแต่ละคำที่สกัดได้ในแต่ละหัวข้อย่อยของคำอธิบายรายวิชาตั้งต้นมาเปรียบเทียบกับทุกคำสำคัญที่ได้จากคำอธิบายรายวิชาเปรียบเทียบ โดยวิธีการเปรียบเทียบทั้ง 4 วิธีการ จากตัวอย่างจะพบว่าคำสำคัญในหัวข้อย่อยที่ 1 ของคำอธิบายรายวิชาตั้งต้นคือ ‘descriptive(JJ) statistics(TE)’ มีความเหมือนกันกับคำสำคัญในหัวข้อย่อยที่ 1 ของคำอธิบายรายวิชาเปรียบเทียบคือ ‘statistics(TE)’ ด้วยวิธีการเปรียบเทียบแบบเซตย่อย/ซูเปอร์เซต แต่เนื่องจากทั้ง 2 คำสำคัญถูกเปรียบเทียบกันด้วยคำที่เป็นชื่อรายวิชา (“Probability and statistics”) ดังนั้นระบบ *eCSCDA* จึงไม่นับว่าคำสำคัญทั้ง 2 คำเหมือนกัน ทำให้หัวข้อย่อยที่ 1 ของคำอธิบายรายวิชาตั้งต้นไม่มีความเหมือนกันกับหัวข้อย่อยที่ 1 ของคำอธิบายรายวิชาเปรียบเทียบ ในทางกลับกันคำสำคัญในหัวข้อย่อยที่ 6 ของคำอธิบายรายวิชาตั้งต้นคือ ‘estimation(NN)’ มีความเหมือนกับคำสำคัญในหัวข้อย่อยที่ 6 ของคำอธิบายรายวิชาที่นำมาเปรียบเทียบคือ ‘estimation(NN)’ ด้วยวิธีการเปรียบเทียบแบบตรง เมื่อพิจารณาวิธีการเปรียบเทียบแบบเซตย่อย/ซูเปอร์เซตจะพบว่าคำสำคัญในหัวข้อย่อยที่ 4 ของคำอธิบายรายวิชาตั้งต้นคือ ‘discrete(TE) probability distribution(TE)’ และ ‘continuous(JJ) probability distribution(TE)’ มีความเหมือนกับคำสำคัญในหัวข้อย่อยที่ 3 ของคำอธิบายรายวิชาเปรียบเทียบคือ ‘probability distribution(TE)’ ต่อมาในวิธีการเปรียบเทียบแบบองค์ประกอบร่วมจะพบว่าคำสำคัญในหัวข้อย่อยที่ 4 ของคำอธิบายรายวิชาตั้งต้นคือ ‘random variable(TE)’ มีความคล้ายกันกับคำสำคัญในหัวข้อย่อยที่ 5 ของคำอธิบายรายวิชาเปรียบเทียบคือ ‘random distribution(TE)’ และในวิธีการเปรียบเทียบเชิงความหมายจะพบว่าคำสำคัญในหัวข้อย่อยที่ 3 ของคำอธิบายรายวิชาตั้งต้นคือ ‘probability(TE) principle(NN)’ มีความคล้ายกันกับคำสำคัญในหัวข้อย่อยที่ 3 ของคำอธิบายรายวิชาเปรียบเทียบคือ ‘probability(TE) theory(TE)’ เมื่อดำเนินการเปรียบเทียบครบทุกคำสำคัญในทุกประโยคของคำอธิบายรายวิชาตั้งต้นแล้ว ทำให้ได้มาซึ่งส่วนของเนื้อหาที่มีความเหมือนกันกับคำอธิบายรายวิชาเปรียบเทียบทั้งสิ้น 6 หัวข้อ ได้แก่ หัวข้อย่อยที่ 3 “principle of probability”, หัวข้อย่อยที่ 4 “ discrete and continuous probability distribution of random variables and computational problems”, หัวข้อย่อยที่ 5 “statistical distribution”, หัวข้อย่อย ที่ 6 “estimation”, หัวข้อย่อยที่ 7 “experiment design and hypothesis testing” และ หัวข้อย่อย ที่

8 “correlation and linear regression analysis” และสามารถนำมาคำนวณหาอัตราร้อยละของความเหมือนกันได้เท่ากับ 60%

บทที่ 5
ผลการดำเนินงาน

บทนี้จะอธิบายถึงผลของการดำเนินงานของระบบ *CSCDA* และระบบ *eCSCDA* ที่ได้นำเสนอไว้ในบทที่ 3 และ 4 ตามลำดับ โดยในการทดลองผู้วิจัยได้ทำการรวบรวมคำอธิบายรายวิชา (เฉพาะส่วนที่เป็นภาษาอังกฤษ) ทั้งสิ้น 607 คำอธิบายรายวิชา จากหลักสูตรวิทยาการคอมพิวเตอร์ของทั้ง 10 มหาวิทยาลัย ซึ่งได้แก่ 1) มหาวิทยาลัยบูรพา (Burapha University), 2) มหาวิทยาลัยเชียงใหม่ (Chiang Mai University), 3) จุฬาลงกรณ์มหาวิทยาลัย (Chulalongkorn University), 4) สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (King Mongkut's Institute of Technology Ladkrabang), 5) มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ (King Mongkut's University of Technology North Bangkok), 6) มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี (King Mongkut's University of Technology Thonburi), 7) มหาวิทยาลัยเกษตรศาสตร์ (Kasetsart University), 8) มหาวิทยาลัยมหิดล (Mahidol University), 9) มหาวิทยาลัยสงขลานครินทร์ (Prince of Songkla University) และ 10) มหาวิทยาลัยธรรมศาสตร์ (Thammasat University) โดยแต่ละมหาวิทยาลัยมีจำนวนรายวิชาในหลักสูตรวิทยาการคอมพิวเตอร์ดังตารางที่ 4

ตารางที่ 4 จำนวนรายวิชาจากหลักสูตรวิทยาการคอมพิวเตอร์ของ 10 มหาวิทยาลัย

ชื่อมหาวิทยาลัย	จำนวนรายวิชา
Burapha University (BUU)	66
Chiang Mai University (CMU)	63
Chulalongkorn University (CU)	47
King Mongkut's Institute of Technology Ladkrabang (KMITL)	87
King Mongkut's University of Technology North Bangkok (KMUTNB)	55
King Mongkut's University of Technology Thonburi (KMUTT)	40
Kasetsart University (KU)	69
Mahidol University (MU)	35
Prince of Songkla University (PSU)	60

ตารางที่ 4 จำนวนรายวิชาจากหลักสูตรวิทยาการคอมพิวเตอร์ของ 10 มหาวิทยาลัย (ต่อ)

ชื่อมหาวิทยาลัย	จำนวนรายวิชา
Thammasat University (TU)	85
จำนวนทั้งหมด	607

ในการประเมินประสิทธิภาพของทั้ง 2 ระบบผู้วิจัยจะแบ่งออกเป็น 2 ส่วน คือ 1) การประเมินประสิทธิภาพการสกัดคำสำคัญ เป็นการประเมินประสิทธิภาพของคำสำคัญที่สกัดได้จาก 4 ขั้นตอนวิธี ได้แก่ คำสำคัญที่สกัดได้จากขั้นตอนวิธีของ *eCSCDA*, *CSCDA*, *TerMine* และ *RAKE* และ ส่วนที่ 2) การประเมินประสิทธิภาพของการเปรียบเทียบคำสำคัญ เป็นการประเมินประสิทธิภาพของการเปรียบเทียบคำสำคัญจาก 4 ขั้นตอนวิธี คือ การเปรียบเทียบคำสำคัญด้วยขั้นตอนวิธีของ *eCSCDA*, *CSCDA*, *spaCy* และ *BERT* ซึ่งแต่ละส่วนการประเมินประสิทธิภาพ จะมีรายละเอียดดังต่อไปนี้

5.1 การประเมินประสิทธิภาพการสกัดคำสำคัญ

การประเมินประสิทธิภาพการสกัดคำสำคัญจะเป็นการประเมินเชิงเปรียบเทียบ โดยทำการเปรียบเทียบวิธีการที่นำเสนอกับวิธีการสกัดคำสำคัญที่ได้ได้รับความนิยมและถูกใช้งานอย่างแพร่หลาย ได้แก่

1) ขั้นตอนวิธี “*TerMine*” (Frantzi, Ananiadou, & Mima, 2000) เป็นวิธีการสกัดคำสำคัญ โดยการสร้างรายการแคนดิเดทของกลุ่มคำที่จะถูกสกัดจากประโยคหนึ่ง ๆ จากการประยุกต์ใช้วิธีการทางภาษาศาสตร์ ซึ่งกลุ่มคำที่จะถูกพิจารณาคือกลุ่มของนามวลี (Noun phrase) กล่าวคือ กลุ่มของ คำคุณศัพท์ + คำนาม, คำกริยา + คำนาม, คำนาม + คำนาม, คำคุณศัพท์ + คำนาม + คำนาม เป็นต้น โดยกลุ่มคำที่ถูกพิจารณาจะถูกระบุเป็นคำสำคัญภายในรายการแคนดิเดท ต่อมาทำการนับความถี่ของคำสำคัญภายในรายการแคนดิเดทโดยใช้วิธีการทางสถิติ จากนั้นนำผู้เชี่ยวชาญเข้ามาประเมินถึงคำสำคัญที่ได้แต่ละคำในรายการแคนดิเดทคำสำคัญ เพื่อทำการสกัดคำสำคัญออกมาเป็นผลลัพธ์ที่ได้ และ

2) ขั้นตอนวิธี “*RAKE* (Rapid automatic keyword extraction)” (Rose, Engel, Cramer, & Cowley, 2010) เป็นวิธีการสกัดคำสำคัญโดยอัตโนมัติจากบทคัดย่อภาษาอังกฤษ โดยการนำวิธีการทางภาษาศาสตร์เข้ามาใช้ในการพิจารณาสกัดคำสำคัญ โดยมีวิธีการดังนี้ 1) พิจารณาคำที่อยู่ติดกันไปเรื่อย ๆ จนกว่าจะเจอคำหยุด แล้วจึงนำคำ

เหล่านั้นมาสร้างรายการแคนดิเดทของคำสำคัญ 2) ทำการสร้างเมทริกซ์ ของแต่ละคำสำคัญที่ปรากฏในรายการแคนดิเดทคำสำคัญ เพื่อคำนวณคะแนนความถี่ของคำที่เกิดขึ้นร่วมกัน 3) รวมคะแนนของคำสำคัญที่ปรากฏในรายการแคนดิเดทคำสำคัญ และทำการเรียงลำดับคำสำคัญในรายการแคนดิเดทคำสำคัญจากคำสำคัญที่มีคะแนนมากไปหาคำสำคัญที่มีคะแนนน้อย และ 4) พิจารณา top-T scoring โดยจะสนใจจำนวนเพียง 1 ใน 3 ของคำสำคัญจากรายการแคนดิเดทคำสำคัญที่ถูกเรียงลำดับ

โดยในส่วนของการประเมินประเมินประสิทธิภาพการสกัดคำสำคัญ ผู้วิจัยได้ทำการประยุต์ใช้วิธีการประเมินในด้านของ จำนวนคำสำคัญที่สกัดได้, ความแม่นยำโดยรวม (Accuracy), ความแม่นยำ (Precision), ความถูกต้อง (Recall) และ ประสิทธิภาพโดยรวม (F-measure) (Tharwat, 2020) เพื่อใช้ในการประเมินประสิทธิภาพของการสกัดคำสำคัญในแต่ละวิธีการ โดยวิธีการประเมินในด้านของความแม่นยำ, ความถูกต้อง และ ประสิทธิภาพโดยรวม สามารถคำนวณได้จากข้อมูลภายในตารางการประเมินประสิทธิภาพความถูกต้อง (Confusion matrix) ระหว่างข้อมูลคำสำคัญที่เกิดขึ้นจริงและข้อมูลคำสำคัญที่สกัดได้ ดังตารางที่ 5

ตารางที่ 5 ตารางการประเมินประสิทธิภาพความถูกต้องระหว่างข้อมูลคำสำคัญที่เกิดขึ้นจริงและข้อมูลคำสำคัญที่สกัดได้

		ข้อมูลคำสำคัญที่เกิดขึ้นจริง	
		คำสำคัญที่ถูกต้อง	คำสำคัญที่ไม่ถูกต้อง
ข้อมูลคำสำคัญที่สกัดได้	คำสำคัญที่ถูกสกัด	*TP*	*FP*
	คำสำคัญที่ไม่ถูกสกัด	*FN*	*TN*

โดยที่ TP (True Positive) คือ คำสำคัญที่ถูกสกัดได้ และ เป็นคำสำคัญที่ถูกต้อง

TN (True Negative) คือ คำสำคัญที่ไม่ถูกสกัด และ เป็นคำสำคัญที่ไม่ถูกต้อง

FP (False Positive) คือ คำสำคัญที่ถูกสกัดได้ แต่ เป็นคำสำคัญที่ไม่ถูกต้อง

FN (False Negative) คือ คำสำคัญที่ไม่ถูกสกัด แต่ เป็นคำสำคัญที่ถูกต้อง

จากตารางที่ 5 สามารถนำมาคำนวณหาค่าของ ความแม่นยำโดยรวม (Accuracy), ความแม่นยำ (Precision), ความถูกต้อง (Recall) และ ประสิทธิภาพโดยรวม (F-measure) ได้ดังต่อไปนี้

$$\text{Accuracy} = \frac{TP}{TP + TN + FP + FN} \quad (6)$$

$$Precision = \frac{TP}{TP + FP} \quad (7)$$

$$Recall = \frac{TP}{TP + FN} \quad (8)$$

$$F - measure = 2 \times \frac{precision \times recall}{preciosion + recall} \quad (9)$$

การประเมินประสิทธิภาพการสกัดคำสำคัญจะทำการเปรียบเทียบระหว่าง 4 ขั้นตอนวิธี คือ *eCSCDA*, *CSCDA*, *TerMine* และ *RAKE* โดยในแต่ละขั้นตอนวิธีจะทำการประเมินผลทั้ง 5 ด้าน ซึ่งจะแสดงผลดังตารางที่ 6, 7, 8, 9 และ 10 ตามลำดับ

ตารางที่ 6 จำนวนคำสำคัญที่สกัดได้และจำนวนคำสำคัญที่มีความถูกต้องของ 4 ขั้นตอนวิธี

มหาวิทยาลัย (จำนวนคำสำคัญที่ถูกต้อง)	จำนวนคำสำคัญที่สกัดได้				จำนวนคำสำคัญที่สกัดได้ถูกต้อง			
	eCSCDA	*CSCDA*	*TerMine*	*RAKE*	*eCSCDA*	*CSCDA*	*TerMine*	*RAKE*
BUU(967)	967	1,297	525	1,142	965	774	338	632
CMU(641)	641	820	390	732	639	525	286	447
CU(414)	415	568	262	474	407	345	179	283
KMITL(1,103)	1,103	1,487	681	1,313	1,096	887	429	726
KMUTNB(730)	727	1,095	485	896	720	537	280	452
KMUTT(627)	623	808	353	734	621	513	250	433
KU(759)	758	1,001	443	901	756	613	311	529
MU(554)	554	678	242	614	547	456	184	409
PSU(756)	750	1,000	480	867	747	616	348	524
TU(985)	985	1,386	578	1,195	975	756	344	638
จำนวน(7,536)	7,523	10,140	4,439	8,868	7,473	6,022	2,949	5,073

ตารางที่ 7 การเปรียบเทียบค่าความแม่นยำโดยรวม (Accuracy) ของ 4 ขั้นตอนวิธี

มหาวิทยาลัย(จำนวนคำสำคัญที่ถูกต้อง)	ค่าความแม่นยำโดยรวม (Accuracy)			
	eCSCDA	*CSCDA*	*TerMine*	*RAKE*
BUU(967)	99.59	51.95	29.29	42.79
CMU(641)	99.38	56.09	38.39	48.27
CU(414)	96.45	54.16	36.02	36.02
KMITL(1,103)	98.74	52.08	31.66	42.96
KMUTNB(730)	97.79	40.28	30.35	38.72
KMUTT(627)	98.73	55.64	34.25	46.66
KU(759)	99.34	53.44	34.90	46.77
MU(554)	97.50	58.76	30.07	53.89
PSU(756)	98.42	54.04	39.19	47.68
TU(985)	97.99	46.81	28.22	41.37
ค่าเฉลี่ย	98.39	52.33	33.23	44.51

ตารางที่ 8 การเปรียบเทียบค่าความแม่นยำ (Precision) ของ 4 ขั้นตอนวิธี

มหาวิทยาลัย(จำนวนคำสำคัญที่ถูกต้อง)	ค่าความแม่นยำ (Precision)			
	eCSCDA	*CSCDA*	*TerMine*	*RAKE*
BUU(967)	99.79	59.68	64.38	55.34
CMU(641)	99.69	64.02	73.33	61.07
CU(414)	98.07	60.74	68.32	59.7
KMITL(1,103)	99.37	59.65	63	55.29
KMUTNB(730)	98.45	49.08	59.05	50.45
KMUTT(627)	99.68	63.49	70.82	58.99
KU(759)	99.74	61.24	70.2	58.71
MU(554)	98.74	67.26	76.03	66.61
PSU(756)	99.6	61.6	72.5	60.44
TU (985)	98.98	54.55	59.52	53.39
ค่าเฉลี่ย	99.21	60.13	67.72	58.00

ตารางที่ 9 การเปรียบเทียบค่าความถูกต้อง (Recall) ของ 4 ขั้นตอนวิธี

มหาวิทยาลัย(จำนวนคำสำคัญที่ถูกต้อง)	ค่าความถูกต้อง (Recall)			
	eCSCDA	*CSCDA*	*TerMine*	*RAKE*
BUU(967)	99.79	80.04	34.95	65.36
CMU(641)	99.69	81.9	44.62	69.73
CU(414)	98.31	83.33	43.24	68.36
KMITL(1,103)	99.37	80.42	38.89	65.82
KMUTNB(730)	98.32	73.45	38.66	61.33
KMUTT(627)	99.04	81.82	39.87	69.06
KU(759)	99.6	80.76	40.97	69.7
MU(554)	98.74	82.31	33.21	73.83
PSU(756)	98.81	81.48	46.03	69.31
TU(985)	98.98	76.75	34.92	64.77
ค่าเฉลี่ย	99.07	80.23	39.54	67.73

ตารางที่ 10 การเปรียบเทียบค่าร้อยละของประสิทธิภาพโดยรวม (F-measure)

มหาวิทยาลัย(จำนวนคำสำคัญที่ถูกต้อง)	ค่าความถูกต้อง (Recall)			
	eCSCDA	*CSCDA*	*TerMine*	*RAKE*
BUU(967)	99.79	68.37	45.31	59.93
CMU(641)	99.69	71.87	55.48	65.11
CU(414)	98.19	70.26	52.96	63.74
KMITL(1,103)	99.37	68.49	48.09	60.1
KMUTNB(730)	98.75	60.46	46.32	55.13
KMUTT(627)	99.36	71.5	51.02	63.63
KU(759)	99.67	69.66	51.75	63.73
MU(554)	98.74	74.03	46.23	70.03
PSU(756)	99.2	70.16	56.31	54.57
TU(985)	98.98	63.77	44.02	58.53
ค่าเฉลี่ย	99.17	68.86	49.75	61.45

จากตารางที่ 6 แสดงให้เห็นถึงจำนวนของคำสำคัญที่สกัดได้ในแต่ละขั้นตอนวิธี ได้แก่ การสกัดคำสำคัญจากขั้นตอนวิธีของ *eCSCDA* ได้ 7,523 คำ, *CSCDA* ได้ 10,140 คำ, *TerMine* ได้ 4,439 คำ และ *RAKE* ได้ 8,868 คำ เมื่อพิจารณาถึงคำสำคัญที่สกัดได้ในแต่ละขั้นตอนวิธีจะพบว่าการสกัดคำสำคัญด้วยขั้นตอนวิธีของ *CSCDA* และ *RAKE* ได้จำนวนของคำสำคัญเป็นอันดับที่ 1 และ 2 ตามลำดับ เนื่องจากคำสำคัญที่สกัดได้ในแต่ละหัวข้อย่อยจากทั้ง 2 ขั้นตอนวิธีมีความซ้ำซ้อนกันอยู่มาก ต่อมาเมื่อพิจารณาถึงจำนวนคำสำคัญที่สกัดได้จากขั้นตอนวิธีของ *TerMine* พบว่าได้จำนวนคำสำคัญน้อยที่สุดจากทั้ง 4 ขั้นตอนวิธี เนื่องจากขั้นตอนวิธีของ *TerMine* ทำการพิจารณาสกัดคำสำคัญจากกลุ่มของคำนามที่ปรากฏในแต่ละหัวข้อย่อยเท่านั้น จึงทำให้คำสำคัญที่สกัดได้มีจำนวนน้อยกว่าทุกขั้นตอนวิธี ในทางกลับกัน เมื่อทำการพิจารณาถึงจำนวนของคำสำคัญที่ถูกต้องทั้งหมด (7,536 คำ) เทียบกับคำสำคัญที่สกัดได้ถูกต้องในแต่ละขั้นตอนวิธีพบว่า จำนวนคำสำคัญที่สกัดได้จากขั้นตอนวิธีของ *eCSCDA* มีจำนวนคำสำคัญที่ถูกต้องใกล้เคียงกับจำนวนคำสำคัญที่ถูกต้องทั้งหมดคือ 7,473 คำ เพราะว่าคำสำคัญที่สกัดได้จากขั้นตอนวิธีของ *eCSCDA* จะไม่มีคำสำคัญที่ซ้ำซ้อนกันเกิดขึ้น ลำดับต่อมา คือ ขั้นตอนวิธีของ *CSCDA* ได้ 6,022 คำ, *RAKE* ได้ 5,073 คำ และ *TerMine* ได้ 2,949 คำ จากการที่ขั้นตอนวิธีของ *eCSCDA* สามารถสกัดคำสำคัญที่มีความถูกต้องได้มากที่สุด ยังส่งผลให้การประเมินในด้านของความแม่นยำโดยรวมในตารางที่ 7 แสดงให้เห็นว่าการสกัดคำสำคัญจากขั้นตอนวิธีของ *eCSCDA* มีค่าเฉลี่ยของความแม่นยำโดยรวมมากกว่าการสกัดคำสำคัญอีก 3 ขั้นตอนวิธี กล่าวคือ การสกัดคำสำคัญจากขั้นตอนวิธีของ *eCSCDA* ได้ 98.39%, *CSCDA* ได้ 52.33%, *TerMine* ได้ 33.23% และ *RAKE* ได้ 44.51%

ตารางที่ 8 และ 9 จะแสดงให้เห็นการประเมินประสิทธิภาพของการสกัดคำสำคัญของทั้ง 4 ขั้นตอนวิธี ในด้านของความแม่นยำและความถูกต้องตามลำดับ จากพิจารณาถึงความแม่นยำของการสกัดคำสำคัญในแต่ละขั้นตอนวิธีพบว่า ขั้นตอนวิธีของ *eCSCDA* มีค่าเฉลี่ยของความแม่นยำสูงที่สุด (*eCSCDA* ได้ 99.21%, CSCDA ได้ 60.13%, *TerMine* ได้ 67.72% และ *RAKE* ได้ 58.00%) และในด้านของความถูกต้องพบว่า ขั้นตอนวิธีการของ *eCSCDA* มีค่าเฉลี่ยของความถูกต้องสูงที่สุดด้วยเช่นกัน (*eCSCDA* ได้ 99.07%, CSCDA ได้ 80.23%, *TerMine* ได้ 39.54% และ *RAKE* ได้ 67.73%) จากการที่การสกัดคำสำคัญของขั้นตอนวิธีของ *eCSCDA* ได้ทั้งค่าเฉลี่ยของความแม่นยำและความถูกต้องสูงที่สุด ทำให้เมื่อพิจารณาต่อมาในด้านของประสิทธิภาพโดยรวม ในตารางที่ 10 พบว่า ค่าเฉลี่ยของประสิทธิภาพโดยรวมของการสกัดคำสำคัญในขั้นตอนวิธีของ *eCSCDA* มีค่าสูงที่สุดเช่นเดียวกัน (*eCSCDA* ได้ 99.17%, *CSCDA* ได้ 68.86%, *TerMine* ได้ 49.75% และ *RAKE* ได้

61.45%) จากที่การสกัดคำสำคัญในขั้นตอนวิธีของ *eCSCDA* มีค่าเฉลี่ยของความแม่นยำโดยรวม, ความแม่นยำ, ความถูกต้อง และ ประสิทธิภาพโดยรวมสูงกว่าทุกขั้นตอนวิธี เพราะคำสำคัญที่สกัดได้จากขั้นตอนวิธีของ *eCSCDA* จะไม่มีคำสำคัญที่ซ้ำซ้อนกันเกิดขึ้น อีกทั้งในการนำคำมาเชื่อมต่อกันด้วยกฎทางภาษาศาสตร์ฉบับปรับปรุง ทำให้คำสำคัญที่สกัดได้ มีความถูกต้องและมีความครอบคลุมเนื้อหามากที่สุด

5.2 การประเมินประสิทธิภาพของการเปรียบเทียบคำสำคัญ

การประเมินประสิทธิภาพของการเปรียบเทียบคำสำคัญ จะเป็นการประเมินเชิงเปรียบเทียบเช่นเดียวกับการประเมินในหัวข้อที่ 5.1 โดยจะทำการเปรียบเทียบวิธีการที่นำเสนอกับวิธีการเปรียบเทียบข้อความเชิงความหมาย ได้แก่ วิธีการเปรียบเทียบโดยขั้นตอนวิธีของ *spaCy* และ *BERT* (Bidirectional Encoder Representation from Transformers) ซึ่งทั้ง 2 วิธีการจะเริ่มจากการสร้างเวคเตอร์ของคำที่นำมาเปรียบเทียบกัน โดยเวคเตอร์ของคำที่สร้างขึ้นจะถูกสร้างจากคลังคำศัพท์ของแต่ละวิธีการที่ได้จัดเตรียมไว้ ซึ่งคลังคำศัพท์ของแต่ละวิธีการก็จะมีการสร้างที่แตกต่างกันออกไป กล่าวคือ คลังคำศัพท์ของขั้นตอนวิธี *spaCy* จะเป็นคลังคำศัพท์ที่ถูกสร้างขึ้นมาโดยการรวบรวมคำศัพท์จากแหล่งต่าง ๆ ด้วยตัวของผู้พัฒนาเอง ในทางกลับกัน คลังคำศัพท์ของขั้นตอนวิธี *BERT* จะเป็นการสร้างจากการประยุกต์ใช้คลังคำศัพท์ของ BookCorpus และ คลังคำศัพท์ของ English Wikipedia มารวมกันเพื่อนำมาสร้างเป็นคลังคำศัพท์ของตนเอง จากการได้มาซึ่งเวคเตอร์ของคำแต่ละคำที่เปรียบทียบกันแล้ว จะนำเวคเตอร์เหล่านั้นมาคำนวณหาค่าของความเหมือนกันระหว่างคำจากผลลัพธ์ของวิธีการเปรียบเทียบทั้ง 2 ขั้นตอนวิธีข้างต้น ทำให้ได้มาซึ่งคู่ของคำสำคัญที่เหมือนกันและค่าของความเหมือนกัน ดังนั้นผู้วิจัยจึงได้มีการตั้งเกณฑ์ในการพิจารณาความเหมือนกันจากค่าความเหมือนที่ได้ในแต่ละคู่คำสำคัญ โดยได้กำหนดค่าขีดแบ่ง (Threshold) ของค่าความเหมือนกันตั้งแต่ 80% ขึ้นไป ที่จะถูกนำมาพิจารณาเป็นผลลัพธ์สุดท้ายจากการเปรียบเทียบของทั้ง 2 วิธี

โดยในส่วนของการประเมินประสิทธิภาพของการเปรียบเทียบคำสำคัญ ผู้วิจัยได้ทำการประยุต์ใช้วิธีการประเมินประสิทธิภาพในด้านของ อัตราร้อยละ (Percentage), ความแม่นยำ (Precision), ความถูกต้อง (Recall) และ ประสิทธิภาพโดยรวม (F-measure) เพื่อใช้ในการประเมินประสิทธิภาพในแต่ละขั้นตอนวิธี โดยวิธีการประเมินในด้านของความแม่นยำ, ความถูกต้อง และประสิทธิภาพโดยรวม สามารถคำนวณได้จากข้อมูลภายในตารางการประเมินประสิทธิภาพความถูกต้อง ระหว่างข้อมูลของการเปรียบเทียบคำสำคัญที่เกิดขึ้นจริงและข้อมูลของการเปรียบเทียบคำสำคัญที่ได้ ดังตารางที่ 11

ตารางที่ 11 ตารางการประเมินประสิทธิภาพความถูกต้องระหว่างข้อมูลการเปรียบเทียบคำสำคัญที่เกิดขึ้นจริงและข้อมูลการเปรียบเทียบคำสำคัญที่ได้

		ข้อมูลการเปรียบเทียบคำสำคัญที่เกิดขึ้นจริง	
		คำสำคัญที่เปรียบเทียบแล้วถูกต้อง	คำสำคัญที่เปรียบเทียบแล้วไม่ถูกต้อง
ข้อมูลการเปรียบเทียบคำสำคัญที่ได้	คำสำคัญที่ถูกเปรียบเทียบ	*TP*	*FP*
	คำสำคัญที่ไม่ถูกเปรียบเทียบ	*FN*	*TN*

โดยที่ TP (True Positive) คือ คำสำคัญที่ถูกเปรียบเทียบ และ เป็นคำสำคัญที่เปรียบเทียบแล้วถูกต้อง

TN (True Negative) คือ คำสำคัญที่ไม่ถูกเปรียบเทียบ และ เป็นคำสำคัญที่เปรียบเทียบแล้วไม่ถูกต้อง

FP (False Positive) คือ คำสำคัญที่ถูกเปรียบเทียบ แต่ เป็นคำสำคัญที่เปรียบเทียบแล้วไม่ถูกต้อง

FN (False Negative) คือ คำสำคัญที่ไม่ถูกเปรียบเทียบ แต่ เป็นคำสำคัญที่เปรียบเทียบแล้วถูกต้อง

โดยการคำนวณหาอัตราร้อยละ (Percentage) สามารถทำการประยุกต์ใช้สมการ $sim(c_{s,x}, c_{s,y})$ ในส่วนของบทที่ 3 หรือ สมการ $per\_sim(c_x, c_y)$ ในส่วนของบทที่ 4 และการคำนวณหาค่าความแม่นยำ (Precision), ความถูกต้อง (Recall) และ ประสิทธิภาพโดยรวม (F-measure) สามารถคำนวณได้จากสมการที่ 7, 8 และ 9 ตามลำดับ

การประเมินประสิทธิภาพของการเปรียบเทียบคำสำคัญจะทำการเปรียบเทียบระหว่าง 4 ขั้นตอนวิธี คือ *eCSCDA*, *CSCDA*, *spaCy* และ *BERT* โดยในแต่ละขั้นตอนวิธีจะทำการประเมินผลทั้ง 4 ด้าน ซึ่งจะแสดงผลดังตารางที่ 12, 13, 14 และ 15 ตามลำดับ

ตารางที่ 12 การเปรียบเทียบอัตราร้อยละ (Percentage) ของ 4 ขั้นตอนวิธี

มหาวิทยาลัย (จำนวนรายวิชาเปรียบเทียบ)	อัตราร้อยละ (Percentage)			
	eCSCDA	*CSCDA*	*spaCy*	*BERT*
CMU(15)	34.82	31.09	41.28	80.95
CU(16)	28.45	21.97	48.67	73.84
KMITL(19)	32.10	25.27	48.10	69.63
KMUTNB(19)	29.66	24.74	29.79	78.44
KMUTT(16)	43.06	34.87	46.91	82.98
KU(20)	37.94	27.94	47.94	72.32
MU(11)	32.27	34.86	47.46	76.20
PSU(22)	35.38	26.68	51.41	80.18
TU(21)	32.27	23.35	38.44	76.91
ค่าเฉลี่ย	33.99	27.86	44.44	76.82

ตารางที่ 13 การเปรียบเทียบค่าความแม่นยำ (Precision) ของ 4 ขั้นตอนวิธี

มหาวิทยาลัย (จำนวนรายวิชาเปรียบเทียบ)	ความแม่นยำ (Precision)			
	eCSCDA	*CSCDA*	*spaCy*	*BERT*
CMU(15)	97.14	96.80	37.88	30.46
CU(16)	79.76	71.81	37.46	25.93
KMITL(19)	81.80	80.33	45.18	28.90
KMUTNB(19)	80.33	79.21	50.52	24.88
KMUTT(16)	88.16	77.87	32.93	32.31
KU(20)	90.63	84.62	31.90	27.64
MU(11)	85.56	65.14	39.45	29.50
PSU(22)	94.85	86.91	40.87	28.18
TU(21)	80.86	76.66	50.47	29.83
ค่าเฉลี่ย	86.57	79.93	40.74	28.62

ตารางที่ 14 การเปรียบเทียบค่าความถูกต้อง (Recall) ของ 4 ขั้นตอนวิธี

มหาวิทยาลัย(จำนวนรายวิชาเปรียบเทียบ)	ความถูกต้อง (Recall)			
	eCSCDA	*CSCDA*	*spaCy*	*BERT*
CMU(15)	100.00	98.33	45.50	72.45
CU(16)	83.33	79.20	55.49	63.31
KMITL(19)	91.30	77.22	56.89	64.20
KMUTNB(19)	84.74	84.21	55.70	65.61
KMUTT(16)	94.74	83.36	39.83	64.71
KU(20)	95.24	88.27	50.84	62.25
MU(11)	93.75	90.95	40.10	57.22
PSU(22)	100.00	97.64	52.63	65.83
TU(21)	85.19	79.41	49.78	59.80
ค่าเฉลี่ย	91.04	85.03	49.64	63.93

ตารางที่ 15 การเปรียบเทียบค่าประสิทธิภาพโดยรวม (F-measure) ของ 4 ขั้นตอนวิธี

มหาวิทยาลัย (จำนวนรายวิชาเปรียบเทียบ)	ประสิทธิภาพโดยรวม (F-measure)			
	eCSCDA	*CSCDA*	*spaCy*	*BERT*
CMU(15)	98.37	95.53	36.07	41.71
CU(16)	81.32	71.41	41.32	33.70
KMITL(19)	85.27	77.72	45.70	37.54
KMUTNB(19)	82.12	81.34	48.87	34.09
KMUTT(16)	90.97	79.18	35.42	40.73
KU(20)	92.58	85.97	36.29	37.24
MU(11)	88.62	74.95	44.06	36.90
PSU(22)	96.91	90.98	46.93	37.55
TU(21)	82.56	76.97	46.90	38.23
ค่าเฉลี่ย	87.54	79.82	42.39	37.52

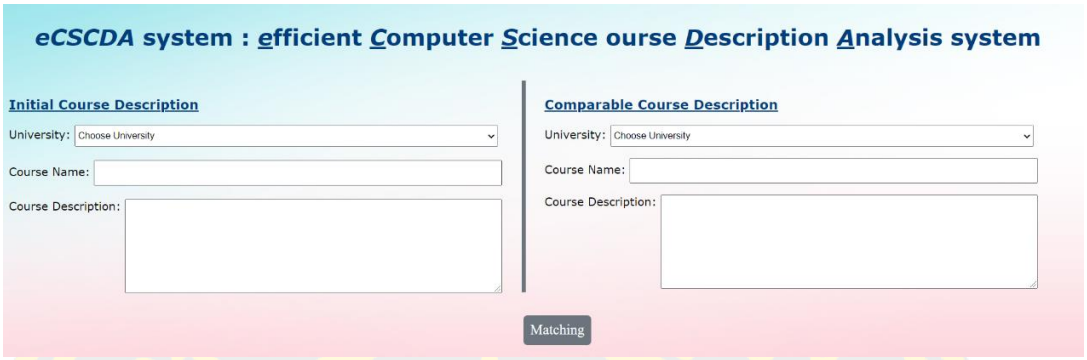
จากตารางที่ 12, 13, 14 และ 15 แสดงให้เห็นถึงการประเมินประสิทธิภาพในการเปรียบเทียบคำสำคัญด้วยขั้นตอนวิธีของ *eCSCDA*, *CSCDA*, *spaCy* และ *BERT* ซึ่งเป็นการดำเนินการเปรียบเทียบระหว่างคำอธิบายรายวิชาของมหาวิทยาลัยบูรพา (กำหนดให้เป็นคำอธิบายรายวิชาตั้งต้น) และ คำอธิบายวิชาของอีก 9 มหาวิทยาลัย (กำหนดให้เป็นคำอธิบายรายวิชาเปรียบเทียบ) โดยมีรายวิชาของมหาวิทยาลัยบูรพาที่สอนเหมือนกันกับมหาวิทยาลัยต่าง ๆ ดังนี้ มี 15 รายวิชาของมหาวิทยาลัยเชียงใหม่, 16 รายวิชาของจุฬาลงกรณ์มหาวิทยาลัย, 19 รายวิชาของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 19 มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ, 16 รายวิชาของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี, 20 รายวิชาของมหาวิทยาลัยเกษตรศาสตร์, 11 รายวิชาของมหาวิทยาลัยมหิดล, 22 รายวิชาของมหาวิทยาลัยสงขลานครินทร์ และ 21 รายวิชาของมหาวิทยาลัยธรรมศาสตร์ ที่สอนเหมือนกันกับมหาวิทยาลัยบูรพา โดนการประเมินประสิทธิภาพในด้านอัตราร้อยละจากตารางที่ 12 พบว่าการเปรียบเทียบคำสำคัญด้วยขั้นตอนวิธีของ *BERT* (76.82%) มีค่าเฉลี่ยของอัตราร้อยละที่เหนือกว่าขั้นตอนวิธีของ *eCSCDA* (33.99%), *CSCDA* (27.86%) และ *spaCy* (44.44%) กล่าวคือ การเปรียบเทียบคำสำคัญด้วยขั้นตอนวิธีของ *BERT* สามารถให้ได้ผลลัพธ์จากการเปรียบเทียบคำสำคัญมากที่สุด เนื่องจากคลังคำศัพท์ที่ใช้ในการเปรียบเทียบเชิงความหมายเป็นคลังคำศัพท์ขนาดใหญ่ที่มีคำศัพท์อยู่ทั้งหมด 3,200 ล้านคำ ในทางกลับกัน เมื่อทำการพิจารณาในด้านของความแม่นยำ, ความถูกต้อง และ ประสิทธิภาพโดยรวม ดังตารางที่ 13, 14 และ 15 ตามลำดับ พบว่าการเปรียบเทียบคำสำคัญด้วยขั้นตอนวิธีของ *eCSCDA* ให้ประสิทธิภาพในทั้ง 3 ด้านที่เหนือกว่าทุก ๆ ขั้นตอนวิธี คือ ด้านความแม่นยำมีค่าเฉลี่ยของแต่ละขั้นตอนวิธี ดังนี้ ขั้นตอนวิธีของ *eCSCDA* ได้ 85.57%, *CSCDA* ได้ 79.93%, *spaCy* ได้ 40.74%, และ *BERT* ได้ 28.62% ด้านความถูกต้องมีค่าเฉลี่ยของแต่ละขั้นตอนวิธี ดังนี้ ขั้นตอนวิธีของ *eCSCDA* ได้ 91.04%, *CSCDA* ได้ 85.03%, *spaCy* ได้ 49.64%, และ *BERT* ได้ 63.93% และ ด้านประสิทธิภาพโดยรวมมีค่าเฉลี่ยของแต่ละขั้นตอนวิธี ดังนี้ ขั้นตอนวิธีของ *eCSCDA* ได้ 87.54%, *CSCDA* ได้ 79.82%, *spaCy* ได้ 42.39%, และ *BERT* ได้ 37.52% โดยสาเหตุที่ทำให้ขั้นตอนวิธีของ *eCSCDA* มีประสิทธิภาพทั้ง 3 ด้านที่เหนือกว่าทุกขั้นตอนวิธี เนื่องจากในการเปรียบเทียบเชิงความหมายจะทำการประยุกต์ใช้คลังคำพ้องความหมายของคำศัพท์เฉพาะ และ คลังคำพ้องความหมายของคำศัพท์ทั่วไป ที่ผ่านการตรวจสอบความเหมาะสมทางการพ้องความหมายจากทั้ง 3 พจนานุกรมมอนไลน์ อีกทั้งยังมีวิธีการเปรียบเทียบคำสำคัญที่ได้ปรับปรุงและพัฒนาขึ้นมาใหม่ ทำให้เมื่อพิจารณาถึงเนื้อหาของหัวข้อย่อยทั้ง 2 ที่ถูกเปรียบเทียบผ่านคู่ของคำสำคัญพบว่ามีความถูกต้องมากกว่าทุกขั้นตอนวิธี

จากผลลัพธ์ของการประเมินประสิทธิภาพทั้ง 2 ส่วนในข้างต้นที่กล่าวมา ได้แก่ การประเมินประสิทธิภาพการสกัดคำสำคัญ เป็นการประเมินประสิทธิภาพของวิธีการสกัดคำสำคัญโดย 4 ขั้นตอนวิธี คือ ขั้นตอนวิธีของ *eCSCDA*, *CSCDA*, *TerMine* และ *RAKE* ซึ่งเมื่อทำการพิจารณาถึงประสิทธิภาพของการสกัดคำสำคัญในแต่ละขั้นตอนวิธี จะพบว่าวิธีการสกัดคำสำคัญด้วยขั้นตอนวิธีของ *eCSCDA* มีประสิทธิภาพที่เหนือกว่าวิธีการสกัดคำสำคัญด้วยขั้นตอนวิธีของ *CSCDA*, *TerMine* และ *RAKE* โดยมีประสิทธิภาพที่เหนือว่าในทุก ๆ ด้าน ได้แก่ จำนวนคำสำคัญที่สกัดได้เมื่อเทียบกับคำสำคัญที่ถูกต้อง, ความแม่นยำโดยรวม, ความแม่นยำ, ความถูกต้อง และ ประสิทธิภาพโดยรวม และ การประเมินประสิทธิภาพของการเปรียบเทียบคำสำคัญ โดยทำการประเมินผลการเปรียบเทียบคำสำคัญด้วย 4 ขั้นตอนวิธี คือ ขั้นตอนวิธีของ *eCSCDA*, *CSCDA*, *spaCy* และ *BERT* ซึ่งเป็นการประเมินประสิทธิภาพในด้านของอัตราร้อยละ, ความแม่นยำ, ความถูกต้อง และ ประสิทธิภาพโดยรวม เมื่อทำการพิจารณาถึงการประเมินประสิทธิภาพในแต่ละด้านจะพบว่า ในด้านของอัตราร้อยละการเปรียบเทียบคำสำคัญด้วยขั้นตอนวิธีของ *BERT* มีประสิทธิภาพที่เหนือกว่าการเปรียบเทียบคำสำคัญจาก 3 ขั้นตอนวิธี แต่ในทางกลับกัน เมื่อพิจารณาถึงประสิทธิภาพในด้านของความแม่นยำ, ความถูกต้อง และ ประสิทธิภาพโดยรวม จะพบว่าการเปรียบเทียบคำสำคัญด้วยขั้นตอนวิธีการของ *eCSCDA* ให้ประสิทธิภาพในทั้ง 3 ด้านที่เหนือกว่าการเปรียบเทียบคำสำคัญอีก 3 ขั้นตอนวิธี ซึ่งจากการประเมินประสิทธิภาพทั้ง 2 ส่วนที่กล่าวมาข้างต้น จะพบว่าทั้งวิธีการสกัดคำสำคัญและวิธีการเปรียบเทียบคำสำคัญด้วยขั้นตอนวิธีของ *eCSCDA* มีประสิทธิภาพที่ดีที่สุดเมื่อเทียบกับขั้นตอนวิธีต่าง ๆ ในการสกัดคำสำคัญและการเปรียบเทียบคำสำคัญ และยังส่งผลให้ผลลัพธ์ที่ได้จากการดำเนินงานของระบบ *eCSCDA* มีประสิทธิภาพมากที่สุด

5.3 การใช้งานระบบวิเคราะห์คำอธิบายรายวิชาในหลักสูตรวิทยาการคอมพิวเตอร์

ในส่วนนี้จะกล่าวถึงระบบวิเคราะห์คำอธิบายรายวิชาในหลักสูตรวิทยาการคอมพิวเตอร์ที่ผู้วิจัยได้พัฒนาขึ้น โดยระบบนี้จะเป็นส่วนช่วยให้คณะกรรมการและอาจารย์ประจำรายวิชาของมหาวิทยาลัยหนึ่ง ๆ ทราบถึงส่วนของเนื้อหาในคำอธิบายรายวิชาของตนเองที่เหมือนและแตกต่างกับคำอธิบายรายวิชาที่เป็นรายวิชาเดียวกันในมหาวิทยาลัยอื่น ๆ ได้ เพื่อเป็นส่วนช่วยในการตัดสินใจสำหรับการพัฒนาและปรับปรุงเนื้อหาที่จะถูกสอนหรือถ่ายทอดให้กับนิสิต/นักศึกษา ให้มีความเหมาะสมและสอดคล้องกับเทคโนโลยีในปัจจุบันมากยิ่งขึ้น ซึ่งระบบ *eCSCDA* ที่สร้างขึ้นประกอบด้วยส่วนของข้อมูลนำเข้า 2 ส่วน คือ 1) ข้อมูลของคำอธิบายรายวิชาตั้งต้น (ฝั่งซ้าย) และ 2) ข้อมูลของคำอธิบายรายวิชาเปรียบเทียบ (ฝั่งขวา) โดยทั้ง 2 ส่วนจะรับข้อมูลนำเข้า ได้แก่ 1) ชื่อ

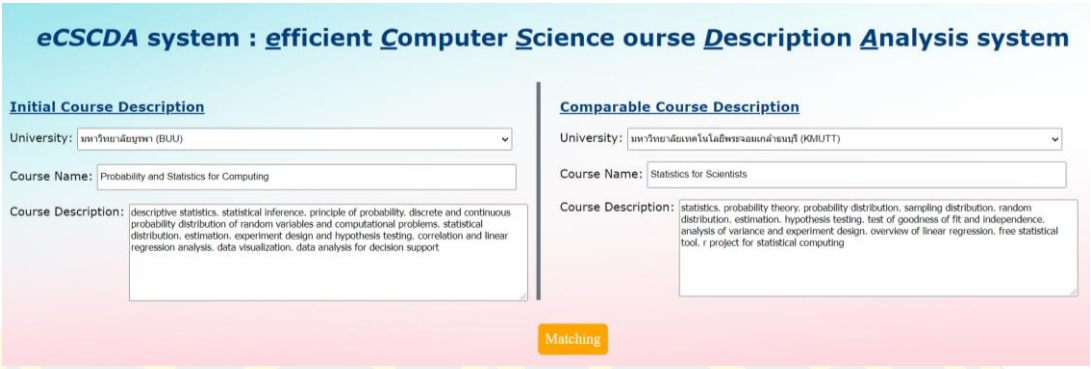
มหาวิทยาลัย (ภาษาอังกฤษ), 2) ชื่อรายวิชา (ภาษาอังกฤษ) และ 3) คำอธิบายรายวิชา (ภาษาอังกฤษ) แสดงดังภาพที่ 39



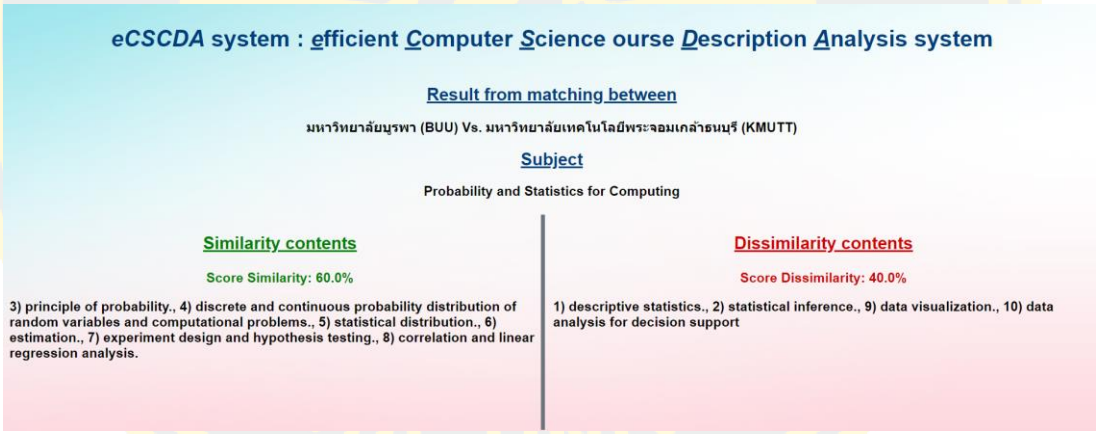
ภาพที่ 39 หน้าต่างของระบบวิเคราะห์คำอธิบายรายวิชาในหลักสูตรวิทยาการคอมพิวเตอร์

เมื่อต้องการใช้งานระบบ *eCSCDA* ผู้ใช้งานสามารถใส่ข้อมูลในส่วนของคำอธิบายรายวิชาตั้งต้น โดยมีข้อมูลที่ต้องใส่ คือ 1) ชื่อมหาวิทยาลัยของคำอธิบายรายวิชาตั้งต้น, 2) ชื่อรายวิชาของคำอธิบายรายวิชาตั้งต้น (ภาษาอังกฤษ) และ 3) คำอธิบายรายวิชาตั้งต้น (ภาษาอังกฤษ) และสามารถใส่ข้อมูลในส่วนของคำอธิบายรายวิชาเปรียบเทียบ โดยมีข้อมูลที่ต้องใส่ คือ 1) ชื่อมหาวิทยาลัยของคำอธิบายรายวิชาเปรียบเทียบ, 2) ชื่อรายวิชาของคำอธิบายรายวิชาเปรียบเทียบ (ภาษาอังกฤษ) และ 3) คำอธิบายรายวิชาเปรียบเทียบ (ภาษาอังกฤษ) เมื่อทำการใส่ข้อมูลครบทั้ง 2 ส่วนแล้ว จากนั้นกดปุ่ม “Matching” ระบบจะดำเนินการเปรียบเทียบคำอธิบายรายวิชาทั้ง 2 คำอธิบายรายวิชาที่ผู้ใช้งานใส่ข้อมูลเข้ามา ซึ่งมีตัวอย่างการใส่ข้อมูลเพื่อดำเนินการวิเคราะห์และเปรียบเทียบคำอธิบายรายวิชา ดังภาพที่ 40 โดยมีส่วนของข้อมูลคำอธิบายรายวิชาตั้งต้น 1) ชื่อมหาวิทยาลัยของคำอธิบายรายวิชาตั้งต้น คือ มหาวิทยาลัยบูรพา (BUU), 2) ชื่อรายวิชาของคำอธิบายรายวิชาตั้งต้น คือ “Probability and Statistics for Computing” และ 3) คำอธิบายรายวิชาตั้งต้น คือ “Descriptive statistics, statistical inference, principle of probability, discrete and continuous probability distribution of random variables and computational problems, statistical distribution, estimation, experiment design and hypothesis testing, correlation and linear regression analysis, data visualization, data analysis for decision support” และส่วนของข้อมูลคำอธิบายรายวิชาเปรียบเทียบ ได้แก่ 1) ชื่อมหาวิทยาลัยของคำอธิบายรายวิชาเปรียบเทียบ คือ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี (KMUTT), 2) ชื่อรายวิชาของคำอธิบายรายวิชาเปรียบเทียบ คือ “Statistics for scientists” และ 3) คำอธิบายรายวิชาเปรียบเทียบ คือ “ Statistics, probability theory, probability distribution, sampling distribution, random

distribution, estimation, hypothesis testing, test of goodness of fit and independence, analysis of variance and experiment design, overview of linear regression, free statistical tool, r project for statistical computing” จากนั้นกดปุ่ม “Matching”



ภาพที่ 40 ตัวอย่างการใส่ข้อมูลคำอธิบายรายวิชาตั้งต้นและคำอธิบายรายวิชาเปรียบเทียบในระบบวิเคราะห์คำอธิบายรายวิชาในหลักสูตรวิทยาการคอมพิวเตอร์

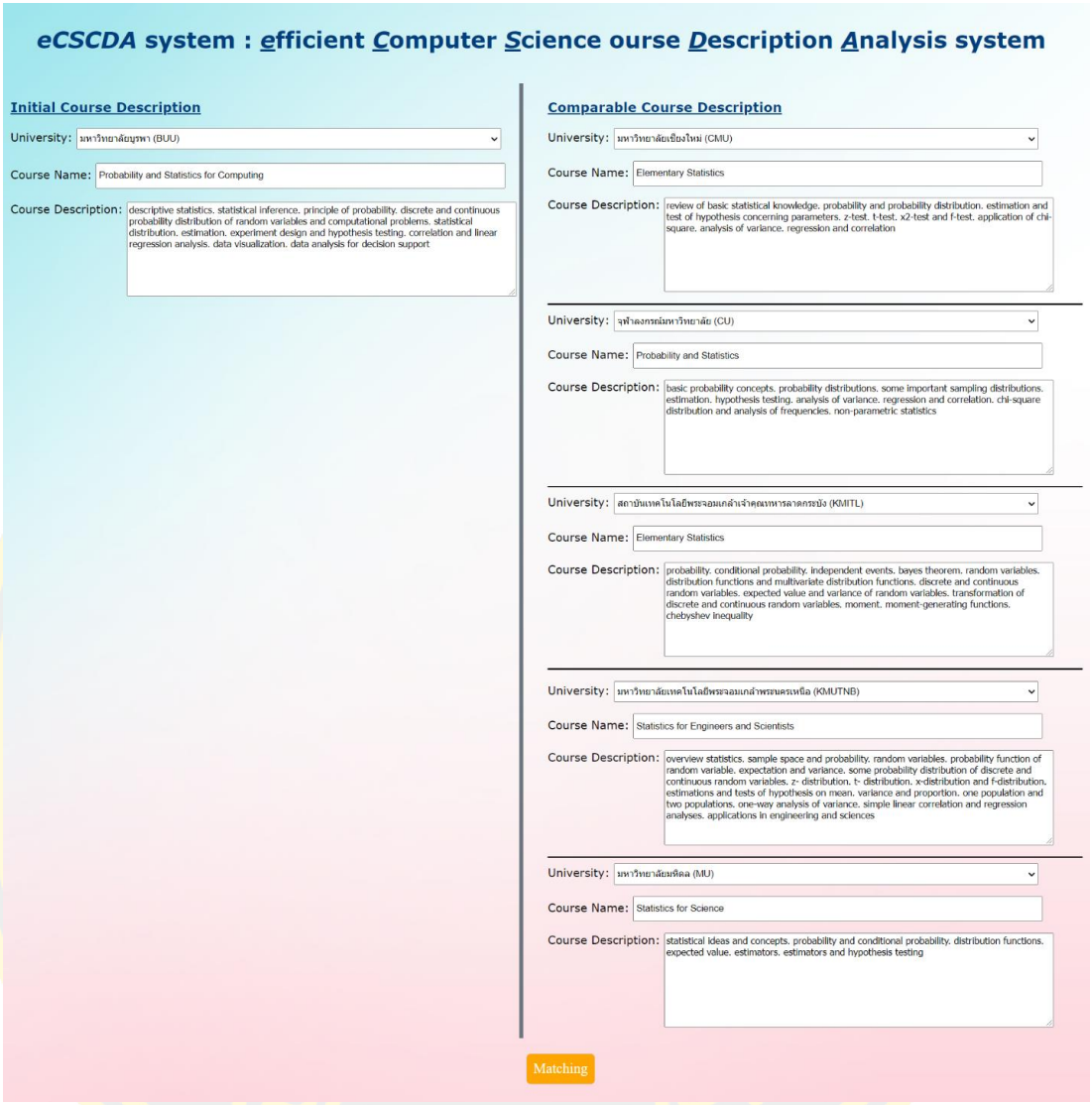


ภาพที่ 41 ผลลัพธ์จากการเปรียบเทียบระหว่างคำอธิบายรายวิชาตั้งต้นและคำอธิบายรายวิชาเปรียบเทียบของระบบวิเคราะห์คำอธิบายรายวิชาในหลักสูตรวิทยาการคอมพิวเตอร์

จากภาพที่ 41 เป็นส่วนของหน้าต่างที่แสดงให้เห็นถึงผลลัพธ์จากการดำเนินการเปรียบเทียบระหว่างข้อมูลคำอธิบายรายวิชาตั้งต้นและข้อมูลคำอธิบายรายวิชาเปรียบเทียบ โดยผลลัพธ์ที่ได้จะแสดงให้เห็นถึงส่วนของเนื้อหาที่เหมือนกันซึ่งประกอบไปด้วย อัตราร้อยละและเนื้อหาของคำอธิบายรายวิชาตั้งต้นที่เหมือนกับเนื้อหาของคำอธิบายรายวิชาที่เปรียบเทียบ และส่วนของเนื้อหาที่แตกต่างกันซึ่งประกอบไปด้วย อัตราร้อยละและเนื้อหาของคำอธิบายรายวิชาตั้งต้นที่แตกต่างกันกับเนื้อหาของคำอธิบายรายวิชาเปรียบเทียบ ซึ่งเมื่อพิจารณาในภาพที่ 41 จะพบว่าเป็นผลลัพธ์ที่ได้จากการเปรียบเทียบระหว่างข้อมูลคำอธิบายรายวิชาตั้งต้นของมหาวิทยาลัยบูรพา (BUU) กับข้อมูลคำอธิบาย

รายวิชาเปรียบเทียบของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี (KMUTT) โดยในส่วนของเนื้อหาที่เหมือนกันจะบ่งบอกถึงอัตราร้อยละของความเหมือนกันที่ 60.0% และมีเนื้อที่เหมือนกัน ได้แก่ หัวข้อย่อยที่ 3) “principle of probability.”, หัวข้อย่อยที่ 4) “discrete and continuous probability distribution of random variables and computational problems.”, หัวข้อย่อยที่ 5) “statistical distribution.”, หัวข้อย่อยที่ 6) “estimation.”, หัวข้อย่อยที่ 7) “experiment design and hypothesis testing.” และ หัวข้อย่อยที่ 8) “correlation and linear regression analysis.” และในส่วนของเนื้อหาที่แตกต่างกันซึ่งจะบ่งบอกถึงอัตราร้อยละความแตกต่างกันที่ 40.0% และมีเนื้อที่แตกต่างกัน ได้แก่ หัวข้อย่อยที่ 1) “Descriptive statistics.”, หัวข้อย่อยที่ 2) “statistical inference.”, หัวข้อย่อยที่ 9) “data visualization.” และ หัวข้อย่อยที่ 10) “data analysis for decision support.”

นอกจากนี้ผู้วิจัยยังได้ออกแบบให้ระบบ *eCSCDA* สามารถดำเนินการเปรียบเทียบระหว่าง 1 ข้อมูลคำอธิบายรายวิชาตั้งต้นกับหลายข้อมูลคำอธิบายรายวิชาเปรียบเทียบได้ ดังภาพที่ 42

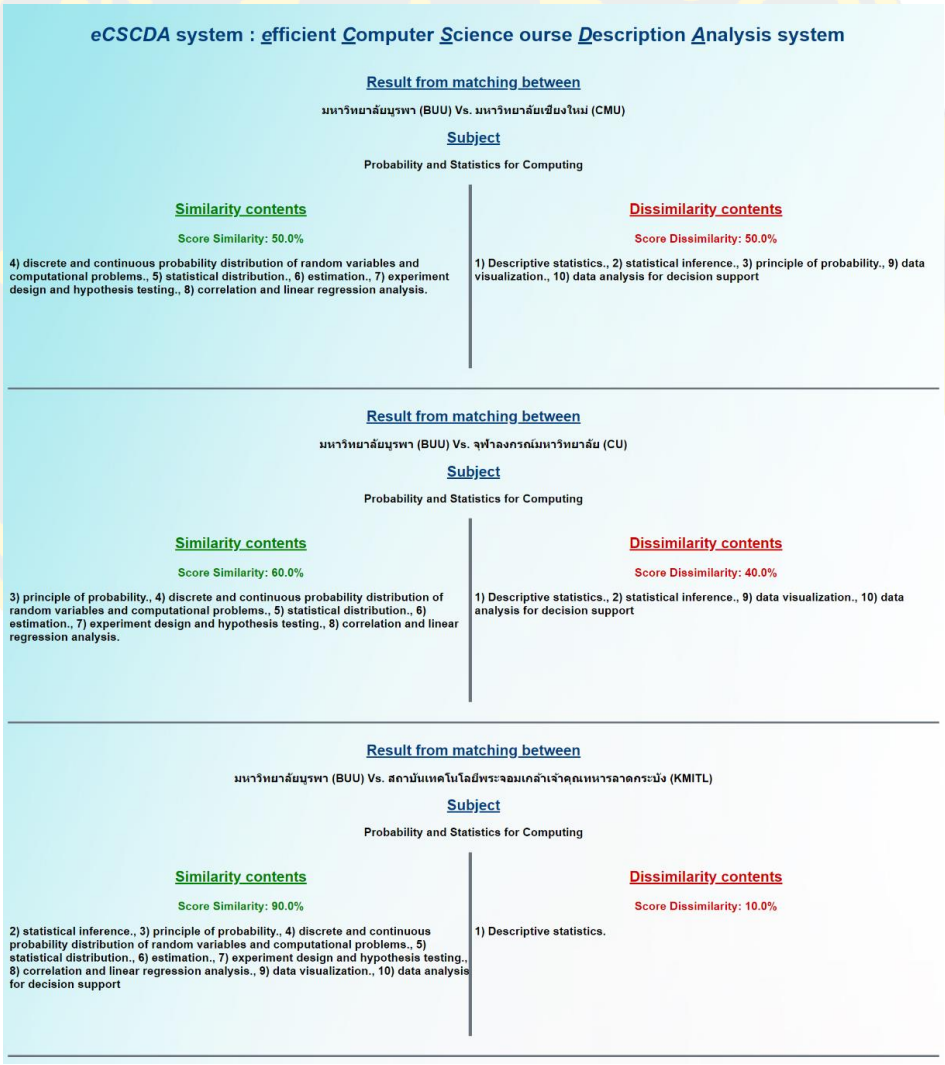


ภาพที่ 42 ตัวอย่างการใส่ข้อมูลสำหรับการเปรียบเทียบระหว่าง 1 คำอธิบายรายวิชาตั้งต้นกับหลายคำอธิบายรายวิชาเปรียบเทียบ

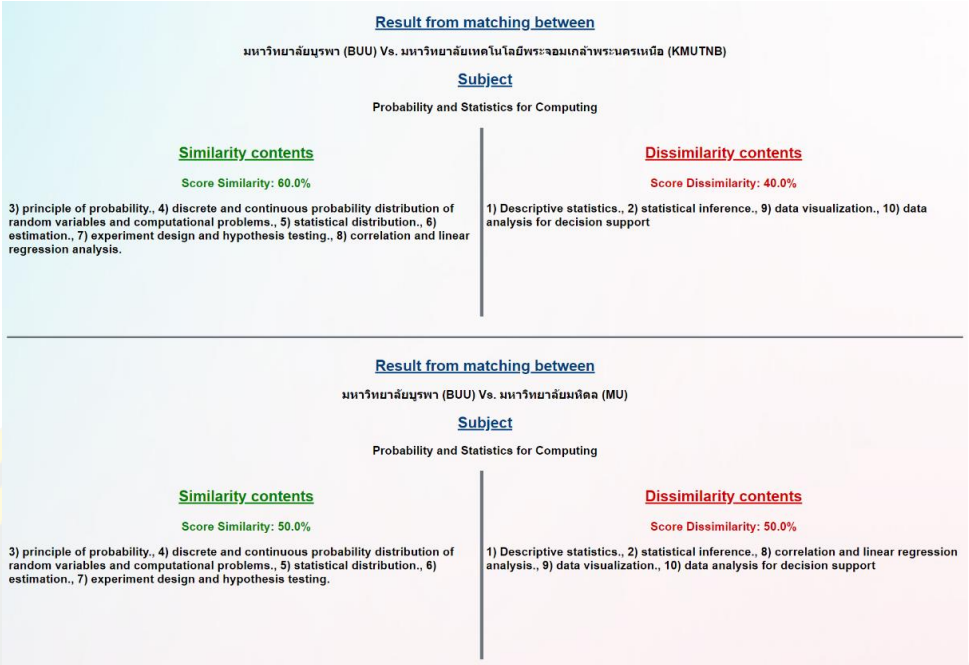
จากภาพที่ 42 แสดงให้เห็นถึงการใส่ข้อมูลในลักษณะของการเปรียบเทียบแบบ 1 ข้อมูลคำอธิบายรายวิชาตั้งต้นกับ 5 ข้อมูลคำอธิบายรายวิชาเปรียบเทียบ โดยมีส่วนของข้อมูลคำอธิบายรายวิชาตั้งต้น ได้แก่ 1) ชื่อมหาวิทยาลัยของคำอธิบายรายวิชาตั้งต้น คือ มหาวิทยาลัยบูรพา (BUU), 2) ชื่อรายวิชาของคำอธิบายรายวิชาตั้งต้น คือ “Probability and Statistics for Computing” และ 3) คำอธิบายรายวิชาตั้งต้น คือ “Descriptive statistics, statistical inference, principle of probability, discrete and continuous probability distribution of random variables and computational problems, statistical distribution, estimation, experiment design and hypothesis testing, correlation and linear regression analysis, data visualization, data

analysis for decision support” กับส่วนของข้อมูลคำอธิบายรายวิชาเปรียบเทียบของข้อมูลที่ 1 ได้แก่ 1) ชื่อมหาวิทยาลัยของคำอธิบายรายวิชาเปรียบเทียบ คือ มหาวิทยาลัยเชียงใหม่ (CMU), 2) ชื่อรายวิชาของคำอธิบายรายวิชาเปรียบเทียบ คือ “Elementary Statistics” และ 3) คำอธิบายรายวิชาเปรียบเทียบ คือ “Review of basic statistical knowledge. probability and probability distribution. estimation and test of hypothesis concerning parameters. z-test. t-test. x2-test and f-test. application of chi-square. analysis of variance. regression and correlation”, ส่วนของข้อมูลคำอธิบายรายวิชาเปรียบเทียบของข้อมูลที่ 2 ได้แก่ 1) ชื่อมหาวิทยาลัยของคำอธิบายรายวิชาเปรียบเทียบ คือ จุฬาลงกรณ์มหาวิทยาลัย (CU), 2) ชื่อรายวิชาของคำอธิบายรายวิชาเปรียบเทียบ คือ “Probability and Statistics” และ 3) คำอธิบายรายวิชาเปรียบเทียบ คือ “Basic probability concepts. probability distributions. some important sampling distributions. estimation. hypothesis testing. analysis of variance. regression and correlation. chi-square distribution and analysis of frequencies. non-parametric statistics”, ส่วนของข้อมูลคำอธิบายรายวิชาเปรียบเทียบของข้อมูลที่ 3 ได้แก่ 1) ชื่อมหาวิทยาลัยของคำอธิบายรายวิชาเปรียบเทียบ คือ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (KMITL), 2) ชื่อรายวิชาของคำอธิบายรายวิชาเปรียบเทียบ คือ “Elementary Statistics” และ 3) คำอธิบายรายวิชาเปรียบเทียบ คือ “Basic concepts of statistics. probability. random variables. probability distribution. binomial. hypergeometric. introduction to sampling techniques. sampling distribution. estimation. test of hypotheses. simple linear regression and correlation analysis”, ส่วนของข้อมูลคำอธิบายรายวิชาเปรียบเทียบของข้อมูลที่ 4 ได้แก่ 1) ชื่อมหาวิทยาลัยของคำอธิบายรายวิชาเปรียบเทียบ คือ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ (KMUTNB), 2) ชื่อรายวิชาของคำอธิบายรายวิชาเปรียบเทียบ คือ “Statistics for Engineers and Scientists” และ 3) คำอธิบายรายวิชาเปรียบเทียบ คือ “Overview statistics, sample space and probability, random variables, probability function of random variable, expectation and variance, some probability distribution of discrete and continuous random variables, Z- distribution, t-distribution, x-distribution and f-distribution, estimations and tests of hypothesis on mean, variance and proportion in case of one population and two populations, one-way analysis of variance, simple linear correlation and regression analyses and applications in engineering and sciences” และ ส่วนของข้อมูลคำอธิบายรายวิชาเปรียบเทียบ

ของข้อมูลที่ 4 ได้แก่ 1) ชื่อมหาวิทยาลัยของคำอธิบายรายวิชาเปรียบเทียบ คือ มหาวิทยาลัยมหิดล (MU), 2) ชื่อรายวิชาของคำอธิบายรายวิชาเปรียบเทียบ คือ “Statistics for Science” และ 3) คำอธิบายรายวิชาเปรียบเทียบ คือ “Statistical ideas and concepts. probability and conditional probability. distribution functions. expected value. estimators. estimators and hypothesis testing” เมื่อทำการใส่ข้อมูลครบแล้ว จากนั้นกดปุ่ม “Matching” เพื่อดำเนินการเปรียบเทียบระหว่าง 1 ข้อมูลคำอธิบายรายวิชาตั้งต้นกับ 5 ข้อมูลคำอธิบายรายวิชาเปรียบเทียบ โดยผลลัพธ์จากการวิเคราะห์และเปรียเทียบระหว่าง 1 ข้อมูลคำอธิบายรายวิชาตั้งต้นกับ 5 ข้อมูลคำอธิบายรายวิชาเปรียบเทียบ จะแสดงดังภาพที่ 43 และ 44



ภาพที่ 43 ผลลัพธ์จากการเปรียบเทียบระหว่าง 1 ข้อมูลคำอธิบายรายวิชาตั้งต้นกับ 5 ข้อมูลคำอธิบายรายวิชาเปรียบเทียบของระบบวิเคราะห์คำอธิบายรายวิชาในหลักสูตรวิทยาการคอมพิวเตอร์



ภาพที่ 44 ผลลัพธ์จากการเปรียบเทียบระหว่าง 1 ข้อมูลคำอธิบายรายวิชาตั้งต้นกับ 5 ข้อมูลคำอธิบายรายวิชาเปรียบเทียบของระบบวิเคราะห์คำอธิบายรายวิชาในหลักสูตรวิทยาการคอมพิวเตอร์ (ต่อ)

จากภาพที่ 43 และ 44 แสดงให้เห็นถึงผลลัพธ์ทั้งหมด 5 คู่ จากการดำเนินการเปรียบเทียบระหว่าง 1 ข้อมูลคำอธิบายรายวิชาตั้งต้นกับ 5 ข้อมูลคำอธิบายรายวิชาเปรียบเทียบ โดยที่ในแต่ละคู่จะแสดงให้เห็นถึงส่วนของเนื้อหาที่เหมือนกันซึ่งประกอบไปด้วย อัตราร้อยละและเนื้อหาของคำอธิบายรายวิชาตั้งต้นที่เหมือนกับเนื้อหาของคำอธิบายรายวิชาเปรียบเทียบ และส่วนของเนื้อหาที่แตกต่างกันซึ่งประกอบไปด้วย อัตราร้อยละและเนื้อหาของคำอธิบายรายวิชาตั้งต้นที่แตกต่างกันกับเนื้อหาของคำอธิบายรายวิชาเปรียบเทียบ อาทิเช่น ผลลัพธ์คู่ที่ 1 จะเป็นผลลัพธ์ที่ได้จากการเปรียบเทียบระหว่างข้อมูลคำอธิบายรายวิชาตั้งต้นของมหาวิทยาลัยบูรพา (BUU) กับข้อมูลคำอธิบายรายวิชาเปรียบเทียบของมหาวิทยาลัยเชียงใหม่ (CMU) โดยในส่วนของเนื้อหาที่เหมือนกันจะบ่งบอกถึงอัตราร้อยละของความเหมือนกันที่ 50.0% และมีเนื้อที่เหมือนกัน ได้แก่ หัวข้อย่อยที่ 4) “ discrete and continuous probability distribution of random variables and computational problems.”, หัวข้อย่อยที่ 5) “statistical distribution.”, หัวข้อย่อยที่ 6) “estimation.”, หัวข้อย่อยที่ 7) “experiment design and hypothesis testing.” และ หัวข้อย่อยที่ 8) “correlation and linear regression analysis.” และในส่วนของเนื้อหาที่แตกต่างกันจะบ่งขอกถึงอัตราร้อยละความแตกต่างกันที่ 50.0% และมีเนื้อที่แตกต่างกัน ได้แก่ หัวข้อย่อยที่ 1)

“Descriptive statistics.”, หัวข้อย่อยที่ 2) “statistical inference.”, หัวข้อย่อยที่3) “principle of probability.”, หัวข้อย่อยที่ 9) “data visualization.” และ หัวข้อย่อยที่ 10) “data analysis for decision support” เป็นต้น

บทที่ 6
สรุปและอภิปรายผล

6.1 สรุปผลการดำเนินงาน

งานวิจัยนี้มีวัตถุประสงค์ในการมุ่งเน้นเพื่อพัฒนาระบบสำหรับการวิเคราะห์คำอธิบายรายวิชาในหลักสูตรวิทยาการคอมพิวเตอร์ ซึ่งจะเป็นส่วนช่วยให้คณะกรรมการและอาจารย์ประจำรายวิชาของมหาวิทยาลัยหนึ่ง ๆ ทราบถึงส่วนของเนื้อหาในคำอธิบายรายวิชา ของตนเองที่เหมือนและแตกต่างกับคำอธิบายรายวิชาที่เป็นรายวิชาเดียวกันในมหาวิทยาลัยต่าง ๆ ได้ อันนำมาซึ่งการเป็นส่วนช่วยในการพัฒนาและปรับปรุงเนื้อหาของคำอธิบายรายวิชาในรายวิชาหนึ่ง ๆ ให้มาตรฐานขององค์ความรู้ที่จะถูกสอนหรือถ่ายทอด ให้กับนิสิต/นักศึกษามีความครบถ้วนสมบูรณ์และมีความสอดคล้องกับเทคโนโลยีในปัจจุบันมากยิ่งขึ้น อีกทั้งยังช่วยลดความเหลื่อมล้ำทางด้านคุณภาพของบัณฑิต (ในสาขาวิทยาการคอมพิวเตอร์) ที่จบจากมหาวิทยาลัยต่างที่แตกต่างกัน แล้วยังส่งเสริมให้บัณฑิตมีคุณภาพเพียงพอต่อการทำงานในภาคอุตสาหกรรม นอกจากนี้ ยังเป็นส่วนช่วยให้นิสิต/นักศึกษาสามารถทราบถึงเนื้อหาที่นอกเหนือจากสิ่งที่ได้เรียนรู้ในชั้นเรียน โดยการพิจารณาถึงส่วนของเนื้อหาในคำอธิบายรายวิชาของมหาวิทยาลัยต่าง ๆ ที่แตกต่างกับเนื้อหาในคำอธิบายรายวิชาของมหาวิทยาลัยตนเอง

จากวัตถุประสงค์ที่ได้กล่าวมาข้างต้น ผู้วิจัยจึงได้ทำการพัฒนา “ระบบวิเคราะห์คำอธิบายรายวิชาในหลักสูตรวิทยาการคอมพิวเตอร์ (*Computer Science Course Description Analysis system)*” หรือเรียกว่า “ระบบ *CSCDA”* ซึ่งเป็นระบบสำหรับการวิเคราะห์และเปรียบเทียบระหว่างคำอธิบายรายวิชาของรายวิชาหนึ่ง ๆ ในศาสตร์ทางด้านวิทยาการคอมพิวเตอร์ ซึ่งคำอธิบายรายวิชาที่สามารถนำมาวิเคราะห์และเปรียบเทียบกัน จะต้องเป็นคำอธิบายของรายวิชาที่เหมือนหรือสอดคล้องเท่านั้น โดยขั้นตอนการดำเนินงานของระบบ *CSCDA* ประกอบด้วย 3 ขั้นตอนการทำงาน คือ i) การรวบรวมข้อมูลนำเข้า (รวบรวมคำอธิบายรายวิชา, รวบรวบคำศัพท์เฉพาะ และ รวบรวมกฎทางภาษาศาสตร์) ii) การสกัดคำสำคัญจากคำอธิบายรายวิชา (การประมวลผลข้อความเบื้องต้น, การระบุคำศัพท์เฉพาะ, การสกัดคำสำคัญ และ การลดทอนเนื้อหาที่ไม่สำคัญ) และ iii) การเปรียบเทียบคำสำคัญระหว่างคำอธิบายรายวิชา (วิธีการเปรียบเทียบแบบตรงตัว, วิธีการเปรียบเทียบแบบเซตย่อย และ วิธีการเปรียบเทียบแบบซูเปอร์เซต)

โดยผลลัพธ์จากการเปรียบเทียบระหว่างคำอธิบายรายวิชาในรายวิชาหนึ่ง ๆ ด้วยระบบ *CSCDA* จะทำให้ทราบถึงข้อมูล 2 ส่วน คือ 1) ส่วนของข้อมูลที่ประกอบด้วยเนื้อหาและอัตราร้อยละของคำอธิบายรายวิชาตั้งต้นที่เหมือนกันกับคำอธิบายรายวิชาเปรียบเทียบ และ 2) ส่วนของข้อมูลที่ประกอบด้วยเนื้อหาและอัตราร้อยละ ของคำอธิบายรายวิชาตั้งต้นที่แตกต่างกันกับคำอธิบายรายวิชาเปรียบเทียบ จากการทราบถึงข้อมูลทั้ง 2 ส่วนทำให้สามารถนำไปประยุกต์ใช้งานได้ในหลาย ๆ ด้าน อาทิเช่น การพัฒนาและปรับปรุงคำอธิบายรายวิชาในรายวิชาหนึ่ง ๆ ให้มีความเหมาะสมกับเทคโนโลยีในปัจจุบันมากยิ่งขึ้น เป็นต้น แต่จากวิธีการเปรียบเทียบของระบบ *CSCDA* ที่เป็นเพียงการเปรียบเทียบในด้านความเหมือนกันของตัวอักษร และลำดับการเกิดขึ้นของตัวอักษรระหว่างคำสำคัญเท่านั้น ทำให้ผลลัพธ์ที่ได้อาจก่อให้เกิดความไม่ครอบคลุมได้ เนื่องจากในบางคำสำคัญที่นำมาเปรียบเทียบกันอาจมีชุดของตัวอักษรที่แตกต่างกันออกไป แต่เมื่อพิจารณาถึงความหมายของคำสำคัญทั้ง 2 คำ อาจพบว่าคำสำคัญทั้ง 2 ให้ความหมายที่เหมือนกัน หรือการที่ไม่สามารถดำเนินการเปรียบเทียบบางส่วนของคำที่เหมือนกันระหว่างคำสำคัญได้

จากปัญหาที่พบในระบบ *CSCDA* ทำให้ผู้วิจัยจำเป็นต้องมีการพัฒนาระบบให้มีประสิทธิภาพมากยิ่งขึ้น โดยระบบที่ได้ทำการพัฒนาต่อยอดขึ้นมาจะถูกเรียกว่า “ระบบวิเคราะห์คำอธิบายรายวิชาในหลักสูตรวิทยาการคอมพิวเตอร์ที่มีประสิทธิภาพ (*An efficient Computer Science Course Description Analysis system*)” หรือเรียกว่า “ระบบ *eCSCDA*” ซึ่งเป็นระบบที่ถูกปรับปรุงและพัฒนาในด้านของการสกัดคำ เพื่อให้ได้คำสำคัญที่มีประสิทธิภาพและความความครอบคลุมเนื้อหามากยิ่งขึ้น นอกจากนี้ ยังมีการปรับปรุงและพัฒนาเพิ่มเติมในส่วนของการเปรียบเทียบระหว่างคำสำคัญ ซึ่งจะเป็นการเปรียบเทียบโดยการพิจารณาถึงองค์ประกอบร่วมและการเปรียบเทียบเชิงความหมายระหว่างคำสำคัญ ทำให้ผลลัพธ์ที่ได้จากการเปรียบเทียบระหว่างคำอธิบายรายวิชาในรายวิชาหนึ่ง ๆ มีประสิทธิภาพและมีความถูกต้องมากยิ่งขึ้น โดยขั้นตอนการดำเนินงานของระบบ *eCSCDA* มีดังนี้ i) การเตรียมข้อมูลนำเข้าและการประมวลผลข้อมูลเบื้องต้น (การประมวลผลข้อมูลเบื้องต้น และ การเตรียมข้อมูลนำเข้า) ii) การสกัดคำสำคัญจากคำอธิบายรายวิชา (การประมวลผลข้อความเบื้องต้น, การระบุคำศัพท์เฉพาะ และ การสกัดคำสำคัญ) และ iii) การเปรียบเทียบคำสำคัญระหว่างคำอธิบายรายวิชา (วิธีการเปรียบเทียบแบบตรงตัว, วิธีการเปรียบเทียบแบบเซตย่อย/ซูเปอร์เซต, วิธีการเปรียบเทียบแบบองค์ประกอบร่วม และ วิธีการเปรียบเทียบเชิงความหมาย)

ในการทดสอบประสิทธิภาพของระบบที่นำเสนอในงานวิจัยนี้ ผู้วิจัยได้ทำการประเมินประสิทธิภาพของการทำงาน 2 ส่วน คือ 1) การสกัดคำสำคัญจากคำอธิบายรายวิชา และ 2) การเปรียบเทียบคำสำคัญระหว่างคำอธิบายรายวิชา โดยแต่ละส่วนมีรายละเอียดดังนี้

1) การประเมินประสิทธิภาพการสกัดคำสำคัญ จะเป็นการประเมินถึงประสิทธิภาพของผลลัพธ์ที่ได้จากการสกัดคำสำคัญทั้ง 4 ขั้นตอนวิธี ได้แก่ ขั้นตอนวิธีของ *eCSCDA, CSCDA, TerMine* และ *RAKE* จากการประเมินประสิทธิภาพของแต่ละขั้นตอนวิธีแสดงให้เห็นว่าผลลัพธ์จากการสกัดคำสำคัญด้วยขั้นตอนวิธีของ *eCSCDA* มีประสิทธิภาพที่เหนือกว่าคำสำคัญที่สกัดได้จากขั้นตอนวิธีอื่น ๆ ทั้งในด้านของ จำนวนคำสำคัญที่สกัดได้เมื่อเทียบกับคำสำคัญที่ถูกต้อง, ความแม่นยำโดยรวม (Accuracy), ความแม่นยำ (Precision), ความถูกต้อง (Recall) และ ประสิทธิภาพโดยรวม (F-measure) ของการสกัดคำสำคัญ จากการที่วิธีการสกัดคำสำคัญด้วยขั้นตอนวิธีของ *eCSCDA* มีประสิทธิภาพที่เหนือกว่าวิธีการสกัดคำสำคัญด้วยขั้นตอนวิธีอื่น ๆ ในทุก ๆ ด้าน ทำให้คำสำคัญที่สกัดได้มีประสิทธิภาพและมีความครอบคลุมเนื้อหามากที่สุด
2) การประเมินประสิทธิภาพของการเปรียบเทียบคำสำคัญ จะเป็นการประเมินถึงประสิทธิภาพของผลลัพธ์ที่ได้จากการเปรียบเทียบคำสำคัญทั้ง 4 ขั้นตอนวิธี ได้แก่ ขั้นตอนวิธีของ *eCSCDA, CSCDA, spaCy* และ *BERT* จากการประเมินประสิทธิภาพของแต่ละขั้นตอนวิธีแสดงให้เห็นว่า ในด้านของอัตราร้อยละความเหมือนกันของเนื้อหา (Percentage) ขั้นตอนวิธีของ *BERT* มีประสิทธิภาพที่เหนือกว่าทั้ง 3 ขั้นตอนวิธี ซึ่งบอกได้ว่าการเปรียบเทียบคำสำคัญด้วยขั้นตอนวิธีของ *BERT* สามารถให้ผลลัพธ์ของคู่คำสำคัญที่มีความเหมือนกันมากที่สุด แต่เมื่อตรวจสอบถึงเนื้อหาของหัวข้อย่อยทั้ง 2 ที่ถูกเปรียบเทียบผ่านคู่ของคำสำคัญพบว่ายังคงมีความไม่ถูกต้องอยู่เป็นจำนวนมาก ในทางกลับกัน เมื่อพิจารณาในด้านของความแม่นยำ (Precision), ความถูกต้อง (Recall) และ ประสิทธิภาพโดยรวม (F-measure) จะพบว่าขั้นตอนวิธีของ *eCSCDA* มีประสิทธิภาพที่เหนือกว่าทุกขั้นตอนวิธี เนื่องจากเมื่อพิจารณาถึงเนื้อหาของหัวข้อย่อยทั้ง 2 ที่ถูกเปรียบเทียบผ่านคู่ของคำสำคัญพบว่ามีความถูกต้องมากกว่าทุกขั้นตอนวิธี ซึ่งแสดงให้เห็นว่าผลลัพธ์ที่ได้จากการเปรียบเทียบด้วยขั้นตอนวิธีของ *eCSCDA* มีประสิทธิภาพและความถูกต้องมากที่สุด

6.2 ข้อเสนอแนะ

1. ระบบที่ผู้วิจัยได้ทำการพัฒนาขึ้นมาเป็นระบบที่มุ่งเน้นในการวิเคราะห์และเปรียบเทียบคำอธิบายรายวิชาในหลักสูตรวิทยาการคอมพิวเตอร์เท่านั้น ในอนาคตผู้ที่สนใจสามารถนำระบบที่ได้ออกแบบไว้ไปประยุต์ใช้ในหลักสูตรอื่น ๆ ได้
2. ระบบที่ผู้วิจัยได้ทำการพัฒนาขึ้นมาเป็นระบบที่มุ่งเน้นในการวิเคราะห์และเปรียบเทียบคำอธิบายรายวิชา ที่มีเนื้อหาของคำอธิบายรายวิชาที่เป็นภาษาอังกฤษเท่านั้น ในอนาคตผู้ที่สนใจอาจนำระบบนี้ไปพัฒนาต่อยอดให้สามารถดำเนินการวิเคราะห์และเปรียบเทียบคำอธิบายรายวิชาในภาษาอื่นๆ ได้

ภาคผนวก

ภาคผนวก ก
กฎทางภาษาศาสตร์สำหรับระบบ *CSCDA*

ตารางที่ 16 กฎทางภาษาศาสตร์ที่ไม่พิจารณาร่วมกับคำศัพท์เฉพาะสำหรับระบบ CSCDA

ลำดับ	รูปแบบหัวข้อ	ผลลัพธ์
1	(W_0 = 'NN' or 'NNS' or 'NNP' or 'NNPS'), ..., (W_n = 'NN' or 'NNS' or 'NNP' or 'NNPS')	• W_0, ..., W_n
2	(W_0 = 'JJ' or 'JJR' or 'JJS') + (W_1 = 'NN' or 'NNS' or 'NNP' or 'NNPS'), ..., (W_n = 'NN' or 'NNS' or 'NNP' or 'NNPS')	• $W_0 + W_1$, ..., W_n
3	(W_0 = 'NN' or 'NNS' or 'NNP' or 'NNPS') + (W_1 = 'IN') + (W_2 = 'NN' or 'NNS' or 'NNP' or 'NNPS') + (W_3 = 'IN') + (W_4 = 'NN' or 'NNS' or 'NNP' or 'NNPS')	• $W_0 + W_1 + W_2$ • $W_2 + W_3 + W_4$
4	(W_0 = 'NN' or 'NNS' or 'NNP' or 'NNPS') + (W_1 = 'NN' or 'NNS' or 'NNP' or 'NNPS') + (W_2 = 'CC') + (W_3 = 'NN' or 'NNS' or 'NNP' or 'NNPS')	• $W_0 + W_1$ • $W_0 + W_3$
5	(W_0 = 'NN' or 'NNS' or 'NNP' or 'NNPS') + (W_1 = 'CC') + (W_2 = 'NN' or 'NNS' or 'NNP' or 'NNPS') + (W_3 = 'NN' or 'NNS' or 'NNP' or 'NNPS')	• $W_0 + W_3$ • $W_2 + W_3$
6	(W_0 = 'NN' or 'NNS' or 'NNP' or 'NNPS') + (W_1 = 'CC') + (W_2 = 'NN' or 'NNS' or 'NNP' or 'NNPS')	• W_0 • W_2
7	(W_0 = 'JJ' or 'JJR' or 'JJS') + (W_1 = 'NN' or 'NNS' or 'NNP' or 'NNPS') + (W_2 = 'CC') + (W_3 = 'JJ' or 'JJR' or 'JJS') + (W_4 = 'NN' or 'NNS' or 'NNP' or 'NNPS')	if $W_1 = W_4$: • $W_0 + W_1$ • $W_3 + W_4$
8	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'IN') + (W_2 = 'NN' or 'NNS' or 'NNP') + (W_3 = 'CC') + (W_4 = 'NN' or 'NNS' or 'NNP')	• $W_0 + W_1 + W_2$ • $W_0 + W_1 + W_4$

ตารางที่ 16 กฎทางภาษาศาสตร์ที่ไม่พิจารณาร่วมกับคำศัพท์เฉพาะสำหรับระบบ CSCDA (ต่อ)

ลำดับ	รูปแบบหัวข้อ	ผลลัพธ์
9	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'CC') + (W_2 = 'NN' or 'NNS' or 'NNP') + (W_3 = 'IN') + (W_4 = 'NN' or 'NNS' or 'NNP')	• $W_0 + W_3 + W_4$ • $W_2 + W_3 + W_4$
10	(W_0 = 'JJ' or 'JJR' or 'JJS') + (W_1 = 'NN' or 'NNS' or 'NNP')	• $W_0 + W_1$
11	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'CC') + (W_2 = 'NN' or 'NNS' or 'NNP') + (W_3 = 'IN') + (W_4 = 'JJ' or 'JJR' or 'JJS') + (W_5 = 'NN' or 'NNS' or 'NNP')	• $W_0 + W_3 + W_4 + W_5$ • $W_2 + W_3 + W_4 + W_5$
12	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'IN') + (W_2 = 'JJ' or 'JJR' or 'JJS') + (W_3 = 'NN' or 'NNS' or 'NNP')	• $W_0 + W_1 + W_2 + W_3$
13	(W_0 = 'JJ' or 'JJR' or 'JJS') + (W_1 = 'NN' or 'NNS' or 'NNP') + (W_2 = 'NN' or 'NNS' or 'NNP') + (W_3 = 'CC') + (W_4 = 'NN' or 'NNS' or 'NNP')	• $W_0 + W_1 + W_2$ • $W_0 + W_1 + W_4$
14	(W_0 = 'JJ' or 'JJR' or 'JJS') + (W_1 = 'CC') + (W_2 = 'JJ' or 'JJR' or 'JJS') + (W_3 = 'NN' or 'NNS' or 'NNP')	• $W_0 + W_1$ • $W_2 + W_3$
15	(W_0 = 'JJ' or 'JJR' or 'JJS') + (W_1 = 'NN' or 'NNS' or 'NNP') + (W_2 = 'CC') + (W_3 = 'NN' or 'NNS' or 'NNP')	• $W_0 + W_1$ • $W_0 + W_3$

ตารางที่ 16 กฎทางภาษาศาสตร์ที่ไม่พิจารณาร่วมกับคำศัพท์เฉพาะสำหรับระบบ CSCDA (ต่อ)

ลำดับ	รูปแบบหัวข้อ	ผลลัพธ์
16	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'CC') + (W_2 = 'NN' or 'NNS' or 'NNP') + (W_3 = 'IN') + (W_4 = 'NN' or 'NNS' or 'NNP')	• $W_0 + W_3 + W_4$ • $W_2 + W_3 + W_4$
17	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'NN' or 'NNS' or 'NNP') + (W_2 = 'CC') + (W_3 = 'NN' or 'NNS' or 'NNP') + (W_4 = 'NN' or 'NNS' or 'NNP')	• $W_0 + W_1$ • $W_3 + W_4$
18	(W_0 = 'JJ' or 'JJR' or 'JJS') + (W_1 = 'NN' or 'NNS' or 'NNP') + (W_2 = 'JJ' or 'JJR' or 'JJS') + (W_3 = 'NN' or 'NNS' or 'NNP')	• $W_0 + W_1$ • $W_2 + W_3$
19	(W_0 = 'JJ' or 'JJR' or 'JJS') + (W_1 = 'JJ' or 'JJR' or 'JJS') + (W_2 = 'NN' or 'NNS' or 'NNP') + (W_3 = 'IN') + (W_4 = 'JJ' or 'JJR' or 'JJS') + (W_5 = 'NN' or 'NNS' or 'NNP') + (W_6 = 'NN' or 'NNS' or 'NNP')	• $W_0 + W_1 + W_2$ • $W_4 + W_5 + W_6$ • $W_0 + W_1 + W_2 + W_3 + W_4 + W_5 + W_6$
20	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'NN' or 'NNS' or 'NNP') + (W_2 = 'IN') + (W_3 = 'NN' or 'NNS' or 'NNP') + (W_4 = 'NN' or 'NNS' or 'NNP')	• $W_0 + W_1 + W_2 + W_3 + W_4$
21	(W_0 = 'JJ' or 'JJR' or 'JJS') + (W_1 = 'CC') + (W_2 = 'JJ' or 'JJR' or 'JJS')	• W_0 • W_2
22	(W_0 = 'JJ' or 'JJR' or 'JJS') + (W_1 = 'JJ' or 'JJR' or 'JJS') + (W_2 = 'NN' or 'NNS' or 'NNP')	• $W_0 + W_1 + W_2$

ตารางที่ 16 กฎทางภาษาศาสตร์ที่ไม่พิจารณาร่วมกับคำศัพท์เฉพาะสำหรับระบบ CSCDA (ต่อ)

ลำดับ	รูปแบบหัวข้อ	ผลลัพธ์
23	(W_0 = 'JJ' or 'JJR' or 'JJS') + (W_1 = 'NN' or 'NNS' or 'NNP') + (W_2 = 'CC') + (W_3 = 'NN' or 'NNS' or 'NNP') + (W_4 = 'NN' or 'NNS' or 'NNP')	• $W_0 + W_1$ • $W_0 + W_3 + W_4$

ตารางที่ 17 กฎทางภาษาศาสตร์ที่พิจารณาร่วมกับคำศัพท์เฉพาะสำหรับระบบ CSCDA

ลำดับ	รูปแบบหัวข้อ	ผลลัพธ์
1	(W_0, ..., W_n = 'RW')	• W_0, ..., W_n
2	(W_0 = 'RW') + (W_1 = 'NN' or 'NNS' or 'NNP')	• $W_0 + W_1$
3	(W_0 = 'RW') + (W_1 = 'NN' or 'NNS' or 'NNP'), ..., (W_n = 'NN' or 'NNS' or 'NNP' or 'NNPS')	• $W_0 + W_1$, ..., W_n
4	(W_0 = 'RW') + (W_1 = 'RW') + (W_2 = 'NN' or 'NNS' or 'NNP')	• $W_0 + W_1 + W_2$
5	(W_0 = 'RW') + (W_1 = 'NN' or 'NNS' or 'NNP') + (W_2 = 'JJ' or 'JJR' or 'JJS') + (W_3 = 'NN' or 'NNS' or 'NNP')	• $W_0 + W_1$ • $W_2 + W_3$
6	(W_0 = 'JJ' or 'JJR' or 'JJS') + (W_1 = 'RW') + (W_2 = 'RW')	• $W_1 + W_2$ • $W_0 + W_1 + W_2$
7	(W_0 = 'JJ' or 'JJR' or 'JJS') + (W_1 = 'RW') + (W_2 = 'NN' or 'NNS' or 'NNP')	• $W_1 + W_2$ • $W_0 + W_1 + W_2$
8	(W_0 = 'JJ' or 'JJR' or 'JJS') + (W_1 = 'JJ' or 'JJR' or 'JJS') + (W_2 = 'RW') + (W_3 = 'NN' or 'NNS' or 'NNP')	• $W_2 + W_3$ • $W_0 + W_1 + W_2 + W_3$

ตารางที่ 17 กฎทางภาษาศาสตร์ที่พิจารณาร่วมกับคำศัพท์เฉพาะสำหรับระบบ CSCDA (ต่อ)

ลำดับ	รูปแบบหัวข้อ	ผลลัพธ์
9	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'RW')	• $W_0 + W_1$
10	(W_0 = 'JJ' or 'JJR' or 'JJS') + (W_1 = 'RW')	• $W_0 + W_1$
11	(W_0 = 'RW') + (W_1 = 'IN') + (W_2 = 'RW')	• $W_0 + W_1 + W_2$
12	(W_0 = 'JJ' or 'JJR' or 'JJS') + (W_1 = 'NN' or 'NNS' or 'NNP') + (W_2 = 'IN') + (W_3 = 'RW')	• W_3 • $W_0 + W_3$ • $W_3 + W_1$
13	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'IN') + (W_2 = 'RW') + (W_3 = 'IN') + (W_4 = 'RW')	• $W_2 + W_0$ • $W_2 + W_0 + W_1 + W_4$
14	(W_0 = 'JJ' or 'JJR' or 'JJS') + (W_1 = 'NN' or 'NNS' or 'NNP') + (W_2 = 'IN') + (W_3 = 'RW') + (W_4 = 'RW')	• $W_3 + W_4$ • $W_0 + W_3 + W_4$ • $W_3 + W_4 + W_1$
15	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'NN' or 'NNS' or 'NNP') + (W_2 = 'IN') + (W_3 = 'RW') + (W_4 = 'RW')	• $W_3 + W_4$ • $W_3 + W_4 + W_0$ • $W_3 + W_4 + W_1$
16	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'IN') + (W_2 = 'RW') + (W_3 = 'RW') + (W_4 = 'RW')	• $W_2 + W_3 + W_4$ • $W_2 + W_3 + W_4 + W_0$

ตารางที่ 17 กฎทางภาษาศาสตร์ที่พิจารณาร่วมกับคำศัพท์เฉพาะสำหรับระบบ CSCDA (ต่อ)

ลำดับ	รูปแบบหัวข้อ	ผลลัพธ์
17	(W_0 = 'RW') + (W_1 = 'NN' or 'NNS' or 'NNP') + (W_2 = 'IN') + (W_3 = 'RW')	• $W_0 + W_1$ • $W_0 + W_1 + W_2 + W_3$
18	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'IN') + (W_2 = 'RW')	• $W_2 + W_0$
19	(W_0 = 'JJ' or 'JJR' or 'JJS') + (W_1 = 'NN' or 'NNS' or 'NNP') + (W_2 = 'IN') + (W_3 = 'RW') + (W_4 = 'RW') + (W_5 = 'NN' or 'NNS' or 'NNP')	• $W_3 + W_4 + W_5$ • $W_3 + W_4 + W_5 + W_1$ • $W_0 + W_3 + W_4 + W_5$
20	(W_0 = 'RW') + (W_1 = 'NN' or 'NNS' or 'NNP') + (W_2 = 'CC') + (W_3 = 'RW')	• $W_0 + W_1$ • W_3
21	(W_0 = 'RW') + (W_1 = 'RW') + (W_2 = 'CC') + (W_3 = 'NN' or 'NNS' or 'NNP')	• $W_0 + W_1$ • $W_0 + W_3$
22	(W_0 = 'RW') + (W_1 = 'NN' or 'NNS' or 'NNP') + (W_2 = 'CC') + (W_3 = 'NN' or 'NNS' or 'NNP') + (W_4 = 'NN' or 'NNS' or 'NNP')	if $W_1 = W_4$: • $W_0 + W_1$ • $W_3 + W_4$
23	(W_0 = 'RW') + (W_1 = 'CC') + (W_2 = 'RW') + (W_3 = 'NN' or 'NNS' or 'NNP')	if $W_0 = W_2$: • W_0 • $W_2 + W_3$
24	(W_0 = 'RW') + (W_1 = 'CC') + (W_2 = 'RW') + (W_3 = 'RW')	• $W_0 + W_3$ • $W_2 + W_3$

ตารางที่ 17 กฎทางภาษาศาสตร์ที่พิจารณาร่วมกับคำศัพท์เฉพาะสำหรับระบบ CSCDA (ต่อ)

ลำดับ	รูปแบบหัวข้อ	ผลลัพธ์
25	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'CC') + (W_2 = 'RW')	• W_0 • W_2
26	(W_0 = 'RW') + (W_1 = 'CC') + (W_2 = 'RW') + (W_3 = 'NN' or 'NNS' or 'NNP')	• $W_0 + W_3$ • $W_2 + W_3$
27	(W_0 = 'RW') + (W_1 = 'CC') + (W_2 = 'RW')	• W_0 • W_2
28	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'RW') + (W_2 = 'CC') + (W_3 = 'RW')	• $W_0 + W_1$ • $W_0 + W_3$
29	(W_0 = 'JJ' or 'JJR' or 'JJS') + (W_1 = 'RW') + (W_2 = 'NN' or 'NNS' or 'NNP') + (W_3 = 'CC') + (W_4 = 'NN' or 'NNS' or 'NNP')	• $W_0 + W_1 + W_2$ • $W_0 + W_1 + W_4$
30	(W_0 = 'RW') + (W_1 = 'CC') + (W_2 = 'NN' or 'NNS' or 'NNP') + (W_3 = 'NN' or 'NNS' or 'NNP')	• W_0 • $W_2 + W_3$
31	(W_0 = 'JJ' or 'JJR' or 'JJS') + (W_1 = 'CC') + (W_2 = 'JJ' or 'JJR' or 'JJS') + (W_3 = 'RW')	• $W_0 + W_3$ • $W_2 + W_3$
32	(W_0 = 'RW') + (W_1 = 'NN' or 'NNS' or 'NNP') + (W_2 = 'CC') + (W_3 = 'RW') + (W_4 = 'NN' or 'NNS' or 'NNP')	• $W_0 + W_1$ • $W_3 + W_4$

ตารางที่ 17 กฎทางภาษาศาสตร์ที่พิจารณาร่วมกับคำศัพท์เฉพาะสำหรับระบบ CSCDA (ต่อ)

ลำดับ	รูปแบบหัวข้อ	ผลลัพธ์
33	(W_0 = 'RW') + (W_1 = 'NN' or 'NNS' or 'NNP') + (W_2 = 'CC') + (W_3 = 'NN' or 'NNS' or 'NNP')	• $W_0 + W_1$ • $W_0 + W_3$
34	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'IN') + (W_2 = 'RW') + (W_3 = 'CC') + (W_4 = 'NN' or 'NNS' or 'NNP')	• W_2 • $W_2 + W_0$ • $W_2 + W_4$
35	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'IN') + (W_2 = 'NN' or 'NNS' or 'NNP') + (W_3 = 'NN' or 'NNS' or 'NNP') + (W_4 = 'CC') + (W_5 = 'JJ' or 'JJR' or 'JJS') + (W_6 = 'RW') + (W_7 = 'NN' or 'NNS' or 'NNP')	• $W_5 + W_6 + W_7$ • $W_2 + W_3$ • $W_5 + W_6 + W_7 + W_0$ • $W_2 + W_3 + W_0$
36	(W_0 = 'RW') + (W_1 = 'IN') + (W_2 = 'RW') + (W_3 = 'NN' or 'NNS' or 'NNP') + (W_4 = 'CC') + (W_5 = 'NN' or 'NNS' or 'NNP') + (W_6 = 'NN' or 'NNS' or 'NNP')	• $W_2 + W_3$ • $W_5 + W_6$ • $W_2 + W_3 + W_0$ • $W_5 + W_6 + W_0$
37	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'IN') + (W_2 = 'NN' or 'NNS' or 'NNP') + (W_3 = 'CC') + (W_4 = 'RW')	• $W_2 + W_0$ • $W_4 + W_0$

ตารางที่ 17 กฎทางภาษาศาสตร์ที่พิจารณาร่วมกับคำศัพท์เฉพาะสำหรับระบบ CSCDA (ต่อ)

ลำดับ	รูปแบบหัวข้อ	ผลลัพธ์
38	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'NN' or 'NNS' or 'NNP') + (W_2 = 'CC') + (W_3 = 'NN' or 'NNS' or 'NNP') + (W_4 = 'IN') + (W_5 = 'RW') + (W_6 = 'NN' or 'NNS' or 'NNP')	• $W_0 + W_1 + W_4 + W_5 + W_6$ • $W_3 + W_1 + W_4 + W_5 + W_6$
39	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'CC') + (W_2 = 'NN' or 'NNS' or 'NNP') + (W_3 = 'IN') + (W_4 = 'RW')	• $W_0 + W_3 + W_4$ • $W_2 + W_3 + W_4$
40	(W_0 = 'RW') + (W_1 = 'NN' or 'NNS' or 'NNP') + (W_2 = 'CC') + (W_3 = 'JJ' or 'JJR' or 'JJS') + (W_4 = 'NN' or 'NNS' or 'NNP')	• $W_0 + W_1$ • $W_3 + W_4$
41	(W_0 = 'RW') + (W_1 = 'CC') + (W_2 = 'RW') + (W_3 = 'NN' or 'NNS' or 'NNP')	• W_0 • $W_2 + W_3$
42	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'CC') + (W_2 = 'RW') + (W_3 = 'RW')	• W_0 • $W_2 + W_3$
43	(W_0 = 'RW') + (W_1 = 'CC') + (W_2 = 'JJ' or 'JJR' or 'JJS') + (W_3 = 'NN' or 'NNS' or 'NNP')	• $W_2 + W_3 + W_0$

ตารางที่ 17 กฎทางภาษาศาสตร์ที่พิจารณาร่วมกับคำศัพท์เฉพาะสำหรับระบบ CSCDA (ต่อ)

ลำดับ	รูปแบบหัวข้อ	ผลลัพธ์
44	(W_0 = 'RW') + (W_1 = 'NN' or 'NNS' or 'NNP') + (W_2 = 'CC') + (W_3 = 'RW') + (W_4 = 'RW')	• $W_0 + W_1$ • $W_3 + W_4$
45	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'IN') + (W_2 = 'RW')	• $W_2 + W_0$
46	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'CC') + (W_2 = 'JJ' or 'JJR' or 'JJS') + (W_3 = 'RW')	• $W_0 + W_3$ • $W_2 + W_3$
47	(W_0 = 'RW') + (W_1 = 'IN') + (W_2 = 'NN' or 'NNS' or 'NNP') + (W_3 = 'NN' or 'NNS' or 'NNP') + (W_4 = 'IN') + (W_5 = 'JJ' or 'JJR' or 'JJS') + (W_6 = 'NN' or 'NNS' or 'NNP')	• $W_2 + W_3 + W_0$ • $W_2 + W_3 + W_0 + W_4 + W_5 + W_6$
48	(W_0 = 'JJ' or 'JJR' or 'JJS') + (W_1 = 'NN' or 'NNS' or 'NNP') + (W_2 = 'CC') + (W_3 = 'NN' or 'NNS' or 'NNP') + (W_4 = 'RW')	• $W_0 + W_1$ • $W_3 + W_4$
49	(W_0 = 'RW') + (W_1 = 'CC') + (W_2 = 'JJ' or 'JJR' or 'JJS') + (W_3 = 'RW') + (W_4 = 'RW')	• $W_2 + W_3 + W_4$ • $W_0 + W_3 + W_4$

ตารางที่ 17 กฎทางภาษาศาสตร์ที่พิจารณาร่วมกับคำศัพท์เฉพาะสำหรับระบบ CSCDA (ต่อ)

ลำดับ	รูปแบบหัวข้อ	ผลลัพธ์
50	(W_0 = 'RW') + (W_1 = 'NN' or 'NNS' or 'NNP') + (W_2 = 'IN') + (W_3 = 'NN' or 'NNS' or 'NNP') + (W_4 = 'NN' or 'NNS' or 'NNP')	• $W_0 + W_1$ • $W_3 + W_4$ • $W_0 + W_1 + W_2 + W_3 + W_4$
51	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'IN') + (W_2 = 'RW') + (W_3 = 'RW')	• $W_2 + W_3 + W_0$
52	(W_0 = 'RW') + (W_1 = 'RW') + (W_2 = 'CC') + (W_3 = 'JJ' or 'JJR' or 'JJS') + (W_4 = 'RW') + (W_5 = 'RW')	• $W_0 + W_1$ • $W_3 + W_4 + W_5$
53	(W_0 = 'JJ' or 'JJR' or 'JJS') + (W_1 = 'RW') + (W_2 = 'CC') + (W_3 = 'JJ' or 'JJR' or 'JJS') + (W_4 = 'NN' or 'NNS' or 'NNP')	• $W_0 + W_1$ • $W_3 + W_4$
54	(W_0 = 'JJ' or 'JJR' or 'JJS') + (W_1 = 'RW') + (W_2 = 'RW') + (W_3 = 'CC') + (W_4 = 'RW') + (W_5 = 'NN' or 'NNS' or 'NNP')	• $W_0 + W_1 + W_2$ • $W_0 + W_4 + W_5$
55	(W_0 = 'JJ' or 'JJR' or 'JJS') + (W_1 = 'CC') + (W_2 = 'JJ' or 'JJR' or 'JJS') + (W_3 = 'RW') + (W_4 = 'NN' or 'NNS' or 'NNP')	• $W_2 + W_3 + W_4$ • $W_0 + W_3 + W_4$

ตารางที่ 17 กฎทางภาษาศาสตร์ที่พิจารณาร่วมกับคำศัพท์เฉพาะสำหรับระบบ CSCDA (ต่อ)

ลำดับ	รูปแบบหัวข้อ	ผลลัพธ์
56	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'IN') + (W_2 = 'RW') + (W_3 = 'NN' or 'NNS' or 'NNP')	• $W_2 + W_3 + W_0$
57	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'CC') + (W_2 = 'JJ' or 'JJR' or 'JJS') + (W_3 = 'NN' or 'NNS' or 'NNP') + (W_4 = 'IN') + (W_5 = 'RW')	• $W_0 + W_4 + W_5$ • $W_2 + W_3 + W_4 + W_5$
58	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'IN') + (W_2 = 'RW') + (W_3 = 'IN') + (W_4 = 'RW') + (W_5 = 'NN' or 'NNS' or 'NNP')	• $W_2 + W_0$ • $W_2 + W_0 + W_3 + W_4 + W_5$
59	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'IN') + (W_2 = 'RW') + (W_3 = 'RW') + (W_4 = 'CC') + (W_5 = 'RW') + (W_6 = 'RW')	• $W_2 + W_3 + W_0$ • $W_5 + W_6 + W_0$
60	(W_0 = 'JJ' or 'JJR' or 'JJS') + (W_1 = 'NN' or 'NNS' or 'NNP') + (W_2 = 'IN') + (W_3 = 'RW') + (W_4 = 'RW')	• $W_0 + W_1$ • $W_3 + W_4$
61	(W_0 = 'RW') + (W_1 = 'RW') + (W_2 = 'NN' or 'NNS' or 'NNP') + (W_3 = 'CC') + (W_4 = 'NN' or 'NNS' or 'NNP')	• $W_0 + W_1 + W_2$ • $W_0 + W_1 + W_4$

ตารางที่ 17 กฎทางภาษาศาสตร์ที่พิจารณาร่วมกับคำศัพท์เฉพาะสำหรับระบบ CSCDA (ต่อ)

ลำดับ	รูปแบบหัวข้อ	ผลลัพธ์
62	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'IN') + (W_2 = 'RW') + (W_3 = 'NN' or 'NNS' or 'NNP')	● $W_2 + W_3 + W_0$
63	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'IN') + (W_2 = 'RW')	● $W_2 + W_0$
64	(W_0 = 'JJ' or 'JJR' or 'JJS') + (W_1 = 'RW') + (W_2 = 'RW')	● $W_0 + W_1 + W_2$
65	(W_0 = 'RW') + (W_1 = 'RW') + (W_2 = 'NN' or 'NNS' or 'NNP') + (W_3 = 'NN' or 'NNS' or 'NNP')	● $W_0 + W_1 + W_2 + W_3$
66	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'IN') + (W_2 = 'RW') + (W_3 = 'NN' or 'NNS' or 'NNP') + (W_4 = 'NN' or 'NNS' or 'NNP')	● $W_2 + W_3 + W_4 + W_0$
67	(W_0 = 'RW') + (W_1 = 'RW') + (W_2 = 'NN' or 'NNS' or 'NNP')	● $W_0 + W_1 + W_2$
68	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'RW') + (W_2 = 'NN' or 'NNS' or 'NNP')	● $W_0 + W_1 + W_2$
69	(W_0 = 'RW') + (W_1 = 'IN') + (W_2 = 'RW')	● $W_0 + W_1 + W_2$

ตารางที่ 17 กฎทางภาษาศาสตร์ที่พิจารณาร่วมกับคำศัพท์เฉพาะสำหรับระบบ CSCDA (ต่อ)

ลำดับ	รูปแบบหัวข้อ	ผลลัพธ์
70	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'IN') + (W_2 = 'JJ' or 'JJR' or 'JJS') + (W_3 = 'RW') + (W_4 = 'CC') + (W_5 = 'NN' or 'NNS' or 'NNP')	• $W_2 + W_3 + W_0$ • $W_5 + W_0$
71	(W_0 = 'RW') + (W_1 = 'CC') + (W_2 = 'NN' or 'NNS' or 'NNP') + (W_3 = 'IN') + (W_4 = 'RW')	• $W_0 + W_3 + W_4$ • $W_2 + W_3 + W_4$
72	(W_0 = 'RW') + (W_1 = 'IN') + (W_2 = 'RW') + (W_3 = 'RW')	• $W_0 + W_1 + W_2 + W_3$
73	(W_0 = 'JJ' or 'JJR' or 'JJS') + (W_1 = 'RW') + (W_2 = 'CC') + (W_3 = 'JJ' or 'JJR' or 'JJS') + (W_4 = 'RW') + (W_5 = 'IN') + (W_6 = 'RW')	• $W_0 + W_1 + W_5 + W_6$ • $W_3 + W_4 + W_5 + W_6$
74	(W_0 = 'RW') + (W_1 = 'IN') + (W_2 = 'JJ' or 'JJR' or 'JJS') + (W_3 = 'RW')	• $W_0 + W_1 + W_2 + W_3$
75	(W_0 = 'RW') + (W_1 = 'CC') + (W_2 = 'JJ' or 'JJR' or 'JJS') + (W_3 = 'NN' or 'NNS' or 'NNP')	• $W_0 + W_3$ • $W_2 + W_3$

ตารางที่ 17 กฎทางภาษาศาสตร์ที่พิจารณาร่วมกับคำศัพท์เฉพาะสำหรับระบบ CSCDA (ต่อ)

ลำดับ	รูปแบบหัวข้อ	ผลลัพธ์
76	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'RW') + (W_2 = 'CC') + (W_3 = 'RW')	• $W_0 + W_1$ • $W_0 + W_3$
77	(W_0 = 'RW') + (W_1 = 'CC') + (W_2 = 'JJ' or 'JJR' or 'JJS') + (W_3 = 'NN' or 'NNS' or 'NNP') + (W_4 = 'NN' or 'NNS' or 'NNP')	• $W_2 + W_3 + W_4$ • $W_0 + W_3 + W_4$
78	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'IN') + (W_2 = 'RW') + (W_3 = 'CC') + (W_4 = 'JJ' or 'JJR' or 'JJS') + (W_5 = 'RW') + (W_6 = 'NN' or 'NNS' or 'NNP')	• $W_2 + W_0$ • $W_4 + W_5 + W_6 + W_0$
79	(W_0 = 'RW') + (W_1 = 'CC') + (W_2 = 'RW')	• W_0 • W_2
80	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'CC') + (W_2 = 'RW')	• W_0 • W_2
81	(W_0 = 'RW') + (W_1 = 'IN') + (W_2 = 'RW') + (W_3 = 'RW')	• $W_0 + W_1 + W_2 + W_3$
82	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'CC') + (W_2 = 'RW') + (W_3 = 'RW') + (W_4 = 'NN' or 'NNS' or 'NNP')	if $W_0 = W_2$: • W_0 • $W_2 + W_3 + W_4$

ตารางที่ 17 กฎทางภาษาศาสตร์ที่พิจารณาร่วมกับคำศัพท์เฉพาะสำหรับระบบ CSCDA (ต่อ)

ลำดับ	รูปแบบหัวข้อ	ผลลัพธ์
83	(W_0 = 'RW') + (W_1 = 'RW') + (W_2 = 'RW') + (W_3 = 'NN' or 'NNS' or 'NNP') + (W_4 = 'CC') + (W_5 = 'NN' or 'NNS' or 'NNP')	• $W_0 + W_1 + W_2 + W_3$ • $W_0 + W_1 + W_2 + W_5$
84	(W_0 = 'RW') + (W_1 = 'NN' or 'NNS' or 'NNP') + (W_2 = 'IN') + (W_3 = 'RW') + (W_4 = 'RW')	• $W_0 + W_1 + W_2 + W_3 + W_4$
85	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'IN') + (W_2 = 'NN' or 'NNS' or 'NNP') + (W_3 = 'CC') + (W_4 = 'RW')	• $W_2 + W_0$ • $W_4 + W_0$
86	(W_0 = 'JJ' or 'JJR' or 'JJS') + (W_1 = 'NN' or 'NNS' or 'NNP') + (W_2 = 'IN') + (W_3 = 'RW')	• $W_3 + W_1$ • $W_0 + W_3$
87	(W_0 = 'JJ' or 'JJR' or 'JJS') + (W_1 = 'RW') + (W_2 = 'NN' or 'NNS' or 'NNP')	• $W_0 + W_1 + W_2$
88	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'IN') + (W_2 = 'RW') + (W_3 = 'CC') + (W_4 = 'JJ' or 'JJR' or 'JJS') + (W_5 = 'RW') + (W_6 = 'NN' or 'NNS' or 'NNP')	• $W_2 + W_0$ • $W_4 + W_5 + W_6 + W_0$

ตารางที่ 17 กฎทางภาษาศาสตร์ที่พิจารณาร่วมกับคำศัพท์เฉพาะสำหรับระบบ CSCDA (ต่อ)

ลำดับ	รูปแบบหัวข้อ	ผลลัพธ์
89	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'CC') + (W_2 = 'NN' or 'NNS' or 'NNP') + (W_3 = 'IN') + (W_4 = 'RW')	• $W_0 + W_3 + W_4$ • $W_2 + W_3 + W_4$
90	(W_0 = 'RW') + (W_1 = 'NN' or 'NNS' or 'NNP')	• $W_0 + W_1$
91	(W_0 = 'JJ' or 'JJR' or 'JJS') + (W_1 = 'RW') + (W_2 = 'IN') + (W_3 = 'RW')	• $W_0 + W_1 + W_2 + W_3$
92	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'IN') + (W_2 = 'RW') + (W_3 = 'NN' or 'NNS' or 'NNP')	• $W_2 + W_3 + W_0$
93	(W_0 = 'RW') + (W_1 = 'NN' or 'NNS' or 'NNP') + (W_2 = 'IN') + (W_3 = 'JJ' or 'JJR' or 'JJS') + (W_4 = 'RW')	• $W_0 + W_1 + W_2 + W_3 + W_4$
94	(W_0 = 'RW') + (W_1 = 'IN') + (W_2 = 'JJ' or 'JJR' or 'JJS') + (W_3 = 'JJ' or 'JJR' or 'JJS') + (W_4 = 'RW')	• $W_0 + W_1 + W_2 + W_3 + W_4$
95	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'NN' or 'NNS' or 'NNP') + (W_2 = 'CC') + (W_3 = 'RW') + (W_4 = 'RW')	• $W_0 + W_1$ • $W_3 + W_4$

ตารางที่ 17 กฎทางภาษาศาสตร์ที่พิจารณาร่วมกับคำศัพท์เฉพาะสำหรับระบบ CSCDA (ต่อ)

ลำดับ	รูปแบบหัวข้อ	ผลลัพธ์
96	(W_0 = 'RW') + (W_1 = 'NN' or 'NNS' or 'NNP') + (W_2 = 'CC') + (W_3 = 'NN' or 'NNS' or 'NNP')	• $W_0 + W_1$ • $W_0 + W_3$
97	(W_0 = 'RW') + (W_1 = 'RW') + (W_2 = 'NN' or 'NNS' or 'NNP')	• $W_0 + W_1 + W_2$
98	(W_0 = 'RW') + (W_1 = 'NN' or 'NNS' or 'NNP') + (W_2 = 'NN' or 'NNS' or 'NNP') + (W_3 = 'CC') + (W_4 = 'NN' or 'NNS' or 'NNP')	• $W_0 + W_1 + W_2$ • $W_0 + W_1 + W_4$
99	(W_0 = 'RW') + (W_1 = 'NN' or 'NNS' or 'NNP') + (W_2 = 'CC') + (W_3 = 'RW')	• $W_0 + W_1$ • $W_0 + W_3$
100	(W_0 = 'RW') + (W_1 = 'NN' or 'NNS' or 'NNP') + (W_2 = 'CC') + (W_3 = 'RW') + (W_4 = 'RW')	• $W_0 + W_1$ • $W_0 + W_3 + W_4$
101	(W_0 = 'RW') + (W_1 = 'IN') + (W_2 = 'RW') + (W_3 = 'RW') + (W_4 = 'NN' or 'NNS' or 'NNP')	• $W_0 + W_1 + W_2 + W_3 + W_4$

ตารางที่ 17 กฎทางภาษาศาสตร์ที่พิจารณาร่วมกับคำศัพท์เฉพาะสำหรับระบบ CSCDA (ต่อ)

ลำดับ	รูปแบบหัวข้อ	ผลลัพธ์
102	(W_0 = 'RW') + (W_1 = 'NN' or 'NNS' or 'NNP') + (W_2 = 'CC') + (W_3 = 'NN' or 'NNS' or 'NNP') + (W_4 = 'IN') + (W_5 = 'RW') + (W_6 = 'NN' or 'NNS' or 'NNP')	• $W_0 + W_1 + W_4 + W_5 + W_6$ • $W_3 + W_4 + W_5 + W_6$
103	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'CC') + (W_2 = 'NN' or 'NNS' or 'NNP') + (W_3 = 'RW')	• $W_0 + W_3$ • $W_2 + W_3$
104	(W_0 = 'JJ' or 'JJR' or 'JJS') + (W_1 = 'NN' or 'NNS' or 'NNP') + (W_2 = 'CC') + (W_3 = 'RW')	• $W_0 + W_1$ • $W_0 + W_3$
105	(W_0 = 'RW') + (W_1 = 'IN') + (W_2 = 'RW') + (W_3 = 'NN' or 'NNS' or 'NNP')	• $W_0 + W_1 + W_2 + W_3$
106	(W_0 = 'RW') + (W_1 = 'CC') + (W_2 = 'RW') + (W_3 = 'RW')	• W_0 • $W_2 + W_3$
107	(W_0 = 'RW') + (W_1 = 'CC') + (W_2 = 'NN' or 'NNS' or 'NNP')	• W_0 • W_2
108	(W_0 = 'RW') + (W_1 = 'NN' or 'NNS' or 'NNP') + (W_2 = 'CC') + (W_3 = 'RW') + (W_4 = 'NN' or 'NNS' or 'NNP')	• $W_0 + W_1$ • $W_3 + W_4$

ตารางที่ 17 กฎทางภาษาศาสตร์ที่พิจารณาร่วมกับคำศัพท์เฉพาะสำหรับระบบ CSCDA (ต่อ)

ลำดับ	รูปแบบหัวข้อ	ผลลัพธ์
109	(W_0 = 'RW') + (W_1 = 'JJ' or 'JJR' or 'JJS') + (W_2 = 'NN' or 'NNS' or 'NNP') + (W_3 = 'RW') + (W_4 = 'IN') + (W_5 = 'JJ' or 'JJR' or 'JJS')	• $W_1 + W_2 + W_3$ • $W_0 + W_1 + W_2 + W_3 + W_4 + W_5$
110	(W_0 = 'RW') + (W_1 = 'JJ' or 'JJR' or 'JJS') + (W_2 = 'NN' or 'NNS' or 'NNP') + (W_3 = 'RW') + (W_4 = 'RW') + (W_5 = 'IN') + (W_6 = 'RW')	• $W_1 + W_2 + W_3 + W_4$ • $W_0 + W_1 + W_2 + W_3 + W_4 + W_5 + W_6$
111	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'IN') + (W_2 = 'RW') + (W_3 = 'NN' or 'NNS' or 'NNP') + (W_4 = 'CC') + (W_5 = 'RW') + (W_6 = 'NN' or 'NNS' or 'NNP')	• $W_2 + W_3 + W_0$ • $W_5 + W_6 + W_0$
112	(W_0 = 'RW') + (W_1 = 'CC') + (W_2 = 'NN' or 'NNS' or 'NNP') + (W_3 = 'NN' or 'NNS' or 'NNP')	• W_0 • $W_2 + W_3$
113	(W_0 = 'RW') + (W_1 = 'RW') + (W_2 = 'NN' or 'NNS' or 'NNP')	• $W_0 + W_1 + W_2$
114	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'CC') + (W_2 = 'RW')	• W_0 • W_2

ตารางที่ 17 กฎทางภาษาศาสตร์ที่พิจารณาร่วมกับคำศัพท์เฉพาะสำหรับระบบ CSCDA (ต่อ)

ลำดับ	รูปแบบหัวข้อ	ผลลัพธ์
115	(W_0 = 'JJ' or 'JJR' or 'JJS') + (W_1 = 'RW') + (W_2 = 'NN' or 'NNS' or 'NNP') + (W_3 = 'CC') + (W_4 = 'RW') + (W_5 = 'NN' or 'NNS' or 'NNP')	• $W_0 + W_1 + W_2$ • $+ W_4 + W_5$
116	(W_0 = 'JJ' or 'JJR' or 'JJS') + (W_1 = 'RW') + (W_2 = 'NN' or 'NNS' or 'NNP') + (W_3 = 'NN' or 'NNS' or 'NNP')	• $W_0 + W_1 + W_2 + W_3$
117	(W_0 = 'JJ' or 'JJR' or 'JJS') + (W_1 = 'NN' or 'NNS' or 'NNP') + (W_2 = 'CC') + (W_3 = 'NN' or 'NNS' or 'NNP') + (W_4 = 'IN') + (W_5 = 'RW')	• $W_5 + W_3$ • $W_5 + W_1$ • $W_0 + W_5$
118	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'CC') + (W_2 = 'RW') + (W_3 = 'RW') + (W_4 = 'NN' or 'NNS' or 'NNP')	• $W_0 + W_4$ • $W_2 + W_3 + W_4$
119	(W_0 = 'JJ' or 'JJR' or 'JJS') + (W_1 = 'RW') + (W_2 = 'NN' or 'NNS' or 'NNP') + (W_3 = 'CC') + (W_4 = 'NN' or 'NNS' or 'NNP')	• $W_0 + W_1 + W_2$ • $W_0 + W_1 + W_4$
120	(W_0 = 'RW') + (W_1 = 'RW') + (W_2 = 'NN' or 'NNS' or 'NNP') + (W_3 = 'NN' or 'NNS' or 'NNP')	• $W_0 + W_1 + W_2 + W_3$

ตารางที่ 17 กฎทางภาษาศาสตร์ที่พิจารณาร่วมกับคำศัพท์เฉพาะสำหรับระบบ CSCDA (ต่อ)

ลำดับ	รูปแบบหัวข้อ	ผลลัพธ์
121	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'IN') + (W_2 = 'RW') + (W_3 = 'CC') + (W_4 = 'NN' or 'NNS' or 'NNP')	• W_2 • $W_2 + W_0$ • $W_2 + W_4$
122	(W_0 = 'RW') + (W_1 = 'NN' or 'NNS' or 'NNP') + (W_2 = 'JJ' or 'JJR' or 'JJS') + (W_3 = 'NN' or 'NNS' or 'NNP')	• $W_0 + W_1$ • $W_2 + W_3$
123	(W_0 = 'JJ' or 'JJR' or 'JJS') + (W_1 = 'NN' or 'NNS' or 'NNP') + (W_2 = 'IN') + (W_3 = 'RW') + (W_4 = 'RW')	• $W_3 + W_4$ • $W_0 + W_3 + W_4$ • $W_3 + W_4 + W_1$
124	(W_0 = 'RW') + (W_1 = 'IN') + (W_2 = 'RW')	• $W_0 + W_1 + W_2$
125	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'IN') + (W_2 = 'RW') + (W_3 = 'IN') + (W_4 = 'RW')	• $W_2 + W_0$ • $W_2 + W_0 + W_3 + W_4$
126	(W_0 = 'JJ' or 'JJR' or 'JJS') + (W_1 = 'RW') + (W_2 = 'NN' or 'NNS' or 'NNP') + (W_3 = ''CC) + (W_4 = 'NN' or 'NNS' or 'NNP') + (W_5 = 'NN' or 'NNS' or 'NNP')	if $W_2 = W_5$: • $W_0 + W_1 + W_2$ • $W_4 + W_5$
127	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'NN' or 'NNS' or 'NNP') + (W_2 = 'IN') + (W_3 = 'RW')	• $W_0 + W_1$ • W_3

ตารางที่ 17 กฎทางภาษาศาสตร์ที่พิจารณาร่วมกับคำศัพท์เฉพาะสำหรับระบบ CSCDA (ต่อ)

ลำดับ	รูปแบบหัวข้อ	ผลลัพธ์
128	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'IN') + (W_2 = 'RW') + (W_3 = 'RW') + (W_4 = 'RW')	• $W_2 + W_3 + W_4$ • $W_2 + W_3 + W_4 + W_0$
129	(W_0 = 'JJ' or 'JJR' or 'JJS') + (W_1 = 'NN' or 'NNS' or 'NNP') + (W_2 = 'RW')	• $W_0 + W_1 + W_2$
130	(W_0 = 'JJ' or 'JJR' or 'JJS') + (W_1 = 'JJ' or 'JJR' or 'JJS') + (W_2 = 'NN' or 'NNS' or 'NNP') + (W_3 = 'RW')	• $W_0 + W_1 + W_2 + W_3$
131	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'RW') + (W_2 = 'CC') + (W_3 = 'RW') + (W_4 = 'NN' or 'NNS' or 'NNP')	• $W_0 + W_1$ • $W_3 + W_4$
132	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'IN') + (W_2 = 'NN' or 'NNS' or 'NNP') + (W_3 = 'NN' or 'NNS' or 'NNP') + (W_4 = 'CC') + (W_5 = 'JJ' or 'JJR' or 'JJS') + (W_6 = 'RW') + (W_7 = 'NN' or 'NNS' or 'NNP')	• $W_5 + W_6 + W_7$ • $W_2 + W_3$ • $W_5 + W_6 + W_7 + W_0$ • $W_2 + W_3 + W_0$
133	(W_0 = 'RW') + (W_1 = 'IN') + (W_2 = 'RW') + (W_3 = 'NN' or 'NNS' or 'NNP') + (W_4 = 'CC') + (W_5 = 'RW') + (W_6 = 'NN' or 'NNS' or 'NNP')	• $W_2 + W_3$ • $W_5 + W_6$ • $W_2 + W_3 + W_0$ • $W_5 + W_6 + W_0$

ตารางที่ 17 กฎทางภาษาศาสตร์ที่พิจารณาร่วมกับคำศัพท์เฉพาะสำหรับระบบ CSCDA (ต่อ)

ลำดับ	รูปแบบหัวข้อ	ผลลัพธ์
134	(W_0 = 'RW') + (W_1 = 'RW') + (W_2 = 'CC') + (W_3 = 'RW')	• $W_0 + W_1$ • $W_0 + W_3$
135	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'NN' or 'NNS' or 'NNP') + (W_2 = 'RW')	• $W_0 + W_1 + W_2$
136	(W_0 = 'JJ' or 'JJR' or 'JJS') + (W_1 = 'RW') + (W_2 = 'NN' or 'NNS' or 'NNP') + (W_3 = ''CC) + (W_4 = 'NN' or 'NNS' or 'NNP')	if $W_2 = W_4$: • $W_0 + W_1 + W_2$ • $W_0 + W_1 + W_4$
137	(W_0 = 'RW') + (W_1 = 'NN' or 'NNS' or 'NNP') + (W_2 = 'IN') + (W_3 = 'JJ' or 'JJR' or 'JJS') + (W_4 = 'RW') + (W_5 = 'NN' or 'NNS' or 'NNP')	• $W_0 + W_1$ • $W_3 + W_4 + W_5$
138	(W_0 = 'JJ' or 'JJR' or 'JJS') + (W_1 = 'NN' or 'NNS' or 'NNP') + (W_2 = 'IN') + (W_3 = 'JJ' or 'JJR' or 'JJS') + (W_4 = 'NN' or 'NNS' or 'NNP') + (W_5 = 'CC') + (W_6 = 'RW')	• $W_0 + W_1$ • $W_3 + W_4$ • $W_0 + W_1 + W_6$ • $W_3 + W_4 + W_6$
139	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'IN') + (W_2 = 'JJ' or 'JJR' or 'JJS') + (W_3 = 'NN' or 'NNS' or 'NNP') + (W_4 = 'CC') + (W_5 = 'RW')	• $W_2 + W_3 + W_0$ • $W_4 + W_0$

ตารางที่ 17 กฎทางภาษาศาสตร์ที่พิจารณาร่วมกับคำศัพท์เฉพาะสำหรับระบบ CSCDA (ต่อ)

ลำดับ	รูปแบบหัวข้อ	ผลลัพธ์
140	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'NN' or 'NNS' or 'NNP') + (W_2 = 'CC') + (W_3 = 'JJ' or 'JJR' or 'JJS') + (W_4 = 'NN' or 'NNS' or 'NNP') + (W_5 = 'IN') + (W_6 = 'RW') + (W_7 = 'NN' or 'NNS' or 'NNP')	• $W_0 + W_1 + W_5 + W_6 + W_7$ • $W_3 + W_4 + W_5 + W_6 + W_7$
141	(W_0 = 'RW') + (W_1 = 'CC') + (W_2 = 'NN' or 'NNS' or 'NNP') + (W_3 = 'NN' or 'NNS' or 'NNP') + (W_4 = 'RW')	• $W_0 + W_4$ • $W_2 + W_3 + W_4$
142	(W_0 = 'JJ' or 'JJR' or 'JJS') + (W_1 = 'CC') + (W_2 = 'JJ' or 'JJR' or 'JJS') + (W_3 = 'NN' or 'NNS' or 'NNP') + (W_4 = 'RW')	• $W_0 + W_3 + W_4$ • $W_2 + W_3 + W_4$
143	(W_0 = 'RW') + (W_1 = 'RW') + (W_2 = 'NN' or 'NNS' or 'NNP') + (W_3 = 'CC') + (W_4 = 'RW') + (W_5 = 'NN' or 'NNS' or 'NNP') + (W_6 = 'NN' or 'NNS' or 'NNP')	• $W_0 + W_1 + W_2$ • $W_4 + W_5 + W_6$
144	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'CC') + (W_2 = 'NN' or 'NNS' or 'NNP') + (W_3 = 'IN') + (W_4 = 'JJ' or 'JJR' or 'JJS') + (W_5 = 'RW')	• $W_0 + W_3 + W_4 + W_5$ • $W_2 + W_3 + W_4 + W_5$

ตารางที่ 17 กฎทางภาษาศาสตร์ที่พิจารณาร่วมกับคำศัพท์เฉพาะสำหรับระบบ CSCDA (ต่อ)

ลำดับ	รูปแบบหัวข้อ	ผลลัพธ์
145	(W_0 = 'JJ' or 'JJR' or 'JJS') + (W_1 = 'RW')	• $W_0 + W_1$
146	(W_0 = 'RW') + (W_1 = 'RW') + (W_2 = 'NN' or 'NNS' or 'NNP') + (W_3 = 'CC') + (W_4 = 'NN' or 'NNS' or 'NNP') + (W_5 = 'NN' or 'NNS' or 'NNP')	• $W_0 + W_1 + W_2$ • $W_4 + W_5$
147	(W_0 = 'JJ' or 'JJR' or 'JJS') + (W_1 = 'RW') + (W_2 = 'RW') + (W_3 = 'CC') + (W_4 = 'RW')	• $W_0 + W_1 + W_2$ • $W_0 + W_1 + W_4$
148	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'IN') + (W_2 = 'NN' or 'NNS' or 'NNP') + (W_3 = 'RW')	• $W_2 + W_3 + W_0$
149	(W_0 = 'RW') + (W_1 = 'RW') + (W_2 = 'CC') + (W_3 = 'RW') + (W_4 = 'RW')	• $W_0 + W_1$ • $W_3 + W_4$
150	(W_0 = 'RW') + (W_1 = 'RW') + (W_2 = 'RW')	• $W_0 + W_1 + W_2$
151	(W_0 = 'JJ' or 'JJR' or 'JJS') + (W_1 = 'IN') + (W_2 = 'RW') + (W_3 = 'RW')	• $W_0 + W_2 + W_3$
152	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'IN') + (W_2 = 'JJ' or 'JJR' or 'JJS') + (W_3 = 'RW')	• $W_2 + W_3 + W_0$

ตารางที่ 17 กฎทางภาษาศาสตร์ที่พิจารณาร่วมกับคำศัพท์เฉพาะสำหรับระบบ CSCDA (ต่อ)

ลำดับ	รูปแบบหัวข้อ	ผลลัพธ์
153	(W_0 = 'JJ' or 'JJR' or 'JJS') + (W_1 = 'JJ' or 'JJR' or 'JJS') + (W_2 = 'RW')	• $W_0 + W_1 + W_2$
154	(W_0 = 'JJ' or 'JJR' or 'JJS') + (W_1 = 'RW') + (W_2 = 'RW') + (W_3 = 'NN' or 'NNS' or 'NNP')	• $W_0 + W_1 + W_2 + W_3$
155	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'IN') + (W_2 = 'RW')	• $W_2 + W_0$
156	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'NN' or 'NNS' or 'NNP') + (W_2 = 'RW') + (W_3 = 'RW')	• $W_2 + W_3$ • $W_0 + W_1 + W_2 + W_3$
157	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'IN') + (W_2 = 'RW') + (W_3 = 'RW') + (W_4 = 'NN' or 'NNS' or 'NNP')	• $W_2 + W_3 + W_4 + W_0$
158	(W_0 = 'JJ' or 'JJR' or 'JJS') + (W_1 = 'CC') + (W_2 = 'JJ' or 'JJR' or 'JJS') + (W_3 = 'RW') + (W_4 = 'RW')	• $W_0 + W_3 + W_4$ • $W_2 + W_3 + W_4$
159	(W_0 = 'JJ' or 'JJR' or 'JJS') + (W_1 = 'RW') + (W_2 = 'NN' or 'NNS' or 'NNP') + (W_3 = 'RW')	• $W_0 + W_1$ • $W_2 + W_3$

ตารางที่ 17 กฎทางภาษาศาสตร์ที่พิจารณาร่วมกับคำศัพท์เฉพาะสำหรับระบบ CSCDA (ต่อ)

ลำดับ	รูปแบบหัวข้อ	ผลลัพธ์
160	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'IN') + (W_2 = 'JJ' or 'JJR' or 'JJS') + (W_3 = 'RW') + (W_4 = 'RW')	• $W_2 + W_3 + W_4 + W_0$
161	(W_0 = 'JJ' or 'JJR' or 'JJS') + (W_1 = 'IN') + (W_2 = 'RW')	• $W_0 + W_2$
162	(W_0 = 'RW') + (W_1 = 'CC') + (W_2 = 'JJ' or 'JJR' or 'JJS') + (W_3 = 'RW')	• W_0 • $W_2 + W_3$
163	(W_0 = 'RW') + (W_1 = 'CC') + (W_2 = 'RW') + (W_3 = 'NN' or 'NNS' or 'NNP') + (W_4 = 'NN' or 'NNS' or 'NNP')	• $W_0 + W_3 + W_4$ • $W_2 + W_3 + W_4$
164	(W_0 = 'JJ' or 'JJR' or 'JJS') + (W_1 = 'NN' or 'NNS' or 'NNP') + (W_2 = 'IN') + (W_3 = 'RW') + (W_4 = 'NN' or 'NNS' or 'NNP')	• $W_3 + W_4 + W_1$ • $W_0 + W_3 + W_4$
165	(W_0 = 'JJ' or 'JJR' or 'JJS') + (W_1 = 'JJ' or 'JJR' or 'JJS') + (W_2 = 'RW') + (W_3 = 'RW') + (W_4 = 'NN' or 'NNS' or 'NNP')	• $W_2 + W_3 + W_4$ • $W_0 + W_1 + W_2 + W_3 + W_4$
166	(W_0 = 'JJ' or 'JJR' or 'JJS') + (W_1 = 'NN' or 'NNS' or 'NNP') + (W_2 = 'RW') + (W_3 = 'RW')	• $W_2 + W_3$ • $W_0 + W_1 + W_2 + W_3$

ตารางที่ 17 กฎทางภาษาศาสตร์ที่พิจารณาร่วมกับคำศัพท์เฉพาะสำหรับระบบ CSCDA (ต่อ)

ลำดับ	รูปแบบหัวข้อ	ผลลัพธ์
167	(W_0 = 'RW') + (W_1 = 'RW') + (W_2 = 'RW') + (W_3 = 'CC') + (W_4 = 'RW') + (W_5 = 'RW')	• $W_0 + W_1 + W_2$ • $W_4 + W_5$
168	(W_0 = 'RW') + (W_1 = 'RW') + (W_2 = 'NN' or 'NNS' or 'NNP') + (W_3 = 'CC') + (W_4 = 'NN' or 'NNS' or 'NNP')	• $W_0 + W_1 + W_2$ • $W_0 + W_1 + W_4$
169	(W_0 = 'RW') + (W_1 = 'RW') + (W_2 = 'RW') + (W_3 = 'CC') + (W_4 = 'RW') + (W_5 = 'NN' or 'NNS' or 'NNP')	• $W_0 + W_1 + W_2$ • $W_0 + W_4 + W_5$
170	(W_0 = 'RW') + (W_1 = 'NN' or 'NNS' or 'NNP') + (W_2 = 'CC') + (W_3 = 'JJ' or 'JJR' or 'JJS') + (W_4 = 'NN' or 'NNS' or 'NNP') + (W_5 = 'IN') + (W_6 = 'RW') + (W_7 = 'NN' or 'NNS' or 'NNP')	• $W_0 + W_1 + W_5 + W_6 + W_7$ • $W_3 + W_4 + W_5 + W_6 + W_7$
171	(W_0 = 'NN' or 'NNS' or 'NNP') + (W_1 = 'IN') + (W_2 = 'RW') + (W_3 = 'RW')	• $W_2 + W_3 + W_0$

ภาคผนวก ข

กฎทางภาษาศาสตร์สำหรับระบบ *eCSCDA*

ตารางที่ 18 กฎทางภาษาศาสตร์สำหรับระบบ eCSCDA

ลำดับ	รูปแบบหัวข้อ	ผลลัพธ์
1	$JJ_1, ..., JJ_p$	● $JJ_i, ..., JJ_k \subseteq JJ_1, ..., JJ_p$ When $JJ_u \in JJ_i, ..., JJ_k \cap JJ_u \notin$ Adjective corpus
2	$JJ_1, ..., JJ_p$ + $NN_1, ..., NN_q$	● $JJ_i, ..., JJ_k \subseteq JJ_1, ..., JJ_p + NN_1, ..., NN_q$ When $JJ_u \in JJ_i, ..., JJ_k \cap JJ_u \notin$ Adjective corpus
3	$JJ_1, ..., JJ_p$ + $TE_1, ..., TE_q$	● $JJ_i, ..., JJ_k \subseteq JJ_1, ..., JJ_p + TE_1, ..., TE_q$ When $JJ_u \in JJ_i, ..., JJ_k \cap JJ_u \notin$ Adjective corpus
4	$NN_1, ..., NN_p$	● $NN_1, ..., NN_p$
5	$NN_1, ..., NN_p$ + $TE_1, ..., TE_q$	● $NN_1, ..., NN_p + TE_1, ..., TE_q$
6	$TE_1, ..., TE_p$	● $TE_1, ..., TE_p$
7	$TE_1, ..., TE_p$ + $NN_1, ..., NN_q$	● $TE_1, ..., TE_p + NN_1, ..., NN_q$
8	$JJ_1, ..., JJ_p$ + $NN_1, ..., NN_q$ + $TE_1, ..., TE_r$	● $JJ_i, ..., JJ_k \subseteq JJ_1, ..., JJ_p + NN_1, ..., NN_q + TE_1, ..., TE_r$ When $JJ_u \in JJ_i, ..., JJ_k \cap JJ_u \notin$ Adjective corpus
9	$JJ_1, ..., JJ_p$ + $TE_1, ..., TE_q$ + $NN_1, ..., NN_r$	● $JJ_i, ..., JJ_k \subseteq JJ_1, ..., JJ_p + TE_1, ..., TE_q + NN_1, ..., NN_r$ When $JJ_u \in JJ_i, ..., JJ_k \cap JJ_u \notin$ Adjective corpus
10	$NN_1, ..., NN_p$ + $TE_1, ..., TE_q$ + $NN_1, ..., NN_r$	● $NN_1, ..., NN_p + TE_1, ..., TE_q + NN_1, ..., NN_r$
11	$TE_1, ..., TE_p$ + $NN_1, ..., NN_q$ + $TE_1, ..., TE_r$	● $TE_1, ..., TE_p + NN_1, ..., NN_q + TE_1, ..., TE_r$

ตารางที่ 18 กฎทางภาษาศาสตร์สำหรับระบบ eCSCDA (ต่อ)

ลำดับ	รูปแบบหัวข้อ	ผลลัพธ์
12	($Phrase_1$) + IN + ($Phrase_2$)	• $Phrase_2$ + $Phrase_1$
13	($Phrase_1$) + IN + ($Phrase_2$) + IN + ($Phrase_3$)	• $Phrase_3$ + $Phrase_2$ + $Phrase_1$
14	($Phrase_1$) + CC + ($Phrase_2$)	• $Phrase_1$ • $Phrase_2$
15	($Phrase_1$) + CC + ($Phrase_2$) *if ($Phrase_2$) has one word	• $Phrase_1$ • $Phrase_1$(without last word) + $Phrase_2$
16	($Phrase_1$) + CC + ($Phrase_2$) * if ($Phrase_1$) has one word	• $Phrase_1$ + $Phrase_2$(without first word) • $Phrase_2$
17	($Phrase_1$) + CC + ($Phrase_2$) * if last word of ($Phrase_1$) = last word of ($Phrase_2$)	• $Phrase_1$ • $Phrase_2$
18	($Phrase_1$) + CC + ($Phrase_2$) * if first word of ($Phrase_1$) = first word of ($Phrase_2$)	• $Phrase_1$ • $Phrase_2$

ตารางที่ 18 กฎทางภาษาศาสตร์สำหรับระบบ eCSCDA (ต่อ)

ลำดับ	รูปแบบหัวข้อ	ผลลัพธ์
19	($Phrase_1$) + CC + ($Phrase_2$) * if ($Phrase_1$) has one word and ($Phrase_1$) = first word of ($Phrase_2$)	• $Phrase_2$
20	($Phrase_1$) + IN + ($Phrase_2$) + CC + ($Phrase_3$)	• $Phrase_2$ + $Phrase_1$ • $Phrase_3$ + $Phrase_1$
21	($Phrase_1$) + CC + ($Phrase_2$) + IN + ($Phrase_3$)	• $Phrase_3$ + $Phrase_1$ • $Phrase_3$ + $Phrase_2$

ภาคผนวก ค

เอกสารรับรองผลการพิจารณาจริยธรรมการวิจัยในมนุษย์

บันทึกข้อความ

ส่วนงาน สำนักงานอธิการบดี กองบริหารการวิจัยและนวัตกรรม โทร. ๒๕๖๑ - ๒๕๖๒
ที่ อว ๘๑๐๐/๐๐๔๙๒ **วันที่** ๑๙ มิถุนายน พ.ศ. ๒๕๖๒
เรื่อง ขอแจ้งรับรองโครงการวิจัยที่ส่งมาขอรับการพิจารณาจริยธรรมการวิจัยในมนุษย์ มหาวิทยาลัยบูรพา

เรียน คณบดีคณะวิทยาการสารสนเทศ

ตามที่นักวิจัยในหน่วยงานของท่าน ได้ยื่นเอกสารเพื่อขอรับการพิจารณาจริยธรรมการวิจัยในมนุษย์ มหาวิทยาลัยบูรพา รหัสโครงการวิจัย Sci 056/2562 โครงการวิจัย การพัฒนาโมเดลสำหรับการวิเคราะห์และเปรียบเทียบคำอธิบายรายวิชาจากมหาวิทยาลัยต่าง ๆ : กรณีศึกษารายวิชาทางด้านวิทยาการคอมพิวเตอร์ โดยมี นายพีระพล กำลังพืช คณะวิทยาการสารสนเทศ เป็นหัวหน้าโครงการวิจัย นั้น

บัดนี้ คณะกรรมการพิจารณาจริยธรรมการวิจัยในมนุษย์ มหาวิทยาลัยบูรพา ได้พิจารณาตามวิธีดำเนินการมาตรฐาน (Standard Operating Procedures, SOP) ฉบับที่ ๑ พ.ศ. ๒๕๖๐ ที่ได้ประกาศใช้เมื่อวันที่ ๙ มกราคม พ.ศ. ๒๕๖๐ แล้วว่า โครงการวิจัยดังกล่าวเป็นโครงการวิจัยที่สามารถให้การรับรอง โดยยกเว้นการลงมติจากที่ประชุม (Exemption Determination) ตามข้อที่ ๖ คือ การวิจัยที่เก็บข้อมูลจากฐานข้อมูลที่เปิดเผยต่อสาธารณชน เช่น เว็บไซต์ ประกาศของหน่วยงานต่าง ๆ ฯลฯ จึงเห็นสมควรให้ดำเนินการวิจัยได้ พร้อมนี้ ได้แนบเอกสารรับรองผลการพิจารณาจริยธรรมการวิจัยในมนุษย์ (หมายเลขใบรับรองที่ ๑๐๓/๒๕๖๒) มายังท่าน เพื่อแจ้งนักวิจัยที่มีรายชื่อข้างต้น นำไปใช้ในการเก็บข้อมูลจริงจากผู้เข้าร่วมโครงการวิจัยต่อไป โดยห้ามนักวิจัยเบี่ยงเบนรายละเอียดต่างๆ ของโครงการวิจัยที่ยื่นมาขอรับการพิจารณาจริยธรรมการวิจัยในมนุษย์ มหาวิทยาลัยบูรพา และเมื่อนักวิจัยดำเนินการวิจัยเสร็จเรียบร้อยแล้ว ขอให้แจ้งปิดโครงการวิจัยมายังคณะกรรมการพิจารณาจริยธรรมการวิจัยในมนุษย์ มหาวิทยาลัยบูรพา ด้วย

จึงเรียนมาเพื่อโปรดทราบ

วิทวัส แจ้ง

(ผู้ช่วยศาสตราจารย์ ดร.วิทวัส แจ้งเอี่ยม)
ประธานคณะกรรมการพิจารณาจริยธรรมการวิจัยในมนุษย์
มหาวิทยาลัยบูรพา

ที่ ๑๐๓/๒๕๖๒

เอกสารรับรองผลการพิจารณาจริยธรรมการวิจัยในมนุษย์
มหาวิทยาลัยบูรพา

คณะกรรมการพิจารณาจริยธรรมการวิจัยในมนุษย์ มหาวิทยาลัยบูรพา ได้พิจารณาโครงการวิจัย

รหัสโครงการวิจัย	Sci 056/2562
โครงการวิจัยเรื่อง	การพัฒนาโมเดลสำหรับการวิเคราะห์และเปรียบเทียบคำอธิบายรายวิชาจากมหาวิทยาลัยต่าง ๆ : กรณีศึกษารายวิชาทางด้านวิทยาการคอมพิวเตอร์
หัวหน้าโครงการวิจัย	นายพีระพล กำลังพืช
หน่วยงานที่สังกัด	นิสิตระดับบัณฑิตศึกษา คณะวิทยาการสารสนเทศ

คณะกรรมการพิจารณาจริยธรรมการวิจัยในมนุษย์ มหาวิทยาลัยบูรพา ได้พิจารณาแล้วเห็นว่าโครงการวิจัยดังกล่าวเป็นไปตามหลักการของจริยธรรมการวิจัยในมนุษย์ โดยที่ผู้วิจัยเคารพสิทธิและศักดิ์ศรีในความเป็นมนุษย์ ไม่มีการล่วงละเมิดสิทธิ สวัสดิภาพ และไม่ก่อให้เกิดภยันตรายแก่ตัวอย่างการวิจัยและผู้เข้าร่วมโครงการวิจัย

จึงเห็นสมควรให้ดำเนินการวิจัยในขอบข่ายของโครงการวิจัยที่เสนอได้ (ดูตามเอกสารตรวจสอบ)

๑. เอกสารโครงการวิจัยฉบับภาษาไทย ฉบับที่ ๑ วันที่ ๑๓ เดือน มิถุนายน พ.ศ. ๒๕๖๒
๒. เอกสารชี้แจงผู้เข้าร่วมโครงการวิจัย ฉบับที่ - วันที่ - เดือน - พ.ศ. -
๓. เอกสารแบบแสดงความยินยอมของผู้เข้าร่วมโครงการวิจัย
ฉบับที่ - วันที่ - เดือน - พ.ศ. -
๔. เอกสารแสดงรายละเอียดเครื่องมือที่ใช้ในการวิจัยซึ่งผ่านการพิจารณาจากผู้ทรงคุณวุฒิแล้ว หรือชุดที่ใช้เก็บข้อมูลจริงจากผู้เข้าร่วมโครงการวิจัย ฉบับที่ - วันที่ - เดือน - พ.ศ. -

การรับรองผลการพิจารณาจริยธรรมการวิจัยในมนุษย์ฉบับนี้ มีผลถึงวันที่ ๑๒ เดือน มิถุนายน พ.ศ. ๒๕๖๓

ออกให้ ณ วันที่ ๑๓ เดือน มิถุนายน พ.ศ. ๒๕๖๒

ลงนาม
(ผู้ช่วยศาสตราจารย์ ดร.วิทวัส แจ้งเอี่ยม)
ประธานคณะกรรมการพิจารณาจริยธรรมการวิจัยในมนุษย์
มหาวิทยาลัยบูรพา

ภาคผนวก ง

เอกสารเผยแพร่ผลงานวิจัย

A new system for analyzing contents of Computer Science courses

Peerapon Kamlangpuech, Komate Amphawan
Computational Innovation Laboratory, Faculty of Informatics, Burapha University, Chonburi, 20131, Thailand
Email: peerapon.peetom@gmail.com, komate@gmail.com

2020 7th International Conference on Advance Informatics: Concepts, Theory and Applications (ICAICTA) | 978-1-7281-8038-0/20/$31.00 ©2020 IEEE | DOI: 10.1109/ICAICTA49861.2020.9429072

***Abstract*—With the growth of technology and the changing of human behavior, the fields of Computer Science (CS), Information Technology (IT), and Data Science (DS) become more popular and should be learned by children, students, and people. Consequently, institutes widely initiate new online/offline courses to serve this need. However, this leads to educate different knowledge and skills in which some of them may not match the needs of companies for hiring employees. To address this issue, we introduce a new system for analyzing content on CS courses, called *CSCDA system* (*Computer Science Course Description Analysis system*). The system can identify similar and dissimilar contents belonging to two CS courses by applying text processing techniques and keyword similarity matching. These identified contents can help to set up a standard, to improve integrity and quality, and to reduce redundancy of the contents will be taught in the courses. Experimental studies are conducted on CS courses of Thai Universities to investigate the effectiveness of the *CSCDA* system based on four measures, *i.e.* percentage of similar contents, precision, recall, and F-measure, respectively. Last, a comparative study is performed and the result shows that our proposed method can effectively analyze course contents and outperforms the other keyword extraction methods.**

***Index Terms*—Content analysis, Course description, Computer Science course**

I. INTRODUCTION

Currently, new computer technologies and innovations have emerged each day. Many software, mobile applications, and devices are continuously launched into the market. This makes the need for companies to hire more employees having skills related to the fields of Computer Science (CS), Information Technology (IT), and/or Data Science (DS), respectively. Consequently, this leads to emerging of a trend to study topics/subjects in the CS's related fields such as programming, software development, Artificial Intelligence, Machine Learning, Natural Language Processing, and so on. Based on this trend, institutes or companies initiate new online/on-site courses to serve the need of people. However, for now, it is overwhelming of curriculums and courses which leads to educate different knowledge and skills in which some of them may not match the needs of companies.

To overcome this issue, there are efforts to investigate consistency between subjects in CS and the Thailand Qualification Framework of Higher Education (TQF: HEd) which can help to improve the standard of the curriculum of Thai universities [1], [2], [3]. These models consider contents in the course description and then map each content into a class of "Body of knowledge" (by applying *semantic-based* and *structure-based ontology mapping*) to know consistency between the content and Body of knowledge assigned in TQF of Computer Science. Moreover, there are approaches applying Bloom's Taxonomy to assess the learning objectives of CS courses [4], [5], [6]. For each topic in a course, its level of knowledge is considered and map to Bloom's levels consisting of recall, comprehension, application, analysis, synthesis, and evaluation, respectively. This can help to assess students' performance and to generate reports of difficulties including the variety of causes hypothesized and solutions adopted. Last, there is an approach to detect similarity through academic content by applying semantic technologies and text mining [7]. This considers career name, course name, course description, and topics to find similar academic contents to solve Students' mobility and credit validation as input. Then, the ontological model and RDF-Ization are applied to regard the semantic of contexts used for similarity calculation. However, the previous works mentioned above do not address on computer technology domain which contains lots of reserved words such as 'Data Science', 'Big data', 'NLTK', 'Natural Language Processing', etc. This causes the missing focus on the important contents that should be considered and loosing of similarity matching and similar context.

Thus, to address the above issue, we introduce a new system for analyzing and comparing course descriptions in the CS domain, called the *CSCDA system* (*Computer Science Course Description Analysis system*). Based on the proposed system, similar (and dissimilar) contents of two course descriptions on a subject (or a pair of related subjects) are identified. These contents can help to check for redundancy, popularity, integrity, and quality of contents in the course descriptions. Last, the number (percentage) of similar contents is calculated in order to draw a conclusion to the course descriptions. To investigate the efficiency of the *CSCDA* system, experiments were conducted on CS course descriptions Burapha University in comparison with eight of top-ten universities in Thailand (ranked in Asian University Ranking.[1]). Two sets of experiments were done to observe the level of similarity of any pair of course descriptions and popular contents usually exist in course descriptions of a subject. In addition, three famous measures, precision, recall, and F-measure are applied to observe the effectiveness of the *CSCDA* system in comparison with two well-known keyword extraction techniques.

[1]asian-university-rankings/2020

978-1-7281-8038-0/20/$31.00 ©2020 IEEE

II. RELATED WORK

Currently, text similarity is applied in several tasks such as text classification, document clustering, information retrieval, topic detection, topic tracking, question answering, question generation, short answer scoring, and text summarization. From the survey of text similarity [8], [9], its measure can be categorized into 4 groups of approaches as follows.

1) String-Based Similarity: operates on string sequences and character combinations. A string metric aims to measure similarity or dissimilarity between two strings for approximate string comparison or matching. The string-based metric can be separated into 2 categories as follows

- *Character-based Measures*:–considers the chain of characters (or words) that are similar or pretty similar such as *Longest Common Substring (LCS)*, *Damerau-Levenshtein*, *N-gram*, *Jaro*, *Jaro–Winkler*, *Needleman-Wunsch*, and *Smith-Waterman*, etc.
- *Term-based Measures*:–calculates the distance between two words by considering several factors *e.g. Manhattan distance* (or *Block Distance*), *Cosine similarity*, *Dice's coefficient*, *Euclidean distance*, *Jaccard similarity*, *Matching Coefficient*, and *Overlap coefficient*.

2) Corpus-based Similarity: is a semantic measure used for determining the similarity between words according to information taken from text corpora (where a Corpus contains large collection texts). The famous corpus-based similarities are *Hyperspace Analogue to Language (HAL)*, *Latent Semantic Analysis (LSA)*, *Generalized Latent Semantic Analysis (GLSA)*, *Explicit Semantic Analysis (ESA)*, *cross-language explicit semantic analysis (CLESA)*, *Pointwise Mutual Information - Information Retrieval (PMI-IR)*, *Second-order co-occurrence pointwise mutual information (SCO-PMI)*, and *Normalized Google Distance (NGD)*, respectively.

3) Knowledge-Based Similarity: is based on the identification of the degree of similarity between words by using information taken from semantic networks. WordNet is the most popular and well-known semantic network containing nouns, verbs, adjectives, and adverbs grouped into sets of cognitive synonyms (or synsets). In addition, synsets are interlinked by means of lexical relations and conceptual-semantic.

Knowledge-based similarity can be divided into two groups *i.e.* semantic similarity measure and semantic relatedness measure. Semantic similarity measure relates to the basis of concepts' likeness and covers a broad range of relationships between concepts such as is-a-kind-of, is-a-specific-example-of, is-a-part-of, is-the-opposite-of. Meanwhile, semantic relatedness is a more general notion of relatedness which is not specifically tied to the shape or form of the concepts.

4) Hybrid Similarity: combines several similarity measures. For example, it combines two corpus-based measures with other three knowledge-based measures. With this, these measures were applied separately, then they were combined.

From above, we applied a hybrid similarity measure that combines string-based similarity (both on character-based similarity measures and term-based similarity measures) with the concept of knowledge-based measure (semantic similarity measure: is-a-part-of). By this, we can calculate the similarity of two words (or phrases) within four cases.

III. PROPOSED METHOD : THE *CSCDA* SYSTEM

Details of the *CSCDA* system is now presented. As shown in Fig. 1, the system consists of three main steps which can be described as follows.

A. Input gathering

The *input gathering* process is to collect and preprocess on CS course descriptions. For a CS curriculum of a university u, all subjects (excepts general education) are considered as subjects on CS. Then, for each subject s, its English course description $c_{s,u}$ is collected in the CS course description corpus (also called *CS-CDC* for short)[2] in the form of 3-tuple $\langle s, u, cc\rangle$ where s is the subject name, u is the name of the university and cc is contents of the course description, respectively. However, before storing $c_{s,u}$ into *CS-CDC*, its course content cc is split into a set of topics indicating a group of contents (chapters or lessons) would be taught. Therefore, the course description $c_{s,u}$ is thus collected as $c_{s,u} = \langle s, u, TP_{s,u}\rangle$ where $TP_{s,u} = \{tp_1, tp_2, \ldots, tp_n\}$ is the set of n topics of $c_{s,u}$. For example, as in Fig. 2, the course description on 'Algorithm Design and Applications' of Burapha University is collected as a 3-tuple where the course content is split into 12 main topics.

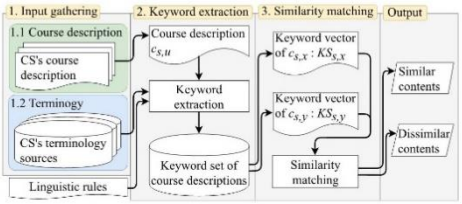


Fig. 1: The architecture of the *CSCDA* system

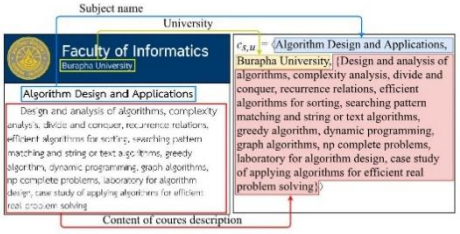


Fig. 2: Example of collecting a course description the corpus

Besides, to understand important contexts of each topic on course description, text analysis is performed by recognizing and identifying reserved words and keywords in the

[2]CS's course description corpus

topic. By this, $28,392$ CS-terminologies are gathered from eight well-known sources *i.e.* Oxford-A Dictionary of Computer Science, Labautopedia, TechTarget, ComputerHope, PC Glossary, Glossary of Computer Related Terms, Wikipedia's Glossary of computer science, GCSE's Computer Science Glossary and collected into our corpus[3], represented as a set $T = \{t_1, t_2, \ldots, t_z\}$ where $t_x \in T$ is a CS-terminology.

B. Keyword Extraction

As CS course descriptions are collected, each course description $c_{s,u} = \langle s, u, TP_{s,u} \rangle \in$ *CS-CDC* is thus analyzed. Each topic $tp_i \in TP_{s,u}$ is considered and processed by text processing techniques, *e.g.* lowercase conversion, error correction (Spelling Mistake Correction, SMC [10]), stop word removal, POS tagging (Stanford POS Tagger [11]). Next, any the reserved word hidden in tp_i is recognized by considering an n-gram of words and then comparing it with a terminology $t_x \in T$. The n-gram will be labeled as 'RW' if it matches with t_x. Otherwise, word stemming and lemmatization are performed and the matching with terminologies procedure is reapplied.

Consequently, important keywords are thus identified by applying linguistic rules of [12][4] (labeled as 'KW'). These keywords can help to explore essential contexts extending (or different) from the identified reserved words. Subsequently, three text processing procedures are designed and applied to remove trivial contexts in the topic tp_i, *i.e. i*) *adjective removal* –remove all unimportant adjectives, *ii*) *sub-keyword removal*– eliminate redundant keywords and *iii*) *subject-name removal*– remove the subject name to recognize contexts differing from the subject name, respectively.

Last, all reserved words and keywords in tp_i are collected in the keyword vector KV_{tp_i}. Thus, for now, each course description $c_{s,u}$ can be regarded as $c_{s,u} = \langle s, u, KS_{c_{s,u}} = \{KV_{tp_1}, \ldots, KV_{tp_n}\} \rangle$ where $KS_{c_{s,u}}$ is a keyword set containing keyword vectors and each keyword vector KV_{tp_p} contains a set of reserved words and keywords occurring in the topic tp_p.

Example: Figure 3 illustrates how the *keyword extraction* works. From the figure, the topic tp_5 = "*efficient algorithms for sorting*" is considered to extract reserved words and keywords. After working on text processing, tp_5 is still the same. Then, the word 'sorting' is recognized as a reserved word and labeled as 'RW'. Word stemming and lemmatization is applied where 's' is removed from the word 'algorithms'. Then, reserved word identification is reapplied to label the word 'algorithm' as 'RW'. Consequently, linguistic rules are applied to extract and maintain keywords to be $KV_{tp_5} = \langle$'efficient algorithm(KW)', 'sorting efficient(KW)', 'sorting algorithm(KW)'$\rangle$. In addition, *adjective removal* is applied where the word 'efficient' is removed from all keywords. Next, *sub-keyword removal* is executed to removes two redundant keywords, *i.e.* 'algorithm(KW)' and 'sorting(KW)' which are sub-keywords of the keyword 'sorting algorithm(KW)'. Last, subject removal is performed causing the removal of the word 'algorithm'. Finally, the keyword vector KV_{tp_5} contains $\langle$'sorting(KW)'$\rangle$.

[3]CS-terminology corpus
[4]Linguistic rules

C. Similarity matching

To investigate similar contents, let's consider two course descriptions $c_{s,x} = \langle s, x, KS_{c_{s,x}} = \{KV_{tp_1}, \ldots, KV_{tp_n}\} \rangle$ and $c_{s,y} = \langle s, y, KS_{c_{s,y}} = \{KV_{tp_1}, \ldots, KV_{tp_m}\} \rangle$ where the former is regarded as the *initial course description* and the latter is considered as the *compared course description*. Each keyword vector $KV_{tp_p} \in KS_{c_{s,x}}$ of $c_{s,x}$ is considered and then each of its keywords, w_i is regarded and compared with a keyword w_j in a keyword vector $KV_{tp_q} \in KS_{c_{s,y}}$ of $c_{s,y}$. Next, the similarity between these two keywords is calculated based on the following three cases: *i*) w_i is equal to w_j (also for the case that w_i is a paraphrase of w_j), *ii*) w_i is a subset of w_j and *iii*) w_i is a superset of w_j, can be defined as

$$sim(w_i, w_j) = \begin{cases} 1 & , w_i = w_j, w_i \subset w_j \text{ or } w_i \supset w_j \\ 0 & , \text{otherwise} \end{cases} \quad (1)$$

In addition, if w_i is similar to w_j, it can be concluded that the topic $tp_p \in c_{s,x}$ (containing w_i) matches with the topic $tp_q \in c_{s,y}$ (containing w_j) and the topic tp_p can be marked as a *matched topic*, represented as a $match(tp_p) = 1$. On the other hand, if all keywords contained in KV_{tp_p} belonging to the topic tp_p do not similar to any keyword of any topic of the course description $c_{s,y}$, it can be marked as a *not-match topic*, represented as a $match(tp_p) = 0$. Thus, the matching of the topic tp_p of $c_{s,x}$ with any topic tp_q of $c_{s,y}$ can be defined as

$$match(tp_p) = \begin{cases} 1 & , \exists w_i \in KV_{tp_p} | sim(w_i, w_j) = 1 \\ & \text{where } w_j \in KV_{tp_q} \text{ of } tp_q \wedge tp_q \in c_{s,y} \\ 0 & , \text{otherwise} \end{cases} \quad (2)$$

After considering all words of all topics belonging to the course description $c_{s,x}$, the percentage of similar contents between the course description $c_{s,x}$ and $c_{s,y}$ can be computed by the ratio between the number of topics in $c_{s,x}$ matching with topics in $c_{s,y}$ and the total number of topics in $c_{s,x}$, defined as

$$sim(c_{s,x}, c_{s,y}) = \frac{\sum_{i=1}^{n} match(tp_i)}{n} \quad (3)$$

Last, after performing all processes of the *CSCDA* system to analyze a pair of course descriptions, the set of similar contents (matched topics), and *ii*) the set of dissimilar contents (not-matched topics) are returned. Moreover, if there are several course descriptions based on the same subject, the system can suggest popular keywords mostly occurring in the course descriptions.

Example: Fig 4 demonstrates the similarity calculation of 'Algorithm' 's course descriptions from Burapha (BUU) and Chiang Mai University (CMU). By this, each keyword vector of BUU is considered and each word is compared to keywords of CMU. The Keyword 'analysis' in KV_{tp_1} is not similar to any keyword of CMU. Thus, the topic tp_1 of BUU is identified as dissimilar content. Meanwhile, the keyword 'divide and

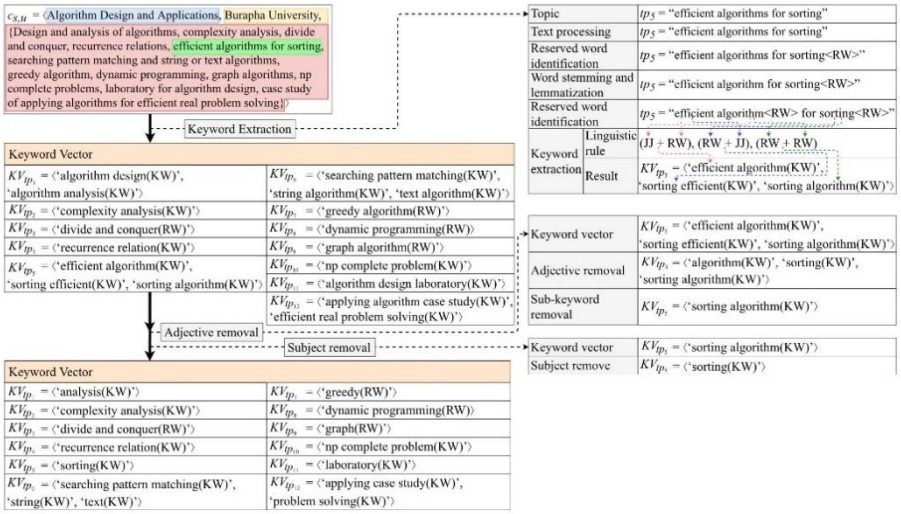


Fig. 3: Example of keyword extraction from 'Algorithm Design and Applications' of Burapha University

conquer' in KV_{tp_3} is equal to that of KV_{tp_5} of CMU. Therefore, the topic tp_3 of BUU is identified as similar content. Last, the keyword 'recurrence relation' in KV_{tp_4} is a subset of 'solving recurrence relation' causing the topic tp_4 of BUU is identified as similar content.

IV. Experimental study

In this section, experimental studies are performed and described to investigate the effectiveness of the *CSCDA* system. As such, 547 English course descriptions are collected from nine Thai universities including Burapha University (BUU, 68 courses) and eight of top-ten universities ranked in Asian university ranking *i.e.* Chulalongkorn University (CU, 46 courses), Mahidol University (MU, 35 courses), Chiang Mai University (CMU, 57 courses), Thammasat University (TU, 84 courses), Kasetsart University (KU, 69 courses), King Mongkut's University of Technology Thonburi (KMUTT, 40 courses), Prince of Songkla University (PSU, 61 courses), and King Mongkut's Institute of Technology Ladkrabang (KMITL, 87 courses), respectively. Besides, 28,392 CS terminologies are collected from the eight well-known sources (as mentioned and detailed in section III-A). In our experiments, each course description of BUU is regarded as an initial course description and then compared with one on the same subject belonging to another university. Then, we applied four criteria to investigate the effectiveness of the *CSCDA* system, *i.e. i*) percentage of similar contents (Eq. 3), *ii*) precision (Eq. 4), *iii*) recall (Eq. 5), and *iv*) F-measure (Eq. 6), respectively.

$$precision = \frac{number\ of\ correct\ matches\ retrieved}{number\ of\ matches\ get\ from\ the\ model} \quad (4)$$

$$recall = \frac{number\ of\ correct\ matches\ retrieved}{number\ of\ correct\ matches} \quad (5)$$

$$F\text{-}measure = 2 \times \frac{precision \times recall}{precision + recall} \quad (6)$$

Figure 5 shows the performance of the *CSCDA* system based on considering the course description of the 'Algorithm' of BUU comparing with that of other universities. First, Fig. 5(a) demonstrates the percentage of similar contents separated by two matching cases: *i*) equal matching or exact matching (including paraphrase matching) and *ii*) sub/superset matching. First, it can be seen that the contents of BUU and CU are not identical due to that of BUU is written in detail whereas that of CU is described in the big picture. Moreover, it can be observed that sub/superset matching can help to improve the percentage of similarity matching up to 25% ($\approx$ 14% on average). Apart from that, Fig. 5(b) presents precision, recall, and F-measure that are likely high tendency (between 80 - 100%). Consequently, the table I indicate precision, recall, and F-measure of the *CSCDA* system in comparison with *TerMine* and *RAKE*. By this, all course descriptions from *BUU* are compared to that of same subjects from other universities. For example, there are 16 common subjects between *BUU* and *CU*. Then, all pairs of common course descriptions are thus compared to observe the average percentage of similar contents, precision, recall, and F-measure of the three approaches. From

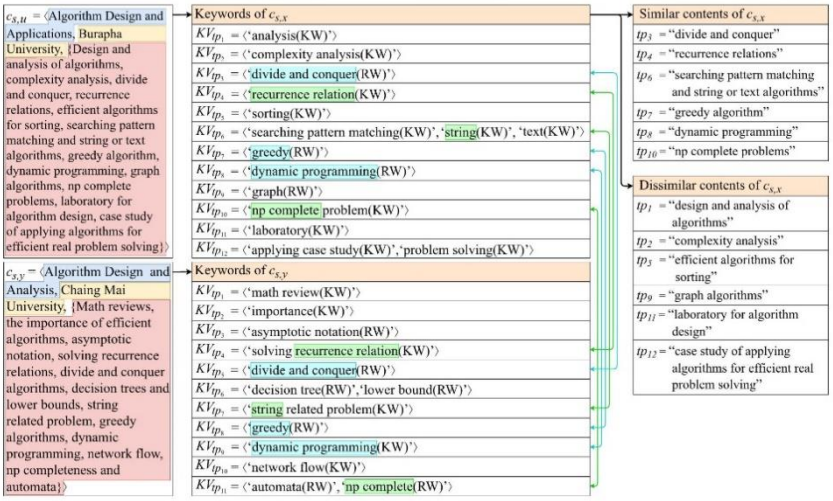
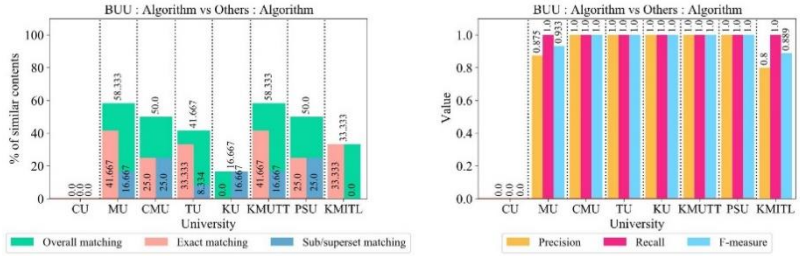


Fig. 4: Similarity matching of 'Algorithm Design and Applications' between Burapha University and Chaing Mai University



(a) Percentage of similar contents (b) Precision, recall, and F-measure

Fig. 5: *CSCDA* : Considering the course content of Algorithm belonging to BUU in comparison with the others

the table, it can be seen that the *CSCDA* system outperforms other approaches for all criteria. The average percentage of similar contents can significantly be improved in between 1 - 19% ($\approx$ 16% against *TerMine* and $\approx$ 3% against *RAKE*). In the meantime, the average precision, recall and F-measure are also improved by 2 - 49% ($\approx$ 38% against *TerMine* and $\approx$ 17% against *RAKE*), 1 - 65% ($\approx$ 54% against *TerMine* and $\approx$ 6% against *RAKE*) and 2 - 58% ($\approx$ 47% against *TerMine* and $\approx$ 13% against *RAKE*).

Figure 6 shows the word cloud indicating the popularity of keywords occurring in the Algorithm's course descriptions from all universities. The keywords, 'divide and conquer', 'dynamic programming' and 'greedy', are most popular (occur seven of nine). Meanwhile, the keywords such as 'complexity analysis', 'network flow', 'brute force' have the least occurrence (occur only twice and shown in orange). With this information, it can help to further analyze differences between contents in a course description from one university and the popular keywords used in other universities. This can also help to improve the integrity and quality of contents in the course description to get the same standard as other universities.

Last, Fig. 7 demonstrates the performance of the *CSCDA* system on keyword extraction and redundancy reduction. By this, the numbers of keywords extracted from the Algorithm's course descriptions from all universities are investigated (as shown in yellow bar). Moreover, thanks to the three text pro-

TABLE I: Computational performance of *CSCDA* against *TerMine* and *RAKE*

BUU vs	Percentage			Precision			Recall			F-measure		
	CSCDA	*TerMine*	*RAKE*	*CSCDA*	*TerMine*	*RAKE*	*CSCDA*	*TerMine*	*RAKE*	*CSCDA*	*TerMine*	*RAKE*
CU (16)	**21.968**	9.331	20.805	**0.718**	0.392	0.628	**0.792**	0.396	0.705	**0.714**	0.383	0.691
MU (11)	**34.861**	15.298	29.090	**0.651**	0.478	0.632	**0.909**	0.335	0.833	**0.749**	0.382	0.684
CMU (15)	**31.086**	11.826	30.310	**0.968**	0.522	0.789	**0.983**	0.400	0.893	**0.955**	0.435	0.826
TU (20)	**23.350**	11.312	20.340	**0.766**	0.333	0.514	**0.794**	0.317	0.743	**0.769**	0.306	0.574
KU (19)	**27.938**	8.623	25.093	**0.846**	0.355	0.599	**0.882**	0.249	0.833	**0.859**	0.273	0.685
KMUTT (16)	**34.872**	16.791	32.985	**0.778**	0.453	0.739	**0.833**	0.307	0.830	**0.791**	0.345	0.614
PSU (21)	**26.682**	7.651	24.173	**0.869**	0.417	0.583	**0.976**	0.326	0.907	**0.909**	0.335	0.668
KMITL (19)	**25.270**	10.158	22.420	**0.803**	0.426	0.556	**0.772**	0.313	0.759	**0.777**	0.319	0.723

Fig. 6: The word cloud from matching in 'Algorithm' subject

cess techniques, *i.e. adjective removal*, *sub-keyword removal*, and *subject-name removal*, help to remove unimportant and redundant keywords by 1 - 9 keywords (up to 50% as shown by whisker plots).

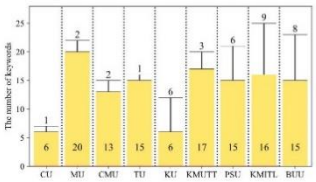


Fig. 7: The number keywords extracted from 'Algorithm' course description

V. Conclusion

In this paper, a new system for analyzing CS course descriptions, named *CSCDA* is introduced. The system can help to check for redundancy, popularity, integrity, and quality of contents. In addition, the percentage of similar contents between course descriptions is calculated in order to draw a conclusion. Based on the *CSCDA* system, CS course descriptions are first gathered and CS terminologies are collected. Simple text processing is applied and reserved word and keyword extraction techniques are designed to recognize important contexts. Besides, three additional techniques are devised to remove trivial and redundant contexts. Last, a similarity matching technique is explored to identify similar (dissimilar) contents. Experiments were done on 547 CS course descriptions gathered from nine Thai Universities. The results show that the proposed *CSCDA* can efficiently analyze course contents with high precision, recall, and F-measure in comparison with the other two well-known keyword extraction techniques.

Acknowledgment

This work was supported by a research grant of Faculty of Informatics, Burapha University (Grant No. 002/2562).

References

[1] C. Nuntawong, C. S. Namahoot, and M. Brückner, "A semantic similarity assessment tool for computer science subjects using extended wu & palmer's algorithm and ontology," in *Information Science and Applications*, 2015, pp. 989–996.

[2] C. S. N. Chayan Nuntawong and M. Brückner, "Home: Hybrid ontology mapping evaluation tool for computer science curricula," *Journal of Telecommunication, Electronic and Computer Engineering*, vol. 9, no. 2-3, pp. 61 – 65, 2017.

[3] C. Nuntawong, C. S. Namahoot, and M. Brückner, "A web based cooperation tool for evaluating standardized curricula using ontology mapping," in *Cooperative Design, Visualization, and Engineering*, 2016, pp. 172–180.

[4] C. W. Starr, B. Manaris, and R. H. Stalvey, "Bloom's taxonomy revisited: Specifying assessable learning objectives in computer science," *SIGCSE Bull.*, vol. 40, no. 1, p. 261–265, 2008.

[5] S. Masapanta-Carrión and J. A. Velázquez-Iturbide, "A systematic review of the use of bloom's taxonomy in computer science education," in *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, 2018, p. 441–446.

[6] A. Pawar and V. Mago, "Similarity between learning outcomes from course objectives using semantic analysis, blooms taxonomy and corpus statistics," *ArXiv*, vol. abs/1804.06333, 2018.

[7] G. O. M. O. N. P. V. Saquicela, F. Baculima and M. Espinoza, "Similarity detection among academic contents through semantic technologies and text mining," in *Proceedings INFOBAE Cuba*, 2018, pp. 1–12.

[8] W. H. Gomaa, A. A. Fahmy *et al.*, "A survey of text similarity approaches," *International Journal of Computer Applications*, vol. 68, no. 13, pp. 13–18, 2013.

[9] M. Vijaymeena and K. Kavitha, "A survey on similarity measures in text mining," *Machine Learning and Applications: An International Journal*, vol. 3, no. 2, pp. 19–28, 2016.

[10] S. Sharma and S. Gupta, "A correction model for real-word errors," *Procedia Computer Science*, vol. 70, pp. 99 – 106, 2015.

[11] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer, "Feature-rich part-of-speech tagging with a cyclic dependency network," in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, 2003, pp. 173–180.

[12] B. Chaisoongnoen, K. Amphawan, and A. Bunpeng, "Supplementary book suggestion for computer science courses," in *2018 5th International Conference on Advanced Informatics: Concept Theory and Applications (ICAICTA)*, 2018, pp. 84–90.

eCSCDA: An efficient system for analyzing contents of Computer Science Courses

Peerapon Kamlangpuech
Computational Innovation Laboratory,
Faculty of Informatics, Burapha University
Chonburi, 20131, Thailand
Email: peerapon.peetom@gmail.com

Komate Amphawan
Computational Innovation Laboratory,
Faculty of Informatics, Burapha University
Chonburi, 20131, Thailand
Email: komate@gmail.com

***Abstract*—This paper aims to introduce a new efficient system, called *eCSCDA*, to efficient analyze Computer Science (CS) course description. The primary task of the system is to identify similar (and dissimilar) contents amount two (or a group of) CS course descriptions which can help to know similar and different focuses on important contents to teach to students. Moreover, it can help to check for integrity and quality and to set up a standard of teaching contents of the course. In *eCSCDA*, text processing procedure is newly rearranged and developed. Besides, the linguistic rules and their derivation are newly updated and applied to extracted important keywords. Moreover, two synonym corpuses, terminology and word synonyms, are recently designed and collected to consider synonyms of the keywords hidden in the course descriptions. Last, to efficiently identify similar contents, two new matching techniques, sub-keyword and semantic matching techniques, are designed and applied together with exact and subset (superset) matching methods. Experiments were conducted on CS course contents gathered from nine Thai Universities to examine the effectiveness of our proposed system in comparison with previous system and related methodologies. From the results, it shows that *eCSCDA* is efficient to analyze the course contents and outperforms other related systems in various terms *e.g.* percentage of similar contents, precision, recall and F-measure, respectively.**

Keywords–Content analysis, Course description, Computer Science course

I. Introduction

Based on the rapid growth of computer technology, new concepts, tools, software, applications, programming languages and libraries are being developed each day. This led to the boom of learning in computer-related fields such as Computer Science (CS), Information Technology (IT), Computer Engineering (CE), Software Engineering (SE), Data Science (DS) and so on. With these emergences, institutes and universities should create news, revise and/or update their current curriculum in order to to keep up with the world. Some courses might be newly created in the curriculum meanwhile, some might be modernized with new contents. In addition, there are some ideas that core courses of the curriculum should be standardized. With these ideas, there are approaches to observe consistency between courses in the curriculum and that of *TQF:HEd* (*Thai Qualification Framework of Higher Education*) [1], [2], [3]. These approaches take contents of each course into account and then map them into a class of "Body of knowledge" by applying *semantic-based* and *structure-based ontology mapping*. Besides, Bloom's Taxonomy is applied to assess learning objectives of CS courses [4], [5], [6]. From these approaches, each of topic of teaching contents is considered and then its level of knowledge is extracted and mapped to Bloom's Taxonomy levels (*i.e. i*) recall, *ii*) comprehension, *iii*) application, *iv*) analysis, *v*) synthesis and *vi*) evaluation, respectively). This can assist to assess students' performance and report difficulties of varieties of causes hypothesized and solutions adopted. Moreover, there is an effort to calculate similarity among teaching contents [7] by considering career name, course name, course description and contents to solve students' mobility and credit validation.

From the above, focusing on checking the contents with Thai Quality framework is quite out of date since TQF:HEd is not updated. Moreover, one with focusing on finding similarity among teaching contents does not focus on the computer technology curriculum which have lots of special terminologies, reserved words and abbreviations. This may lead to losing of focusing on the important contents that should be considered and losing efficiency of similarity matching and extracting similar contexts from comparing two course contents. From these issues, a system, called *CSCDA* (Computer Science Course Description Analysis), is introduced [8]. The *CSCDA* takes two course descriptions (either on the same or different subjects) as input. It then performs text processing and extracts important contents (*i.e.* keywords) from each course description. Next, keywords of one course description are compared with ones from another course description to gain similar/dissimilar contents and to calculate level of similarity between the two course contents. These information can help to investigate and check for redundancy, integrity, popularity and quality of contents in the course descriptions. However, even *CSCDA* can well perform in analyzing course contents, but it still has not high precision and recall due to it applies only lexical similarity matching. Thus, there is room to improve the ability of the *CSCDA* system by considering semantic of contents of the course description.

Thus, this paper aims to improve the efficiency of the *CSCDA* by introducing a new improved system, called *eCSCDA* (efficient *CSCDA)*. In *eCSCDA*, text processing procedure is revised in order to efficiently collect important

978-1-6654-1743-3/21/$31.00 ©2021 IEEE

words. Terminology detection is improved by doing twice, once before and after word stemming & lemmatization. New linguistic rules and their derivations are applied to accurately extract keywords. Two synonyms corpuses are newly prepared to consider the semantic of contents. Last, two new matching methods are designed and applied together with two existing matching techniques used in the *CSCDA* system. Experiments were done on CS course descriptions gathered from nine Thai universities. Then, the percentage of similar contents, precision, recall and F-measure are applied to investigate efficiency of the proposed *eCSCDA* in comparison with *CSCDA* and *Word2Vec* [9]. From the results, it is shown that *eCSCDA* outperforms the others on all measures.

II. Related work

Measuring the similarity of texts based on considering words, sentences, paragraphs, and documents is an important task. It is widely applied in many tasks of information retrieval, automatic question-answering, machine translation, dialogue systems, document matching, plagiarism detection, text summarization, and so on. The similarity calculation is divided into 2 groups [10] as discussed as follows.

1) *Text distance*–describes the proximity between two texts, words, or phrases from the perspective of distance. There are three categories of text distance described as follows:
   - *Length Distance*–calculates similarity from the distance of two texts using numerical characteristics, *e.g.* Euclidean distance, Cosine similarity, Manhattan distance, etc.
   - *Distribution Distance*–computes similarity by investigating the distribution of texts such as JS divergence, KL divergence, and so on.
   - *Semantic Distance*–considers distance of texts at the semantic level.
2) *Text Representation*–represents the texts as numerical features where texts can be similar in lexically or semantically. Words in texts are lexically similar if they have the same character sequence. Meanwhile, they are semantically similar if they are used in the same way or same context. This technique can be divided into 4 categories.
   a) *String-Based*–measures similarity by considering string sequences and character composition which consisting of *i*) Character-Based–considers similarity between characters *e.g.* editing distance, LCS (longest common substring), and Jaro similarity; and *ii*) Phrase-Based–considers similarity phrase words *e.g.* Jaccard and dice coefficient.
   b) *Corpus-Based*–uses additional information collected in a corpus, *e.g.* textual feature or co-occurrence probability, to calculate similarity *e.g.* distributed representation, bag-of-words model, and matrix factorization methods.
   c) *Semantic Text Matching*–determines similarity of texts by their meaning *e.g.* Single semantic text matching–and Multi-semantic document matching.
   d) *Graph Structure*–calculates text similarity by regarding links between nodes of the graph *e.g.* Knowledge Graph–projects entities and relationships in the graph into a continuous space; and Graph Neural Network–captures dependency of the graph through message transmission between nodes.

From the various methods mentioned above. In this research, we applied String-Based (both on Character-Based and Phrase-Based), Semantic Text Matching (Single Semantic Text Matching), and Graph Structure (Knowledge Graph: is-a-part-of) methods. These methods allow us to calculate similarities between keywords from course descriptions.

III. Proposed system

In this section, components and details of computation of the proposed *eCSCDA* system are described. As in Fig. 1, the system consists of three main procedures as follows.

A. Input and preprocessing

Before getting input, *eCSCDA* prior prepares three corpus and linguistic rules for further computation. First, as in [8], $28,392$ terminologies in Computer domain were gathered from eight well-known sources and stored in terminology corpus. Second, terminology synonym corpus is created by considering each terminology of the terminology corpus and then searched for synonyms from three sources *i.e. Longdo Dictionary*, *google translation corpus* and *Cambridge Dictionary*, respectively. Note that we also tried to consider the other sources but most of them provide too many synonyms with different levels of relevance with the target terminology. Third, word synonym corpus is built by considering each word from *www.dictionary.com* and then looks for its synonyms in the same manner as above. Last, linguistic rules from [11], [12] are applied to extract important contents which are in the form of keywords and/or terminologies.

Next, to feed input to the *eCSCDA* system, two (a group of) CS course descriptions (in English) should be in the form of 2-tuple $\langle s, cc \rangle$ (see Fig. 2) where s is the course name, and cc is teaching contents of the course.

B. Keyword Extraction

When any two course descriptions c_x and c_y (or a group of course descriptions $c_u, c_{u+1}, \ldots, c_v$) are input, their teaching contents cc_x and cc_y (or $cc_u, cc_{u+1}, \ldots, cc_v$) are first considered. Text-processing is performed on cc_x (also for the cc_y) by applying *i*) sentence tokenization, *ii*) word tokenization, *iii*) lowercase conversion *iv*) error correction, *v*) stopword removal, and *vi*) POS tagging, respectively. With these processes, the teaching content cc_x is divided into a set of topics, defined as $cc_x = \{tp_{1,x}, tp_{2,x}, \ldots, tp_{n,x}\}$. Then, each topic $tp_{i,x}$ is decomposed into a sequence of words with its tag to describe its duty in the topic, denoted as $tp_{i,x} = < (w_1^{tp_{i,x}}, tag), (w_2^{tp_{i,x}}, tag), \ldots, (w_n^{tp_{i,x}}, tag) >$.

Next, each n-gram of words of the topic $tp_{i,x}$ is considered to search for CS terminology hidden in the topic. This can help to recognize important contents. In this procedure, terminology

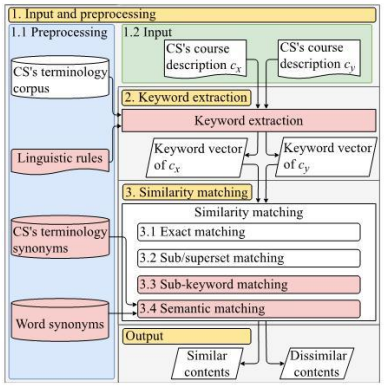


Fig. 1: The framework of the *eCSCDA* system

matching (comparing the n-gram with any terminology prior collected in the terminology corpus where the n-gram is thus grouped together and tagged as "TE" (terminology) if it matches with a terminology) is performed twice, once before and once after doing word stemming & lemmatization. Last, each n-gram of words of the topic $tp_{i,x}$ is reconsidered and the linguistic rules of [12] are thus applied to extract important keywords where a keyword might be in the form of *i*) Terminology, *ii*) Noun, *iii*) Adjective + Terminology, *iv*) Adjective + Noun, *v*) Noun + Noun, *vi*) Noun + Terminology, *vii*) Terminology + Terminology, *viii*) Terminology + Noun, *ix*)JJ + Noun + Terminology, or *X*) Noun + Terminology + Terminology, respectively. After this, a list of keyword of each topic $tp_{i,x}$ is collected as $K^{tp_{i,x}} = \{k_1^{tp_{i,x}}, k_2^{tp_{i,x}}, \ldots, k_n^{tp_{i,x}}\}$, but note that in practice, each topic mostly contains only one or two keywords.

C. Content matching

To recognize similar and dissimilar contents of c_x and c_y, each keyword list, $K^{tp_{i,x}}$ of topic $tp_{i,x}$ extracted from previous step, is considered. It is then compared to any keyword list $K^{tp_{u,y}}$ of $tp_{u,y}$ of course description c_y by the four matching techniques as follows:

1) exact matching – the keyword $K^{tp_{i,x}}$ is exactly the same as the keyword $K^{tp_{u,y}}$ (this also includes the case that $K^{tp_{i,x}}$ is equal to the paraphrase of words in $K^{tp_{u,y}}$),
2) subset matching – the keyword $K^{tp_{i,x}}$ is a subset of the keyword $K^{tp_{u,y}}$ (or $K^{tp_{i,x}}$ is a superset of the keyword $K^{tp_{u,y}}$),
3) sub-keyword matching – a part of the keyword $K^{tp_{i,x}}$ is exactly the same as a part of the keyword $K^{tp_{u,y}}$, and
4) semantic matching –the keyword or a part of the keyword $K^{tp_{i,x}}$ has the same semantic as the keyword or a part of the keyword $K^{tp_{u,y}}$ (Thanks to the terminology and word synonym corpuses), respectively.

From above, if the keyword $K^{tp_{i,x}}$ match with the keyword $K^{tp_{u,y}}$ by exact or subset matching, it can be concluded that the topic $tp_{i,x}$ of the course description c_x is similar to the topic $tp_{u,y}$ of the course description c_y. On the other hand, for matching on sub-keyword or semantic matching, it can be identified that topic $tp_{i,x}$ relates the topic $tp_{u,y}$.

Note that if the topic $tp_{i,x}$ matches with $tp_{u,y}$ by sub-keyword matching, the remaining words of $tp_{i,x}$ should be reconsidered and compared with the remaining words of $tp_{u,y}$ by semantic matching (also for semantic matching and then sub-keyword matching). If all of both keyword lists matches by these two cases, it can be concluded that the topic $tp_{i,x}$ is exactly the same as the topic $tp_{u,y}$.

When, the keyword $K^{tp_{i,x}}$ match with the keyword $K^{tp_{u,y}}$ by one of the four cases above, the matching score between c_x and c_y is set as 1 if the matched keyword is not equal to (or being subset of) the course name of c_x and c_y , calculated as follows :

$$match(tp_{i,x}) = \begin{cases} 1 & ,\{\exists tp_{u,y} \in c_y | K^{tp_{i,x}} \text{ matches with } \\ & K^{tp_{u,y}}, K^{tp_{i,x}} \not\subset s_x, K^{tp_{i,x}} \not\subset s_y\} \\ 0 & ,\text{otherwise} \end{cases}$$

Last, after considering all topics in the course description c_x, the percentage of similar contents between the two course descriptions c_x and c_y can be calculated by

$$per\_sim(c_x, c_y) = \frac{\sum_{i=1}^{n} match(tp_{i,x})}{n} \quad (1)$$

where n is the number of topics in in c_x.

D. Example

Let's consider two course descriptions on "Probability and Statistics" from Burapha University (BUU) and King Mongkut's University of Technology Thonburi (KMUTT)) as shown in Fig. 2(a). First, the course from BUU is divided into 10 and that of KMUTT is also decomposed into 12 topics by applying text-processing as shown in Fig. 2(b). Second, terminologies hidden in each course description are recognized and labeled as "TE" as shown in the red highlight of Fig. 2(c). Third, by applying linguistic rules and their derivations, the two words, { ('descriptive', 'JJ'), ('statistics', 'TE')} of the topic $t_{1,BUU}$ are grouped and identified as a keyword, meanwhile, the tag of each word is still retained for matching procedure (see Fig. 2(d)). Fourth, matching of keywords from the course descriptions is performed, for example, the keyword { descriptive (JJ) statistics(TE)} of the topic $t_{1,BUU}$ matches with the keyword {statistics(TE)} of the topic $t_{1,KMUTT}$ by subset matching. Then, when we look at the matched word, {statistics(TE)} , it is a subset of course name ("Probability and Statistics") where does not indicate important content. It is then eliminated. The topic $t_{2,BUU}$ is also matched with $t_{1,KMUTT}$ by semantic matching, but it is also eliminated since the match keyword is a subset of the course name. For the keyword { probability(TE) principle(NN)} of the topic $t_{3,BUU}$, it matches with { probability(TE) theory(TE)} of the topic $t_{2,KMUTT}$ by two cases : *i*) the word 'probability(TE)'

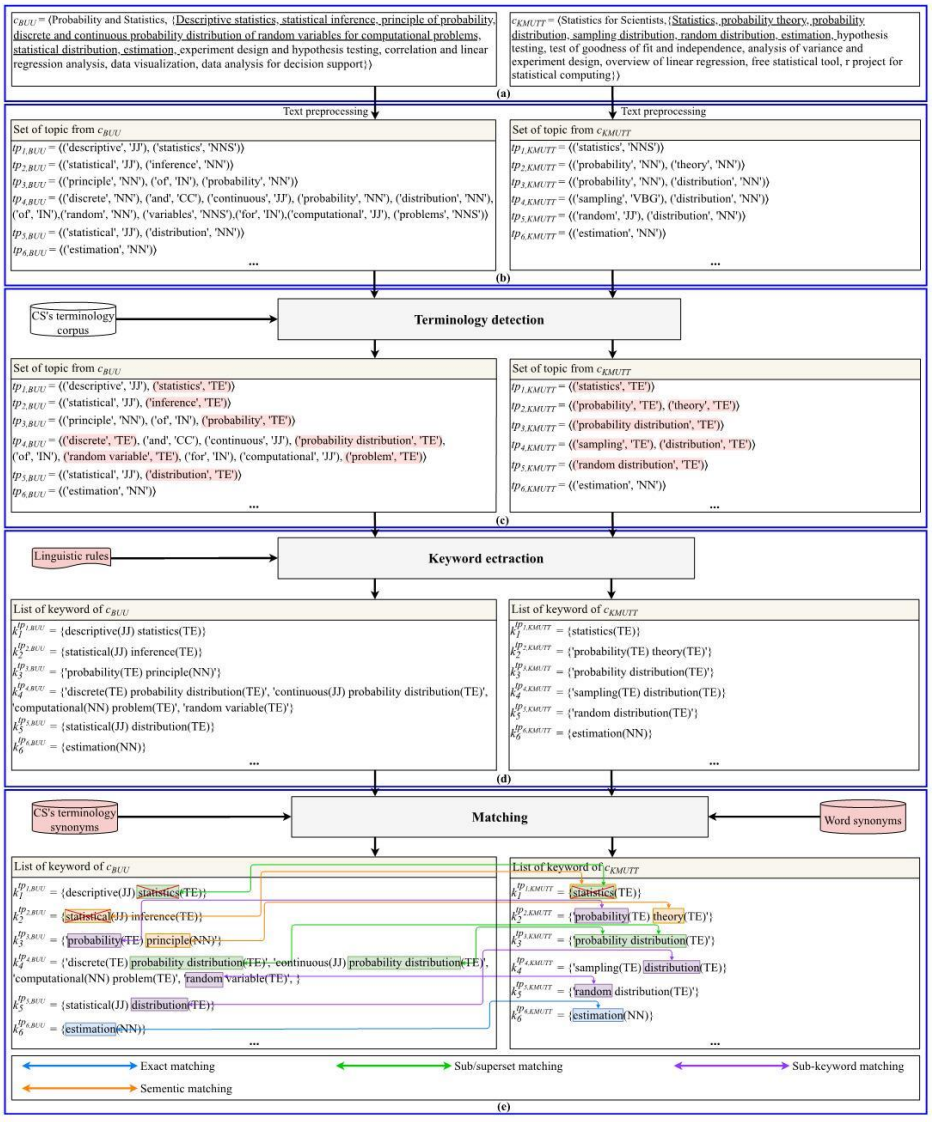


Fig. 2: Example of *eCSCDA* system on considering contents of "Probability and statistics" from BUU and KMUTT

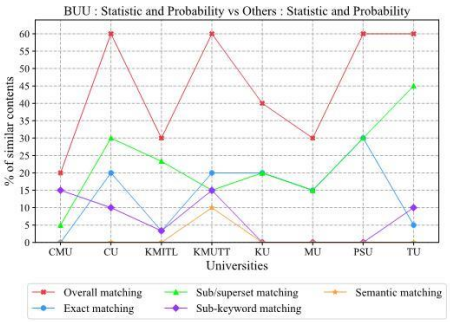


Fig. 3: Percentage of similar contents of *eCSCDA* on "Probability and Statistics" course of BUU against that of others

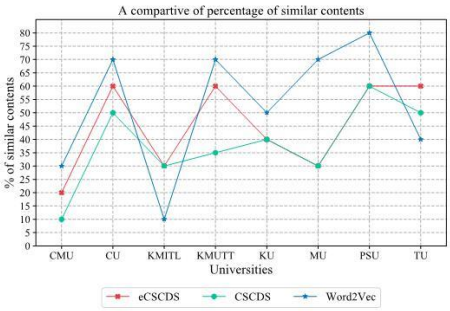


Fig. 4: Percentage of similar contents of *eCSCDA* against *CSCDA*, and *Word2Vec* on "Probability and Statistics"

matched by sub-keyword matching and *ii* the word 'principle(NN)' matched with 'theory(TE)' by semantic matching. With these matches, it can be concluded that the topic $t_{3,BUU}$ and $t_{2,KMUTT}$ is similar. Last, after matching all topics, the percentage of similar contents of BUU and that of KMUTT is calculated as the number of labeled topics of BUU divided by total number topic of BUU, computed as $\frac{4}{10} = 0.4$ (40%). On the other hand, the percentage of similar contents of KMUTT and BUU is the number of labeled topics of KMUTT divided by total number topic of KMUTT, calculated as $\frac{5}{12} = 0.417$ (41.7%), respectively.

IV. Experimental results

Experiments were conducted on 550 CS course description collected (only English part) from 9 Thai universities having CS curriculum (*i.e.* Burapha University (BUU) 66 courses, Chiang Mai University (CMU) 63 courses, Chulalongkorn University (CU) 47 courses, King Mongkut's Institute of Technology Ladkrabang (KMITL) 87 courses, King Mongkut's University of Technology Thonburi (KMUTT) 40 courses, Kasetsart University (KU) 69 courses, Mahidol University (MU) 35 courses, Prince of Songkla University (PSU) 60 courses, and Thammasat University (TU) 85 courses).

TABLE I: Percentage of similar contents of *ecscda* against *cscda*, and *word2vec* on all courses of BUU

BUU(46) vs.	Percentage of similar contents		
	eCSCDA	*CSCDA*	*Word2Vec*
CMU (15)	**34.82**	31.09	41.28
CU (16)	**28.45**	21.97	48.67
KMITL (19)	**32.10**	25.27	48.10
KMUTT (16)	**43.06**	34.87	46.91
KU (20)	**37.94**	27.94	47.94
MU (11)	**32.27**	34.86	47.46
PSU (22)	**35.38**	26.68	51.41
TU (21)	**32.27**	23.35	38.44
Avg	**34.54**	28.25	46.28

In the experiments, one teaching description must be assigned as an *initial course description* and another one (or a group of ones) is set to be a *comparable course description*. Thus, a teaching description of a course from BUU is regarded as *an initial course description* and then compared with ones (on the same course) belonging to other universities. Four measures are applied to investigate the efficiency of the *eCSCDA* system in the term of number of detection and accuracy of matching similar/dissimilar contents defined as *i*) percentage of similar contents (*per_sim* calculated as in Eq. 1), *ii*) $precision = \frac{TP}{TP+FP}$, *iii*) $recall = \frac{TP}{TP+FN}$ and *iv*) $F\text{-}measure = 2 \times \frac{precision \times recall}{precision + recall}$, where TP is the number of correct matching on topics of descriptions given by the system, FP is the number of wrong matching, and FN is the number of mismatching. Last, a comparative study is conducted by comparing the *eCSCDA* system with other related systems *i.e. CSCDA* and *Word2Vec*, respectively.

Fig. 3 shows the percentage of similar contents (*per_sim*) of the "Probability and Statistics" course of BUU in comparison with that of other universities. As shown in the red line, the value of *per_sim* of comparing the course of BUU with another one is between 20 and 60% of the total number of contents of BUU (and $\approx$ 45% on average). This can be inferred that all universities have different perspectives and focuses on the teaching contents. Moreover, the figure also indicates the level of similar contents by the value of *per_sim* identified by each matching technique. With this, there are some contents that are described by using the same words (as shown in the blue line, the *per_sim* identified by exact matching which is $\approx$ 14 on average). Meanwhile, the most similar contents are just related to each other as shown by the value of *per_sim* recognized by subset, sub-keyword, and semantic matching. The average of the summation of *per_sim* identified by these three matching techniques is $\approx$ 31% of $\approx$ 45%. This can let us know that the teaching contents are pretty the same but it might be different on writing style or else which can be further analyzed. Meanwhile, Fig. 4 illustrates the comparison of the percentage of similar contents calculated from *eCSCDA*,

TABLE II: Precision, recall, and f-measure of *ecscda* against *cscda,* and *word2vec* on all courses of BUU

BUU(46) vs.	Precision			Recall			F-measure		
	eCDCDA	*CSCDA*	*Word2Vec*	*eCDCDA*	*CSCDA*	*Word2Vec*	*eCDCDA*	*CSCDA*	*Word2Vec*
CMU (15)	**0.97**	0.97	0.38	**1.00**	0.98	0.46	**0.98**	0.96	0.36
CU (16)	**0.80**	0.72	0.38	**0.83**	0.79	0.56	**0.81**	0.71	0.41
KMITL (19)	**0.82**	0.80	0.45	**0.91**	0.77	0.57	**0.85**	0.78	0.46
KMUTT (16)	**0.88**	0.78	0.33	**0.95**	0.83	0.40	**0.91**	0.79	0.33
KU (20)	**0.91**	0.85	0.32	**0.95**	0.88	0.51	**0.93**	0.86	0.35
MU (11)	**0.86**	0.65	0.40	**0.94**	0.91	0.40	**0.89**	0.75	0.36
PSU (22)	**0.95**	0.87	0.41	**1.00**	0.98	0.53	**0.97**	0.91	0.44
TU (21)	**0.81**	0.77	0.51	**0.85**	0.79	0.50	**0.83**	0.77	0.47
Avg	**0.88**	0.80	0.40	**0.93**	0.87	0.49	**0.90**	0.82	0.40

CSCDA and *Word2Vec*, respectively. It is shown that for some cases *eCSCDA* has the same percentage of similar contents as *CSCDA* but there are some that *eCSCDA* can give higher. The reason is that in some cases the teaching contents are similar only by using the same words or being subset (or superset) of each others. On the other hand, it is also shown that *eCSCDA* is better than *Word2Vec* in some cases but it gives higher precision, recall and F-measure for all cases.

Next, Table I shows a comparative study on the percentage of similar contents calculated by the three methods. From 66 courses from BUU, there are only 46 courses teaches by other universities where 15 of 46 are identical with CMU, 16 with CU, 19 with KMITL, 16 with KMUTT, 20 with KU, 11 with MU, 22 with PSU and 21 with TU, respectively. When looking at the results, it can be seen that teaching courses of BUU are mostly similar to that of KMUTT ($\approx 43\%$ on average) and then follow by MU ($\approx 37\%$), PSU ($\approx 34\%$), and so on. On the other hand, BUU has least similar contents to CU ($\approx 28\%$) which can let us know that both universities have a lot of different contents. It is then can be deeply analyzed for causation of these differences such as different focuses and/or writing styles, lack of updates, etc. Moreover, with the looking at efficiency, it can be seen that our *eCSCDA* can give a higher percentage of similar contents than *CSCDA* $\approx 6\%$ on average but it can give less than *Word2Vec* $\approx 12\%$ on average. This can express that *eCSCDA* outperforms *CSCDA*. However, even though *eCSCDA* give less number of similar contents *Word2Vec* than but its can give the highest values on precision, recall, and F-measure in comparison with the others. With all the results, it can be concluded that *eCSCDA* can efficiently match similar contents hidden in the course descriptions. Thanks to the terminology and word synonyms corpus with the new matching techniques that can give more similar contents.

V. Conclusion

In this paper, a new system, called *eCSCDA* (*efficient Computer Science Course Description Analysis* system) is introduced to improve the performance of *CSCDA* system for analyzing the course descriptions of Computer Science courses. In the new system, new linguistic rules and an efficient keyword extraction technique are applied to precisely identify important contents (*i.e.* keywords and/or terminologies) from a course description. Moreover, two corpuses, terminology and word synonyms, are settled and collected. Then, two matching methods, semantic and sub-keyword matching, based on the new corpuses are designed and applied to improve the task of matching similar contents occurring in course descriptions. From the experiments on 550 Computer Science course descriptions, the results show that the new improved *eCSCDA* outperforms the previous related systems (*i.e. CSCDA* and *Word2Vec*) in the terms of precision, recall, F-measure and percentage of similar content matching, respectively.

References

[1] C. Nuntawong, C. S. Namahoot, and M. Brückner, "A semantic similarity assessment tool for computer science subjects using extended wu & palmer's algorithm and ontology," in *Information Science and Applications*, 2015, pp. 989–996.

[2] C. S. N. Chayan Nuntawong and M. Brückner, "Home: Hybrid ontology mapping evaluation tool for computer science curricula," *Journal of Telecommunication, Electronic and Computer Engineering*, vol. 9, no. 2-3, pp. 61 – 65, 2017.

[3] C. Nuntawong, C. S. Namahoot, and M. Brückner, "A web based cooperation tool for evaluating standardized curricula using ontology mapping," in *Cooperative Design, Visualization, and Engineering*, 2016, pp. 172–180.

[4] C. W. Starr, B. Manaris, and R. H. Stalvey, "Bloom's taxonomy revisited: Specifying assessable learning objectives in computer science," *SIGCSE Bull.*, vol. 40, no. 1, p. 261–265, 2008.

[5] S. Masapanta-Carrión and J. A. Velázquez-Iturbide, "A systematic review of the use of bloom's taxonomy in computer science education," in *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, 2018, p. 441–446.

[6] A. Pawar and V. Mago, "Similarity between learning outcomes from course objectives using semantic analysis, blooms taxonomy and corpus statistics," *ArXiv*, vol. abs/1804.06333, 2018.

[7] G. O. M. O. N. P. V. Saquicela, F. Baculima and M. Espinoza, "Similarity detection among academic contents through semantic technologies and text mining," in *Proceedings INFOBAE Cuba*, 2018, pp. 1–12.

[8] P. Kamlangpuech and K. Amphawan, "A new system for analyzing contents of computer science courses," in *2020 7th International Conference on Advance Informatics: Concepts, Theory and Applications (ICAICTA)*, 2020, pp. 1–6.

[9] X. Rong, "word2vec parameter learning explained," 2016.

[10] J. Wang and Y. Dong, "Measurement of text similarity: a survey," *Information*, vol. 11, no. 9, p. 421, 2020.

[11] B. Chaisoongnoen, K. Amphawan, and A. Bunpeng, "Supplementary book suggestion for computer science courses," in *2018 5th International Conference on Advanced Informatics: Concept Theory and Applications (ICAICTA)*, 2018, pp. 84–90.

[12] B. Chaisoongnoen and K. Amphawan, "An improvement of supplementary book suggestion system," in *The 9th International Conference on Smart Media and Applications (SMA 2020)*, 2020.

บรรณานุกรม

บรรณานุกรม

Apatu, E., Sinnott, W., Piggott, T., Butler-Jones, D., Anderson, L., Alvarez, E., . . . Neil-Sztramko, S. (2020). A content analysis of Canadian master of public health course descriptions and core competencies. *European Journal of Public Health,* 30(Supplement_5), ckaa166. 627.

Balakrishnan, V., & Lloyd-Yemoh, E. (2014). Stemming and lemmatization: a comparison of retrieval performances.

Barrón-Cedeno, A., Rosso, P., Agirre, E., & Labaka, G. (2010). *Plagiarism detection across distant language pairs.* Paper presented at the Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010).

Brew, C., & McKelvie, D. (1996). *Word-pair extraction for lexicography.* Paper presented at the Proceedings of the 2nd international conference on new methods in language processing.

Chaisoongnoen, B., & Amphawan, K. (2020). An improvement of supplementary book suggestion system.

Chaisoongnoen, B., Amphawan, K., & Bunpeng, A. (2018). *Supplementary book suggestion for computer science courses.* Paper presented at the 2018 5th International Conference on Advanced Informatics: Concept Theory and Applications (ICAICTA).

Chung, H., & Kim, J. (2016). *A new personalized syllabus model based on achievement standards analysis and evaluation.* Paper presented at the Proceedings of the World Congress on Engineering and Computer Science 2016.

Corley, C. D., & Mihalcea, R. (2005). *Measuring the semantic similarity of texts.* Paper presented at the Proceedings of the ACL workshop on empirical modeling of semantic equivalence and entailment.

Cross, V., Mokrenko, V., Crockett, K., & Adel, N. (2020). *Using fuzzy set similarity in sentence similarity measures.* Paper presented at the 2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE).

Dai, Y., Asano, Y., & Yoshikawa, M. (2016). Course Content Analysis: An Initiative Step toward Learning Object Recommendation Systems for MOOC Learners. *International Educational Data Mining Society*.

Das, S., Chong, E. I., Eadon, G., & Srinivasan, J. (2004). *Supporting ontology-based semantic matching in RDBMS.* Paper presented at the Proceedings of the Thirtieth international conference on Very large data bases-Volume 30.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:*1810.04805.

Dice, L. R. (1945). Measures of the amount of ecologic association between species. *Ecology,* 26(3), 297-302.

Farouk, M. (2020). Measuring text similarity based on structure and word embedding. *Cognitive Systems Research,* 63, 1-10.

Frantzi, K., Ananiadou, S., & Mima, H. (2000). Automatic recognition of multi-word terms:. the c-value/nc-value method. *International journal on digital libraries,* 3(2), 115-130.

Gali, N., Mariescu-Istodor, R., & Fränti, P. (2016). *Similarity measures for title matching.* Paper presented at the 2016 23rd International Conference on Pattern Recognition (ICPR).

Gomaa, W. H., & Fahmy, A. A. (2012). Short answer grading using string similarity and corpus-based similarity. *International Journal of Advanced Computer Science and Applications (IJACSA),* 3(11).

Gomaa, W. H., & Fahmy, A. A. (2013). A survey of text similarity approaches. *international journal of Computer Applications,* 68(13), 13-18.

Guo, J., Fan, Y., Ai, Q., & Croft, W. B. (2016). *Semantic matching by non-linear word transportation for information retrieval.* Paper presented at the Proceedings of the 25th ACM International on Conference on Information and Knowledge Management.

Gupta, S. (2015). A correction model for real-word errors. *Procedia Computer Science,* 70, 99-106.

Hall, P. A., & Dowling, G. R. (1980). Approximate string matching. *ACM computing surveys (CSUR),* 12(4), 381-402.

Homa, N., Hackathorn, J., Brown, C. M., Garczynski, A., Solomon, E. D., Tennial, R., . . . Gurung, R. A. (2013). An analysis of learning objectives and content coverage in introductory psychology syllabi. *Teaching of Psychology,* 40(3), 169-174.

Islam, A., Milios, E., & Kešelj, V. (2012). *Text similarity using google tri-grams.* Paper presented at the Canadian Conference on Artificial Intelligence.

Jaro, M. A. (1989). Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida. *Journal of the American Statistical Association,* 84(406), 414-420.

Jaro, M. A. (1995). Probabilistic linkage of large public health data files. *Statistics in medicine,* 14(5-7), 491-498.

Lenz, M., Ollinger, S., Sahitaj, P., & Bergmann, R. (2019). *Semantic textual similarity measures for case-based retrieval of argument graphs.* Paper presented at the International Conference on Case-Based Reasoning.

Liu, Z., Xiong, C., Sun, M., & Liu, Z. (2018). Entity-duet neural ranking: Understanding the role of knowledge graph semantics in neural information retrieval. *arXiv preprint arXiv:*1805.07591.

Lopez-Gazpio, I., Maritxalar, M., Lapata, M., & Agirre, E. (2019). Word n-gram attention models for sentence similarity and inference. *Expert Systems with Applications,* 132, 1-11.

Lukyamuzi, A., Ngubiri, J., & Okori, W. (2020). Polarity and Similarity Measures Towards Classifying an Article on Food Insecurity. *International Journal of Technology and Management,* 5(2), 1-10.

Maher, K., & Joshi, M. S. (2016). Effectiveness of different similarity measures for text classification and clustering. *IJCSIT,* 7, 1715-1720.

Metzler, D., Dumais, S., & Meek, C. (2007). *Similarity measures for short segments of text.* Paper presented at the European conference on information retrieval.

Mihalcea, R., Corley, C., & Strapparava, C. (2006). *Corpus-based and knowledge-based measures of text semantic similarity.* Paper presented at the Aaai.

Miller, G. A. (1995). WordNet: a lexical database for English. *Communications of the ACM,* 38(11), 39-41.

Mohammed, D. A., & Kadhim, N. J. (2020). Extractive Multi-Document Summarization Model Based On Different Integrations of Double Similarity Measures. *Iraqi Journal of Science*, 1498-1511.

Mumtaz, S., & Giese, M. (2020). *Frequency-Based vs. Knowledge-Based Similarity Measures for Categorical Data.* Paper presented at the AAAI Spring Symposium: Combining Machine Learning with Knowledge Engineering (1).

Nuntawong, C., Namahoot, C. S., & Brückner, M. (2017). Home: Hybrid ontology mapping evaluation tool for computer science curricula. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC),* 9(2-3), 61-65.

Qurashi, A. W., Holmes, V., & Johnson, A. P. (2020). *Document Processing: Methods for Semantic Text Similarity Analysis.* Paper presented at the 2020 International Conference on INnovations in Intelligent SysTems and Applications (INISTA).

Rose, S., Engel, D., Cramer, N., & Cowley, W. (2010). Automatic keyword extraction from individual documents. *Text mining: applications and theory,* 1, 1-20.

Shamsi, J. A., ul Hassan, S. Z., Bawany, N., & Shoaib, N. (2018). *A Comprehensive Course on Big Data for Undergraduate Students.* Paper presented at the 2018 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW).

Singh, R., & Singh, S. (2021). Text Similarity Measures in News Articles by Vector Space Model Using NLP. *Journal of The Institution of Engineers (India): Series B,* 102(2), 329-338.

Sitikhu, P., Pahi, K., Thapa, P., & Shakya, S. (2019). *A comparison of semantic similarity methods for maximum human interpretability.* Paper presented at the 2019 artificial intelligence for transforming business and society (AITB).

Starr, C. W., Manaris, B., & Stalvey, R. H. (2008). Bloom's taxonomy revisited: specifying assessable learning objectives in computer science. *ACM SIGCSE Bulletin,* 40(1), 261-265.

Tessem, B. (2019). *Analogical news angles from text similarity.* Paper presented at the International Conference on Innovative Techniques and Applications of Artificial Intelligence.

Tharwat, A. (2020). Classification assessment methods. *Applied Computing and Informatics*.

Toutanova, K., Klein, D., Manning, C. D., & Singer, Y. (2003). *Feature-rich part-of-speech tagging with a cyclic dependency network.* Paper presented at the Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics.

Vijaymeena, M., & Kavitha, K. (2016). A survey on similarity measures in text mining. *Machine Learning and Applications: An International Journal,* 3(2), 19-28.

Wang, J., & Dong, Y. (2020). Measurement of text similarity: a survey. *Information,* 11(9), 421.

Wang, S., & Jiang, J. (2016). A compare-aggregate model for matching text sequences. *arXiv preprint arXiv:*1611.01747.

Xu, J., & Xu, G. (2011). *Learning similarity function for rare queries.* Paper presented at the Proceedings of the fourth ACM international conference on Web search and data mining.

Zhang, Y., Baldridge, J., & He, L. (2019). PAWS: Paraphrase adversaries from word scrambling. *arXiv preprint arXiv:*1904.01130.

ประวัติย่อของผู้วิจัย

ชื่อ-สกุล	พีระพล กำลังพืช
วัน เดือน ปี เกิด	04 ธันวาคม 2536
สถานที่เกิด	จ.พระนครศรีอยุธยา
สถานที่อยู่ปัจจุบัน	บ้านเลขที่ 113 หมู่ 3 ตำบล บางขันหมาก อำเภอ เมือง จังหวัด ลพบุรี 15000
ตำแหน่งและประวัติการทำงาน	วิทยาศาสตรบัญฑิต สาขาวิทยาการคอมพิวเตอร์ คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา
ประวัติการศึกษา	ประถมศึกษา โรงเรียนกองทัพบกอุปถัมภ์ค่ายนารายณ์ศึกษา มัธยมศึกษาตอนต้น โรงเรียนพระนารายณ์ศึกษา มัธยมศึกษาตอนปลาย โรงเรียนพระนารายณ์ศึกษา ปริญญาตรี วิทยาศาสตรบัญฑิต สาขาวิทยาการคอมพิวเตอร์ คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา