



PROJECT NO. 67

ระบบจัดเก็บและจัดการเอกสารภายในห้องบรรณสารสนเทศ

MR.AKARAPON BOONSERMSAKUL

MS.THANAPORN PITIANUSORN

MR.ANNOP KONGSOMBATCHAROEN

A PROJECT SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENTS FOR

THE DEGREE OF BACHELOR OF ENGINEERING (COMPUTER ENGINEERING)

FACULTY OF ENGINEERING

KING MONGKUT'S UNIVERSITY OF TECHNOLOGY THONBURI

2020

Project No. 67

ระบบจัดเก็บและจัดการเอกสารภายในห้องบรรณสารสนเทศ

Mr.Akarapon Boonsermsakul

Ms.Thanaporn Pitianusorn

Mr.Annop Kongsombatcharoen

A Project Submitted in Partial Fulfillment

of the Requirements for

the Degree of Bachelor of Engineering (Computer Engineering)

Faculty of Engineering

King Mongkut's University of Technology Thonburi

2020

Project Committee

Project Advisor

(Asst.Prof.Suthathip Maneewongvatana, Ph.D.)

Committee Member

(Dr.Prapong Prechaprapranyawong, Ph.D.)

Committee Member

(Asst.Prof.Sanan Srakaew)

Committee Member

(Asst.Prof.Surapont Toomnark)

Project Title	Project No. 67 ระบบจัดเก็บและจัดการเอกสารภายในห้องสมุดสารสนเทศ
Credits	3
Member(s)	Mr.Akarapon Boonsermsakul Ms.Thanaporn Pitianusorn Mr.Annop Kongsombatcharoen
Project Advisor	Asst.Prof.Suthathip Maneewongvatana, Ph.D.
Program	Bachelor of Engineering
Field of Study	Computer Engineering
Department	Computer Engineering
Faculty	Engineering
Academic Year	2020

Abstract

KMUTT's library have collected the archive of valued documents. Because these document have not transformed into digital form, there is vital problem in searching for information in these document for librarian and patrons. In this project, we developed web platform to digitize these document into digital format and implement the search function that facilitate the librarian and patron to search for information. The platform consists of 2 components. The first part is importing documents and digitization. In this step, we applied image processing techniques such as Morphology Transformation to preprocess the images of documents and transform the images to full text data by using Tesseract. After getting the text files, we tokenize the text into words by using the Deepcut library and find the significant words of the document by using the TF-IDF algorithm. In the second part, we start by getting the input from the user and use the word2Vec model to find a similar word. And take input and similar words to get the TF-IDF score that we generate at first to find the best document for the input word. Comparing the result between using OCR and using OCR with correction system, using only OCR have correction score around 74.75 percents and using OCR with correction system have correction score 76.61 percents. And the accuracy and recall of search system without Word2Vec are 75 percents and 88.24 percents accordingly. But after using Word2Vec models the accuracy drop to 61.45 percents while recall is still the same as without Word2Vec.

Keywords: Natural language processing / RESTful Service / Optical character recognition / Image Processing / Information retrieval / Term Frequency-Inverse Document Frequency / Word2Vec / Word Embedded

หัวข้อปริญญาในพินธ์	ระบบจัดเก็บและจัดการเอกสารภายในห้องรับนักสารสนเทศ KMUTT Archives Management Platform
หน่วยกิต	3
ผู้เขียน	นายอัครพล บุญเสริมศักดิ์กุล นางสาวอรอนพร ปิติอนุสรณ์ นายอรรถนพ กองสมบัติเจริญ
อาจารย์ที่ปรึกษา	ผศ.ดร.สุราทิพย์ มนีวงศ์วัฒนา
หลักสูตร	วิศวกรรมศาสตรบัณฑิต
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
ภาควิชา	วิศวกรรมคอมพิวเตอร์
คณะ	วิศวกรรมศาสตร์
ปีการศึกษา	2563

บทคัดย่อ

การจะสืบค้นข้อมูลจากเอกสารหรือขันหนังสือที่มีการรวบรวมข้อมูลไว้ตั้งแต่อดีตจนเป็น ปัจจุบันย่างหนึ่งของเจ้าหน้าที่บรรณาธิการที่ต้องทำการคูณแลเอกสารเหล่านี้ เนื่องจาก การที่ยังไม่มีการเก็บหนังสือและเอกสารให้อยู่ในรูปแบบของข้อมูลดิจิทัลทำให้ต้อง สืบค้นโดยการค้นหาเอกสารและหนังสือแต่ละเล่มโดยการคูณจากเนื้อหาสารบัญเพื่อให้ ได้หนังสือที่ตรงกับข้อมูลที่ต้องการมากที่สุด ซึ่งการที่ค้นหาจากหน้าสารบัญของ หนังสือแต่ละเล่มก็จะทำให้การค้นหาเป็นไปอย่างล่าช้า และบางครั้งการคูณเพียง แค่สารบัญก็อาจจะทำให้ได้หนังสือที่ไม่ตรงกับความต้องการของผู้ที่เข้ามายืนหนังสือ ในโครงการนี้เราได้ทำการพัฒนาการระบบจัดเก็บและค้นหาเอกสารอิเล็กทรอนิกส์ โดยแบ่งออกเป็น 2 ขั้นตอนคือ การนำเข้าข้อมูล และการสร้างระบบค้นหา โดยขั้นตอนการนำเข้าข้อมูล เราจะเริ่มจากการเตรียมข้อมูลรูปภาพ เพื่อเตรียมข้อมูลรูปภาพที่ได้มา ก่อนจะนำไปผ่านกระบวนการ OCR เพื่อแปลงรูปภาพเหล่านี้ให้อยู่ในรูปของข้อมูลดิจิทัล โดยการเก็บข้อมูลในรูปแบบของ Information Retrieval เพื่อช่วยให้ความเร็วการค้นหาใหม่ประสิทธิภาพมากยิ่งขึ้น และนำข้อมูลมาทำการตัดคำ และเช็คคำพิเศษ จากนั้นจะนำมาหาคำสำคัญของหนังสือหรือเอกสารนั้น ๆ โดยการใช้การหาค่าคะแนนแบบ TF-IDF ส่วนการสร้างระบบการค้นหา จะเริ่มจากการรับคำค้นมาจากผู้ใช้และทำการนำคำที่ได้ไปเข้าโน๊ಡล word2Vec เพื่อหาคำที่ใกล้เคียง จากนั้นนำคำใกล้เคียงและคำค้นหาไปดึงคะแนน TF-IDF ที่เก็บไว้เพื่อค้นหาว่า มีเอกสารหรือหนังสือเล่มไหนที่มีคะแนนที่ต้องแรงใกล้เคียงกับคำค้นหากำกที่สุด โดยผลลัพธ์จากการทำ OCR ถูกต้อง 74.75 % และเมื่อนำมาผ่านกระบวนการแก้คำพิเศษได้ความถูกต้องอยู่ที่ 76.61% และมีผลลัพธ์ในการค้นหาโดยที่ไม่ใช้ Word2Vec มีความแม่นยำอยู่ที่ 75 % และความครอบคลุมอยู่ที่ 88.24 % แต่หลังจากใช้งาน Word2Vec มีความแม่นยำอยู่ที่ 61.45 % และความครอบคลุมอยู่ที่ 88.24 %

คำสำคัญ: Natural language processing / RESTful Service / Optical character recognition / Image Processing / Information retrieval / Term Frequency-Inverse Document Frequency / Word2Vec / Word Embedded

กิตติกรรมประกาศ

ขอขอบคุณพศ.ดร.สุชาทิพย์ มณีวงศ์วัฒนา อาจารย์ที่ปรึกษา ผู้ช่วยอาจารย์ที่ปรึกษา(อาจารย์ปริยากร) คณะกรรมการ เจ้าหน้าที่หอบรรณ
สารสนเทศโดยเฉพาะพี่เพรvisor (นางสาวอารยา ศรีบวบาน) และ รวมทั้งเจ้าหน้าที่ภายใน หอสมุดมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี
ที่เสียสละเวลาให้ความรู้ความเข้าใจและคอยช่วยเหลือ ทั้งในเรื่องการเก็บข้อมูลและคอยแนะนำวิธีการจัดการกับ ปัญหาต่างๆ ที่เกิดขึ้น นำ
มาสู่การทำหัวข้อปริญญาในพนธน์ฉบับนี้ให้สำเร็จตามที่ต้องการ

สารบัญ

หน้า

ABSTRACT	ii
บทคัดย่อ	iii
กิตติกรรมประกาศ	iv
สารบัญ	ix
สารบัญตาราง	x
สารบัญรูปภาพ	xii
บทที่ 1 บทนำ	1
1.1 คำสำคัญ	1
1.2 ความสำคัญของปัญหา	1
1.3 ประเภทของโครงงาน	1
1.4 วิธีการที่นำเสนอ	1
1.5 วัตถุประสงค์	2
1.6 ขอบเขตของงานวิจัย	2
1.7 เนื้อหาทางวิศวกรรมที่เป็นต้นฉบับ	2
1.8 การแยกย่อยงาน และร่างแผนการดำเนินงาน	3
1.9 ตารางการดำเนินงาน	4
1.9.1 ผลการดำเนินงานในภาคการศึกษาที่ 1	5
1.9.2 ผลการดำเนินงานในภาคการศึกษาที่ 2	5
บทที่ 2 ที่มา ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	6
2.1 บทนำ	6
2.2 แนวความคิดทางทฤษฎี	6
2.2.1 การเตรียมข้อมูลรูปภาพ	6
2.2.1.1 คอนทัวร์ (Contour)	6

2.2.1.2	การเปลี่ยนแปลงทางสัณฐานวิทยา (Morphology Transformation)	7
2.2.2	Optical Character Recognition (OCR)	7
2.2.3	Natural language processing	8
2.2.3.1	Information retrieval	8
2.2.3.2	TF-IDF	9
2.2.3.3	Cosine Similarity	9
2.2.3.4	Minimum Edit Distance	10
2.2.3.5	Spelling Corrector by Peter Norvig	11
2.2.4	RESTful Service	11
2.2.5	Word Embedding	12
2.2.5.1	Word2Vec	12
2.2.5.2	Skip-gram	13
2.2.5.3	CBOW	14
2.3	ภาษาคอมพิวเตอร์และเทคโนโลยี	14
2.3.1	Open source Computer Vision (OpenCV)	14
2.3.2	Tesseract OCR	14
2.3.3	DeepCut	15
2.3.4	ReactJS	15
2.3.5	Python	15
2.3.5.1	Django	15
2.3.6	NodeJS	15
บทที่ 3	การออกแบบและระบบวิธีจัด	16
3.1	ภาพรวมของระบบ	16
3.2	การออกแบบการทดลอง	16
3.2.1	การแปลงข้อมูลในหนังสือ (Digitization)	16
3.2.2	การเตรียมข้อมูลรูปภาพ	16

3.2.2.1	การคัดเลือกข้อมูล	17
3.2.2.2	การหมุนรูป	17
3.2.2.3	การลบพื้นหลัง	22
3.2.3	การแปลงข้อมูลรูปภาพให้อยู่ในรูปแบบดิจิทัล	28
3.2.4	การเตรียมข้อมูลตัวอักษร	28
3.2.5	การคำนวณค่าความสำคัญของคำตามที่ปรากฏในเอกสาร	28
3.2.5.1	การอัพเดทค่าคะแนน TF-IDF	29
3.2.6	การค้นหา	29
3.2.6.1	การทำโมเดล Word2vec	30
3.2.7	การจัดการหนังสือดิจิทัล	30
3.2.8	Login	31
3.3	System requirements	31
3.4	โครงสร้างฐานข้อมูล	33
3.4.1	Database Structure	37
3.4.2	Database Dictionary	38
3.5	UML Design	39
3.5.1	Use case diagram	39
3.5.2	Sequence diagram	39
3.5.2.1	Use case Add Document	39
3.5.2.2	Use case Manage word in document	40
3.5.2.3	Use case Verify Document to Generate Keyword	41
3.5.2.4	Use case Edit Document	42
3.5.2.5	Use case Delete Document	43
3.5.2.6	Use case View Document & Search Document	44
3.5.2.7	Use case Login	45
3.6	GUI Design	47
3.6.1	Homepage	47

3.6.2	Homepage2	48
3.6.3	Login	49
3.6.4	Insert Book (1)	50
3.6.5	Insert Book (2)	51
3.6.6	Insert Book (3)	52
3.6.7	Insert Book (4)	53
3.6.8	Insert Book (5)	54
3.6.9	Insert Book (6)	55
3.6.10	Search	56
3.6.11	Document View	57
3.6.12	Manage book	58
3.6.13	Edit Book	59
3.6.14	Upload Status Page	62
3.6.15	Evaluate Process Design	62
บทที่ 4	ผลการดำเนินงาน	65
4.1	ผลลัพธ์ที่ได้จากการแปลงข้อมูลรูปภาพให้เป็นข้อมูลดิจิทัล	65
4.1.1	ผลลัพธ์ที่ได้จากการประสิทธิภาพของกราฟิก	65
4.1.2	ผลการเปรียบเทียบประสิทธิภาพในการทำ OCR ของ การทำการเตรียมข้อมูลรูปภาพ แต่ละแบบ	65
4.1.2.1	แบบที่ 1 การใช้การคัดเลือกข้อมูล, การหมุน, การลบรูปภาพ, การลบเส้น และการจัดกลุ่ม	66
4.1.2.2	แบบที่ 2 ใช้การลบพื้นหลัง	67
4.1.3	ผลการเปรียบเทียบข้อมูล training set 2 ชุดสำหรับการทำ OCR	67
4.1.4	ประสิทธิภาพการแก้ไขคำผิด	69
4.2	ผลลัพธ์จากการค้นหา	69
4.2.1	ผลการเปรียบเทียบประสิทธิภาพเวลาในการค้นหา	73
4.3	ผลลัพธ์จากการดำเนินงานในส่วนของการทำเว็บไซต์	73
4.3.1	การประเมินการใช้งานของเว็บไซต์	73

4.3.2	การประเมินความพึงพอใจของบรรณารักษ์ต่อการออกแบบ UX/UI	74
4.3.3	หน้าหลัก	74
4.3.4	การเข้าสู่ระบบเว็บไซต์	75
4.3.5	การเพิ่มหนังสือเข้าสู่ระบบฐานข้อมูล	75
4.3.5.1	เพิ่มข้อมูลของหนังสือ	75
4.3.5.2	การแก้ไขและตรวจสอบคำก่อนนำเข้าสู่ระบบ	77
4.3.5.3	การตรวจสอบแก้ไขค่าสำคัญ	78
4.3.6	การแสดงสถานะการเพิ่มหนังสือ	78
4.3.7	การแสดงการค้นหาหนังสือ	79
4.3.8	การแสดงข้อมูลหนังสือ	79
4.3.9	การแสดงการแก้ไขข้อมูลของหนังสือ	80
บทที่ 5 สรุปผล		82
5.1	ผลการดำเนินงาน	82
5.2	สรุปผลการดำเนินงาน	82
5.3	บัญหาที่พบและการแก้ไข	84
5.3.1	บัญหาน้ำเสียอันยาก	84
5.3.2	บัญหาการหมุนไม่ต่อ	84
5.3.3	บัญหาการแก้ไขคำผิด	84
5.3.4	บัญหาระยะเวลาในการเพิ่มข้อมูลหนังสือ	84
5.3.5	บัญหาของการหาคำใหม่่อนของโมเดล Word2Vec	85
5.4	ข้อจำกัดและข้อเสนอแนะ	85
หนังสืออ้างอิง		86
APPENDIX		87
A	Database Dictionary	88

สารบัญตาราง

ตารางที่	หน้า
1.1 ตารางการดำเนินงาน ภาคการศึกษาที่ 1/2563	4
1.2 ตารางการดำเนินงาน ภาคการศึกษาที่ 2/2563	5
2.1 Information retrieval ในลักษณะ Boolean Retrieval	8
3.1 ตารางประเมินการทำ OCR	62
3.2 ตารางประเมินระบบการค้นหา	63
3.3 ตารางประเมินความพึงพอใจการออกแบบ UX/UI	64
3.4 ตารางประเมินการทดสอบเว็บไซต์	64
4.1 ตารางประเมินการทำการเตรียมข้อมูลรูปภาพแบบที่ 1	66
4.2 ตารางประเมินการทำการเตรียมข้อมูลรูปภาพแบบที่ 2	67
4.3 ตารางประเมินข้อมูลชุด training set ที่ 1	68
4.4 ตารางประเมินข้อมูลชุด training set ที่ 2	68
4.5 ตารางประเมินข้อมูลชุด training set ที่ 1 ที่ไม่ผ่านการแก้ไขคำผิด	69
4.6 ตารางแสดงการทำ Confusion matrix	69
4.7 ตารางแสดงรายละเอียด Confusion matrix	70
4.8 ตารางแสดงผลการค้นหาจากชุดข้อมูล 44 เล่มที่ไม่ผ่านการแก้ไขคำผิดจากมนุษย์	70
4.9 ตารางแสดงผลการค้นหาจากชุดข้อมูล 6 เล่มที่ไม่ผ่านการแก้ไขคำผิดจากมนุษย์	70
4.10 ตารางแสดงผลการค้นหาจากชุดข้อมูล 6 เล่มที่ผ่านการแก้ไขคำผิดจากมนุษย์	70
4.11 ตารางแสดงผลการค้นหาจากชุดข้อมูล 6 เล่มที่ไม่ผ่านการแก้ไขคำผิดจากมนุษย์แบบไม่มีคำเฉพาะ	72
4.12 ตารางแสดงผลการค้นหาจากชุดข้อมูล 6 เล่มที่ไม่ผ่านการแก้ไขคำผิดจากมนุษย์แบบไม่มีคำเฉพาะ	73
4.13 ตารางแสดงผลการประเมินการทำเว็บไซต์	73
4.14 ตารางประเมินความพึงพอใจการออกแบบ UX/UI	74
5.1 ตารางสรุปผลลัพธ์การทำนิยงาน	82
A.2 ตารางอธิบายความหมายตาราง term_word	88
A.3 ตารางอธิบายความหมายตาราง user	88

A.4	ตารางอธิบายความหมายตาราง score	89
A.5	ตารางอธิบายความหมายตาราง pre_term_in_page	89
A.6	ตารางอธิบายความหมายตาราง page_in_document	89
A.7	ตารางอธิบายความหมายตาราง nodejs_log	90
A.8	ตารางอธิบายความหมายตาราง knex_migrations_lock	90
A.9	ตารางอธิบายความหมายตาราง knex_migrations	90
A.10	ตารางอธิบายความหมายตาราง indexing_publisher_document	91
A.11	ตารางอธิบายความหมายตาราง indexing_publisher_email_document	91
A.12	ตารางอธิบายความหมายตาราง indexing_issued_date_document	91
A.13	ตารางอธิบายความหมายตาราง indexing_creator_orgname_document	91
A.14	ตารางอธิบายความหมายตาราง indexing_creator_document	92
A.15	ตารางอธิบายความหมายตาราง dc_contributors	92
A.16	ตารางอธิบายความหมายตาราง indexing_contributor_document	92
A.17	ตารางอธิบายความหมายตาราง indexing_contributor_role_document	92
A.18	ตารางอธิบายความหมายตาราง dc_type	93
A.19	ตารางอธิบายความหมายตาราง dc_relation	93
A.20	ตารางอธิบายความหมายตาราง dc_keyword	93
A.21	ตารางอธิบายความหมายตาราง document	93

สารบัญรูป

รูปที่		หน้า
2.1	แสดงการหาค่าโครงร่างภายในรูป	6
2.2	แสดงการทำการขยายภาพ (Dilation) เพื่อเพิ่มพื้นที่สีขาว	7
2.3	แสดงการทำกร่อนภาพ (Erosion) เพื่อกร่อนพื้นที่สีขาว	7
2.4	Information retrieval ในลักษณะ Index Retrieval	9
2.5	หลักการการเช็ค edit distance [1]	10
2.6	ตัวอย่างตารางการทำ Minimum edit distance [1]	10
2.7	แสดงถึงโครงสร้างของ HTTP Request [2]	11
2.8	แสดงถึงโครงสร้างของ HTTP Response [2]	12
2.9	ภาพแสดงโมเดล Skip-gram และ CBOW [3]	13
2.10	แสดงการทำงานของ Skip-gram [3]	13
2.11	แสดงการทำางานของ CBOW [3]	14
3.1	ภาพรวมของระบบ	16
3.2	ภาพแสดงความถี่ของภาพพื้นหลังสีและภาพพื้นหลังขาวดำ	17
3.3	ภาพแสดงขั้นตอนการคัดเลือกข้อมูล	17
3.4	ภาพแสดงการทำกร่อนภาพ (Erosion) และการทำขยายภาพ (Dilation)	18
3.5	ภาพแสดงการเปรียบเทียบการทำกร่อนภาพ (Erosion) และการทำขยายภาพ (Dilation)	18
3.6	ภาพแสดงเกณฑ์การวัดบรรทัดของตัวอักษร	18
3.7	ภาพแสดงการคัดแยกองตัว (Contour) ที่ไม่ใช่ตัวอักษร	19
3.8	ภาพแสดงการทำ Mask ในส่วนที่ไม่ใช่ตัวอักษร	19
3.9	ภาพแสดงการคัดตัวอักษรเพื่อนำไปหางานในการหมุน	19
3.10	ภาพแสดงจุดของคอนทัวร์ (Contour) เล็กในคอนทัวร์ (Contour) ใหญ่	20
3.11	ภาพแสดงฟังก์ชันการลับรูปภาพออกจากหนังสือ	20
3.12	ภาพแสดงการสร้าง Mask เพื่อลบรูปภาพ	20
3.13	ภาพแสดงการสร้าง Mask โดยเว้นที่ตัวอักษร	21

3.14 ภาพแสดงการหาองศาในการหมุน	21
3.15 ภาพแสดงผลลัพธ์ที่ได้จากการหมุน	22
3.16 ภาพแสดงขั้นตอนในการลบพื้นหลังสี	22
3.17 รูปภาพสีก่อนถูกนำเข้ามาทำการลบพื้นหลัง	23
3.18 รูปภาพการแปลงภาพสีเป็น gray scale	23
3.19 รูปภาพที่ผ่านการทำกรวยขยาย โดยใช้รูปแบบสี่เหลี่ยมขนาด 5x5	24
3.20 รูปภาพที่ผ่านการลบพื้นหลัง	24
3.21 รูปภาพที่ผ่านทำการ threshold แบบ THRESH_BINARY_INV	25
3.22 รูปภาพแสดงการใช้การลบพื้นหลังด้วย kernel ขนาด 3*3	25
3.23 รูปภาพแสดงการใช้การลบพื้นหลังด้วย kernel ขนาด 5*5	26
3.24 รูปภาพแสดงการใช้การลบพื้นหลังด้วย kernel ขนาด 7*7	26
3.25 รูปภาพแสดงการใช้การลบพื้นหลังด้วย kernel ขนาด 15*15	26
3.26 รูปภาพแสดงการใช้การลบพื้นหลังด้วย kernel ขนาด 25*25	27
3.27 รูปภาพแสดงการใช้การลบพื้นหลังด้วย kernel ขนาด 15*15 ที่กินพื้นที่รอบข้าง	27
3.28 รูปภาพแสดงการคำนวณ Cosine Similarity	30
3.29 แสดง ER Diagram ของฐานข้อมูล	33
3.30 แสดง ER Diagram ส่วนของคีย์เวิร์ดและคะแนนความสำคัญในระบบ	34
3.31 แสดง ER Diagram ส่วนของการเก็บคำจากแต่ละหน้าที่แปลงมาจากหนังสือ	34
3.32 แสดง ER Diagram ส่วนของประวัติของผู้ใช้งานมีการสร้างหรือแก้ไขหนังสือ	34
3.33 แสดง ER Diagram ส่วนของการเก็บข้อมูล keyword, relation, type ของหนังสือ	34
3.34 แสดง ER Diagram ส่วนของการเก็บข้อมูล Contributors ว่ามีความเกี่ยวข้องกับหนังสือในรูปแบบใดบ้าง	35
3.35 แสดง ER Diagram ส่วนของ Creator มีความเกี่ยวข้องกับหนังสือในรูปแบบใดบ้าง	35
3.36 แสดง ER Diagram ส่วนของ Creator Organized Name มีความเกี่ยวข้องกับหนังสือในรูปแบบใดบ้าง	35
3.37 แสดง ER Diagram ส่วนของ Publisher มีความเกี่ยวข้องกับหนังสือในรูปแบบใดบ้าง	35
3.38 แสดง ER Diagram ส่วนของ Publisher Email มีความเกี่ยวข้องกับหนังสือในรูปแบบใดบ้าง	36

3.39 แสดง ER Diagram ส่วนของ Issued Date มีความเกี่ยวข้องกับหนังสือไหนบ้าง	36
3.40 แสดง ER Diagram ส่วนของ Knex module ที่ใช้สำหรับ Migration ฐานข้อมูล	36
3.41 แสดง ER Diagram ส่วนของการเก็บประวัติการ HTTP Request NodeJS ไปยัง Django	37
3.42 Use case diagram	39
3.43 แสดง Scenario 1 เพิ่มหนังสือเข้าระบบ	40
3.44 แสดง Scenario 2 การจัดการคำที่ถูกเก็บได้จากหนังสือในระบบ	41
3.45 แสดง Scenario 3 ยืนยันว่าพร้อมสำหรับการถูกนำไปสร้างคีย์เวิร์ด	42
3.46 แสดง Scenario 4 แก้ไขข้อมูลหนังสือ	43
3.47 แสดง Scenario 5 ลบหนังสือ	44
3.48 แสดง Scenario 6 ดูข้อมูลหนังสือ และการค้นหาหนังสือ	45
3.49 แสดง Scenario 7 ระบบล็อกอิน	46
3.50 ภาพแสดงหน้าหลักของเว็บไซต์	47
3.51 ภาพแสดงหน้าหลักของเว็บไซต์หลังจากการกดเปิดเมนู	48
3.52 ภาพแสดงหน้าเข้าสู่ระบบ	49
3.53 ภาพแสดงขั้นตอนการเพิ่มหนังสือเข้าสู่ระบบขั้นเลือกไฟล์	50
3.54 ภาพแสดงขั้นตอนการเพิ่มหนังสือเข้าสู่ระบบขั้นกรอกข้อมูลขั้นที่ 1	51
3.55 ภาพแสดงขั้นตอนการเพิ่มหนังสือเข้าสู่ระบบขั้นกรอกข้อมูลขั้นที่ 2	52
3.56 ภาพแสดงขั้นตอนการเพิ่มหนังสือเข้าขั้นโหลดข้อมูลเข้าสู่ระบบ	53
3.57 ภาพแสดงขั้นตอนการเพิ่มหนังสือเข้าสู่ระบบขั้นแก้ไขคำพิเศษ	54
3.58 ภาพแสดงขั้นตอนการเพิ่มหนังสือเข้าสู่ระบบขั้นแก้ไขและเพิ่มคำสำคัญ	55
3.59 ภาพแสดงหน้าค้นหาข้อมูล	56
3.60 ภาพแสดงหน้าดูหนังสือ	57
3.61 ภาพแสดงหน้าการจัดการหนังสือที่เพิ่มเข้าสู่ระบบ	58
3.62 ภาพแสดงขั้นตอนการแก้ไขหนังสือขั้นที่ 1	59
3.63 ภาพแสดงขั้นตอนการแก้ไขหนังสือขั้นที่ 2	60
3.64 ภาพแสดงขั้นตอนการแก้ไขหนังสือขั้นที่ 3	61

3.65	ภาพแสดงหน้าการโหลดข้อมูล	62
4.1	ภาพแสดงผลลัพธ์การทวนรูปที่ผิดพลาด	65
4.2	ภาพแสดงผลการเปรียบเทียบการใช้โมเดลที่ผ่านการแก้ไขคำผิด และไม่ผ่านการแก้ไขคำผิด	71
4.3	ภาพแสดงค่าแนวการค้นหาคำใหม่จากโมเดล word2vec	71
4.4	ภาพแสดงผลการเปรียบเทียบการใช้ word2vec และไม่ใช้ word2vec	72
4.5	ภาพแสดงหน้าเว็บหลัก	74
4.6	ภาพแสดงหน้าเข้าสู่ระบบ	75
4.7	ภาพแสดงขั้นตอนการเพิ่มหนังสือขั้นตอนการเพิ่มไฟล์	75
4.8	ภาพแสดงขั้นตอนการเพิ่มหนังสือเข้าสู่ระบบขั้นกรอกข้อมูลขั้นที่ 1	76
4.9	ภาพแสดงขั้นตอนการเพิ่มหนังสือเข้าสู่ระบบขั้นกรอกข้อมูลขั้นที่ 2	76
4.10	ภาพแสดงขั้นตอนการเพิ่มหนังสือเข้าสู่ระบบขั้นการเตรียมข้อมูล	76
4.11	ภาพแสดงขั้นตอนการเพิ่มหนังสือเข้าสู่ระบบขั้นการแก้ไขคำผิด	77
4.12	ภาพแสดงหน้าต่างยืนยันการแก้คำ	77
4.13	ภาพแสดงขั้นตอนการเพิ่มหนังสือเข้าสู่ระบบขั้นการสร้างคำสำคัญ	77
4.14	ภาพแสดงขั้นตอนการเพิ่มหนังสือเข้าสู่ระบบขั้นการแก้ไขคำสำคัญ	78
4.15	ภาพแสดงสถานะของการเพิ่มข้อมูลเข้าสู่ระบบ	78
4.16	ภาพแสดงหน้าการค้นหา	79
4.17	ภาพแสดงหน้าการแสดงหนังสือ	79
4.18	ภาพแสดงข้อมูลของหนังสือ	80
4.19	ภาพแสดงหน้าการค้นหาในหน้าการจัดการหนังสือ	80
4.20	ภาพแสดงหน้าการลบหนังสือ	80
4.21	ภาพแสดงหน้าการแก้ไขข้อมูลขั้นที่ 1	81
4.22	ภาพแสดงหน้าการแก้ไขข้อมูลขั้นที่ 2	81
4.23	ภาพแสดงหน้าการแก้ไขคำสำคัญ	81

บทที่ 1 บทนำ

1.1 คำสำคัญ

Natural language processing, RESTful Service, Optical character recognition, Image Processing, Information retrieval, Term Frequency-Inverse Document Frequency, Word2Vec, Word Embedded

1.2 ความสำคัญของปัญหา

นับตั้งแต่การก่อตั้งหอสมุดมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ได้มีการเก็บรวบรวมองค์ความรู้จากประสบการณ์การทำงานของคณะอาจารย์ผู้เชี่ยวชาญในทางด้านศาสตร์ต่าง ๆ ในรูปแบบลายมือและสื่อสิ่งพิมพ์ไม่จำกัดเป็น หนังสือ หนังสือรวมถึงบันทึกเหตุการณ์ในอดีตในรูปของจดหมายเหตุเพื่อส่งต่อประวัติศาสตร์ความรู้ไปยังคนรุ่นหลังโดยมีการจัดเก็บอยู่ภายในห้องหมายเหตุที่มีเจ้าหน้าที่บรรณารักษ์เป็นผู้ดูแล และเนื่องจากการที่ หนังสือ หนังสือยังไม่ได้มีการจัดเก็บในรูปแบบดิจิทัลทำให้มีบุคลากรยานอกรที่ต้องการข้อมูลเพื่อนำไปทำกิจกรรมต่าง ๆ ไม่ว่าจะเป็นการทำวิจัย รายงาน หรือหาข้อมูลเพื่อประกอบการประชุมก็ตามแต่ ก็จำเป็นที่จะต้องมาติดต่อเจ้าหน้าที่บรรณารักษ์ผู้ดูแลเพื่อที่จะให้เจ้าหน้าที่บรรณารักษ์ทำการค้นหาหนังสือที่มีเนื้อหาตามที่เราต้องการ ซึ่งการค้นหาข้อมูลที่ต้องการนั้นเจ้าหน้าที่จะต้องทำการค้นหาด้วยระบบเครื่องสำอางให้ทำการค้นหาข้อมูลดำเนินการไปอย่างล่าช้า นอกจากนั้นวิธีการหาข้อมูลของเจ้าหน้าที่บรรณารักษ์จะเลือกตรวจสอบข้อมูลของหนังสือจากการดูสารบัญทำให้ข้อมูลที่ได้รับมาอาจจะแตกหักจากข้อมูลเดิมอีกด้วย

เพื่ออำนวยความสะดวกให้กับบรรณารักษ์ในการสืบค้นข้อมูลและทำให้การบริการในการสืบค้นหนังสือต่าง ๆ และให้บุคลากรยานอกรสามารถทำการค้นหาข้อมูลได้ด้วยตนเองครบถ้วนทางคณิตศาสตร์จัดทำโครงการจึงได้พัฒนาระบบการจัดเก็บหนังสือและระบบการค้นหาโดยการใช้เครื่องมือในการทำ OCR เพื่อแปลงหนังสือให้อยู่ในรูปแบบของหนังสือ ดิจิทัล และหาคำสำคัญในการสร้างคำสำคัญ ด้วยวิธี Term Frequency - Inverse Document Frequency เพื่อเพิ่มประสิทธิภาพให้กับการค้นหา

1.3 ประเภทของโครงงาน

นำเสนอความต้องการของผู้ใช้งานได้ส่วนเสียงภาษาไทย

1.4 วิธีการที่นำเสนอ

ระบบการค้นหาหนังสือ มีขั้นตอนการทำงานดังนี้

1. นำหนังสือมาแปลงเป็นรูปภาพในรูปแบบสแกน
2. นำรูปภาพเข้าสู่ระบบโดยใช้การรับส่งข้อมูลแบบ RESTful API ในระบบประเภทของการใช้งาน
3. นำรูปภาพผ่านกระบวนการเตรียมข้อมูลรูปภาพ โดยใช้ OpenCV ในการลบส่วนอื่น ๆ ที่ไม่ใช่ข้อความออกและตัดเฉพาะข้อความเพื่อนำไปใช้ในขั้นตอน
4. นำรูปที่ผ่านการเตรียมข้อมูลรูปภาพ มาเข้าสู่ระบบ OCR เพื่อแปลงข้อมูลจากรูปภาพมาเป็นข้อความในระบบดิจิทัล
5. นำข้อมูลที่เก็บไว้มาทำการตัดแบ่งคำภาษาไทยและแก้คำผิด
6. ค้นหาคำสำคัญโดยใช้วิธี TF-IDF เพื่อนำมาใช้ในการสร้างคำสำคัญ

7. นำข้อมูลที่ถูกแปลงเก็บและข้อมูลเกี่ยวกับคำสำคัญ ลงในดาต้าเบส
8. ทำระบบค้นหาในรูปแบบ Cosine Similarity
9. ทำระบบหากำลังโดยใช้วิธี Word2Vec
10. ทำแพลตฟอร์มเว็บไซต์เพื่อเป็น User Interface ให้กับผู้ใช้งานได้ใช้งานสำหรับการใช้งานในการค้นหาข้อมูลและเพิ่มข้อมูลหนังสือลงไปในฐานข้อมูลเพิ่ม

1.5 วัตถุประสงค์

1. สร้างระบบแปลงข้อมูลหนังสือให้อยู่ในรูปแบบดิจิทัล
2. สร้าง web platform เพื่อทำการค้นหาหนังสือจากคำค้น และพัฒนาเครื่องมือสนับสนุนการทำงานของบรรณาธิการประจำห้องบรรณสารสนเทศ
3. สร้างระบบการค้นหาโดยการใช้วิธีการ อินโฟเมชันเรทิฟวอล ซึ่งวัดความใกล้เคียงกันระหว่างคำค้นและข้อมูลในฐานข้อมูลโดยวิธี Cosine Similarity
4. เพิ่มประสิทธิภาพในการเข้าถึงข้อมูลในรูปแบบดิจิทัล
5. เรียนรู้เรื่องการเตรียมข้อมูลรูปภาพ

1.6 ขอบเขตของงานวิจัย

1. ระบบแปลงข้อมูลจากหนังสือและหนังสือเก่า รองรับเฉพาะหนังสือที่เป็นตัวอักษรแบบพิมพ์ และรองรับไฟล์หนังสือเฉพาะ PDF เท่านั้น
2. ระบบตัดคำ Stop word ภาษาไทยโดยอ้างอิงมาจาก pythainlpr แลกภาษาอังกฤษจาก nltk
3. ระบบค้นหาแบบ Cosine Similarity ในระบบ Information retrieval
4. ข้อมูลหนังสือที่นำมาใช้คือหนังสือจำพวก งานแสดงกตเวทิตาจิต หนังสือรายงานประจำปี ตั้งแต่ปี พุทธศักราช 2527 ถึง 2560 รวมประมาณ 43 เล่ม จากหอดหมายเหตุมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี
5. ทำ platform เว็บไซต์ในรูปแบบ responsive แต่ไม่รองรับขนาดมือถือ รองรับเฉพาะคอมพิวเตอร์หรือโน้ตบุ๊ก
6. การแปลงสิ่งพิมพ์เป็นดิจิทัลใช้ Tesseract ใน การแปลงหนังสือและหนังสือเป็นรูปแบบดิจิทัล
7. การตัดคำภาษาไทยทางคณบัญชี จะใช้ freeware เช่น DeepCut มาใช้ในส่วนของการตัดคำภาษาไทย

1.7 เนื้อหาทางวิศวกรรมที่เป็นต้นฉบับ

- การเตรียมข้อมูลรูปภาพ สำหรับการเตรียมภาพก่อนนำไปทำ OCR

โปรเจคของเราทำเกี่ยวกับการทำ OCR เพื่ออ่านภาพให้กลายเป็น text แต่ถึงแม้ว่าภาพที่ได้มาจะจากการสแกนหรือการถ่ายรูป แต่ถึงอย่างนั้น OCR ที่ใช้ก็ยังคงมีข้อจำกัดในเรื่องของคุณภาพของภาพที่ใช้ ถ้าเกิดว่าภาพที่ใช้มี noise จะทำให้การอ่านมีประสิทธิภาพน้อยลง นอกจากนี้การตัดภาพแยกย่อหน้าแต่ละย่อหน้าทำให้การอ่านมีความถูกต้องมากยิ่งขึ้น

- การพัฒนาเว็บไซต์สำหรับการค้นหาหนังสือในห้องสมุด

เว็บไซต์ของเราจะใช้ ReactJS, NodeJs, python ใน การพัฒนาเว็บไซต์เป็น Interface ให้กับผู้ใช้งาน สำหรับการใช้งานระบบการค้นหาหนังสือ รวมถึงการอัปโหลดหนังสือเพื่อแปลงหนังสือเข้าสู่ระบบดิจิทัลและ API ต่าง ๆ

- คัดเลือกคำสำคัญอ กามาเพื่อสร้างคำสำคัญของหนังสือ

สำหรับแบ่งแยกหมวดหมู่ของหนังสือโดยใช้ หลักการของ TF-IDF ใน การค้นหาคำสำคัญของหนังสือเพื่อนำมาสร้างคำสำคัญของหนังสือเล่นน้ำใจในระบบ และใช้สำหรับการค้นหาข้อมูล

- ทำระบบค้นหาโดยใช้คำที่มีความหมายใกล้เคียง

สำหรับการค้นหารายละเอียดของหนังสือโดยใช้ หลักการของ TF-IDF มาใช้เป็นคะแนนเพื่อใช้ในการค้นหาแบบ Cosine Similarity และค้นหาคำใกล้เคียง (Query Expansion) เพื่อทำให้การค้นหาเจอผลลัพธ์ที่ต้องการเพิ่มมากขึ้น

1.8 การแยกอ ยงาน และร่างแผนการดำเนินงาน

- ศึกษาและค้นคว้าปัญหาของโครงการ
- เสนอหัวข้อโปรเจค
- ค้นหาข้อมูลเกี่ยวกับเทคโนโลยีที่ใช้ในโปรเจค
- ประเมินความเป็นไปได้และกำหนดขอบเขตของโปรเจค
- จัดเก็บ requirement จากกลุ่มผู้ใช้งาน
 - ติดต่อเจ้าหน้าที่ของห้องสมุด
 - เก็บข้อมูลที่ต้องการแปลงเข้าสู่ระบบดิจิทัล
- นำเสนอโครงการครั้งที่ 1
- ออกแบบ UX/UI
- การแปลงรูปภาพให้อยู่ในรูปแบบดิจิทัล
 - นำหนังสือมาแปลงเป็นรูปภาพในรูปแบบสแกน
 - ศึกษาการใช้งาน OpenCV
 - สร้างระบบการเตรียมข้อมูลรูปภาพ เพื่อทำการปรับแต่งรูปภาพและทำการปรับแต่งจนได้ระบบที่รองรับกับ Data ที่มี
 - นำรูปที่ผ่านการเตรียมข้อมูลรูปภาพ มาเข้าสู่ระบบ OCR เพื่อแปลงข้อมูลจากรูปภาพมาเป็นข้อความในระบบดิจิทัล
- นำข้อมูลที่เก็บไวมาทำการตัดแบ่งคำภาษาไทยและหาคำสำคัญโดยใช้ TF-IDF
 - ทำการตัดแบ่งคำ (Tokenization)
 - ลบ stop word ออกจากข้อมูล
- ทำระบบค้นหา
 - ทำระบบค้นหาโดยใช้หลักการ Cosine Similarity
 - ทำการค้นหาด้วยคำใกล้เคียงโดยใช้ Word2Vec
- จัดทำเว็บไซต์แพลตฟอร์ม
- ทดสอบระบบ

13. ปรับปรุงแก้ไข
 14. นำเสนอโครงการ

1.9 ตารางการดำเนินงาน

ตารางที่ 1.1 ตารางการดำเนินงาน ภาคการศึกษาที่ 1/2563

ตารางที่ 1.2 ตารางการดำเนินงาน ภาคการศึกษาที่ 2/2563

ที่	หัวข้อ	ตารางการดำเนินงาน ภาคการศึกษาที่ 2/2563													
		มกราคม				กุมภาพันธ์				มีนาคม					
1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	จัดทำระบบการค้นหา														
2	จัดทำเว็บไซต์														
3	ทดสอบระบบ														
4	ปรับปรุงแก้ไข														
5	นำเสนอโครงการ														

1.9.1 ผลการดำเนินงานในภาคการศึกษาที่ 1

- ทำระบบเตรียมข้อมูลรูปภาพ สำหรับการเตรียมรูปภาพสำหรับการแปลงข้อมูลเป็นดิจิทัล
- ทำ API ในการตัดคำและจัดการ stop word สำหรับการเตรียมการเตรียมข้อมูลตัวอักษร
- ทำระบบ Term Frequency-Inverse Document Frequency สำหรับการค้นหาคำสำคัญเพื่อระบุคำสำคัญของหนังสือเล่มนั้นภายในระบบ
- ทำส่วนของการทำการค้นหาข้อมูลเบื้องต้น

1.9.2 ผลการดำเนินงานในภาคการศึกษาที่ 2

- ทำระบบค้นหาให้เสร็จสิ้น
- ปรับปรุงระบบค้นหาให้ตอบโจทย์มากยิ่งขึ้น
- ทำเว็บไซต์ทั้งผึ้ง frontend และ backend

บทที่ 2 ที่มา ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 บทนำ

โดยทฤษฎีที่เกี่ยวข้องกับโปรเจคนี้มีหลากหลายสาขาด้วยกันโดยจะแบ่งเป็นส่วนของการเตรียมข้อมูลรูปภาพ โดยการใช้ Open source Computer Vision (OpenCV) เพื่อนำไปใช้กับส่วนของการทำ Optical Character Recognition (OCR), Tesseract OCR และส่วนของการทำ Natural language processing (NLP) โดยการใช้ Team Frequency Inverse Document Frequency (TF-IDF), Minimum Edit Distance, Deep Cut ส่วนต่อไปคือสร้างระบบการค้นหาได้ใช้ Cosine Similarity และในส่วนการสร้างเว็บไซต์โดยใช้ RESTful API และส่วนสุดท้ายการทำ Word Embedding

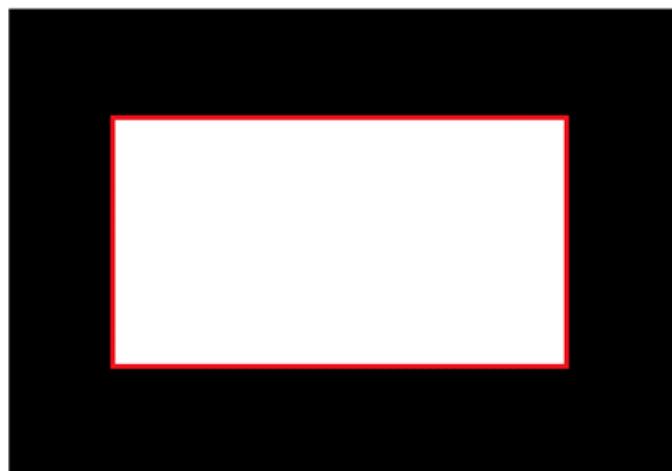
2.2 แนวความคิดทางทฤษฎี

2.2.1 การเตรียมข้อมูลรูปภาพ

เป็นการประมวลผลรูปภาพที่แปลงภาพให้เป็นข้อมูลทางดิจิทัลเพื่อใช้สำหรับปรับคุณภาพของภาพให้ตรงตามความต้องการ อย่างการตัดสิ่งรบกวน การลบกรอบ การหมุนรูป หรือการปรับรับให้ภาพมีความคมชัดมากยิ่งขึ้น ในโปรเจกของเรานั้นนำมาใช้ในการปรับคุณภาพของรูปภาพเพื่อช่วยให้การทำ OCR แม่นยำมากยิ่งขึ้น

2.2.1.1 คอนทัว (Contour)

คอนทัว (Contour) [4] คือเส้นค้าโครงของรูปภาพ ที่วิ่งทางรอบเขตพื้นที่ที่มีค่าสีต่อเนื่องกัน หรือค่าเดียวกัน โดยใช้การเปลี่ยนให้รูปภาพอยู่ในรูปของ matrix และเช็คดูว่าค่าสีที่มีความแตกต่างอย่างชัดเจนเริ่มที่ตรงไหนและสร้างเป็นเส้นค้าโครงขึ้นมาดังรูป 2.1 ซึ่งการทำเส้นค้าโครงจะทำงานได้ถูกต้องเมื่อเป็นรูปภาพแบบ Binary

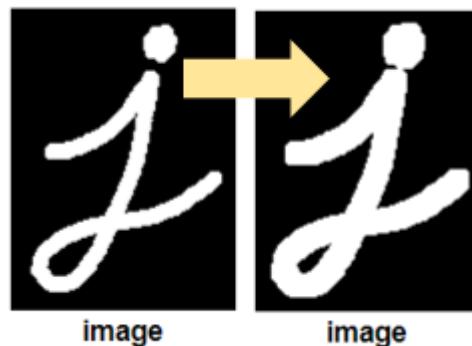


รูปที่ 2.1: แสดงการทำค้าโครงภายในรูป

2.2.1.2 การเปลี่ยนแปลงทางสัณฐานวิทยา (Morphology Transformation)

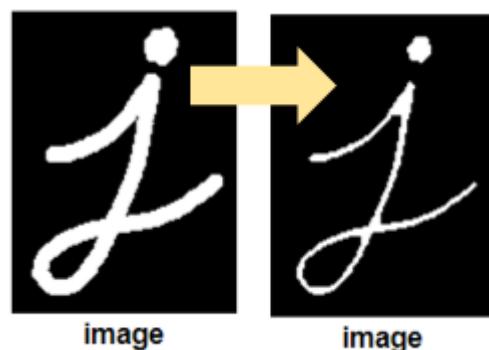
เป็นกระบวนการเตรียมข้อมูลรูปภาพ ที่จะทำการนำรูปภาพมาทำการเปลี่ยนแปลงลักษณะ รูปร่างของวัตถุภายในภาพ ปกติแล้วจะใช้ภาพที่เป็น Binary ซึ่งส่วนใหญ่จะใช้สำหรับการกำจัด noise การซ่อมแซมรูปร่างของภาพ หรือการเพิ่มขนาดให้กับวัตถุนั้นๆ โดยการทำการเปลี่ยนแปลงทางสัณฐานวิทยา (Morphology Transformation) นั้นจะมีวิธีการดำเนินการพื้นฐานอยู่ 2 วิธีคือการขยายภาพ และการร่อนภาพ

Dilation คือการเพิ่มพื้นที่สีขาวของรูปเพิ่มพื้นที่สีขาวตามขอบพื้นที่สีขาวและจะเปลี่ยนพื้นที่สีดำให้กลายเป็นสีขาวทำให้พื้นที่สีขาวมีความหนามากขึ้นดังรูป



รูปที่ 2.2: แสดงการทำการขยายภาพ (Dilation) เพื่อเพิ่มพื้นที่สีขาว

Erosion คือการกร่อนภาพ หรือคือจะลดพื้นที่สีขาวของภาพออกไปซึ่งวิธีการนี้ส่วนใหญ่จะใช้สำหรับการแยกสิ่งที่ของที่อยู่ติดกัน หรือลบ pepper noise ที่เป็น noise เเล็กๆได้ โดยจะใช้หลักการเดียวกับการขยายภาพ (Dilation) เพียงแต่จะเปลี่ยนจากพื้นที่สีขาวให้กลายเป็นพื้นที่สีดำดังรูป



รูปที่ 2.3: แสดงการทำกร่อนภาพ (Erosion) เพื่อกร่อนพื้นที่สีขาว

2.2.2 Optical Character Recognition (OCR)

OCR เป็นกระบวนการของการแปลงอักษรบนสื่อสิ่งพิมพ์ให้เป็นข้อความที่สามารถค้นหา เปลี่ยนแปลงและแก้ไขได้โดยที่ไม่ต้องพิมพ์ขึ้นมาใหม่ ด้วยการทำ Deep learning ใน การเรียนรู้ภาพเพื่อแปลงอักษรเป็นตัวอักษร ซึ่งในปัจจุบันทางผู้จัดทำต้องทำระบบเกี่ยวกับคันหาที่จะต้องคัดคำอ่านมาจากสิ่งพิมพ์เหล่านั้น จึงจำเป็นที่จะต้องใช้ OCR ในการแปลงภาพดันแบบอักษรให้เป็นตัวอักษรก่อนที่จะนำไปใช้งานต่อ

จากการศึกษาพบว่าการทำ OCR ภาษาไทยนั้นมืออยู่มากมายในปัจจุบัน หนึ่งในนั้นมี T - OCR ซึ่งเป็น Library ของ AI For Thai [5] และ Tesseract ของ Google [6] ที่ใช้สำหรับแปลงภาพเป็นตัวอักษร โดยกลุ่มของเราเลือกที่จะใช้ Tesseract ในการทำ OCR เนื่องจากไม่เสียค่าใช้จ่ายเมื่อเทียบกับการใช้ OCR ของ AI For Thai นอกจากนั้นเรื่องของการเรียกใช้งานอย่างต่อเนื่อง Tesseract สามารถทำได้ดีกว่าเนื่องจากไม่จำเป็นต้องเรียกใช้งาน AI For Thai จากภายนอก

2.2.3 Natural language processing

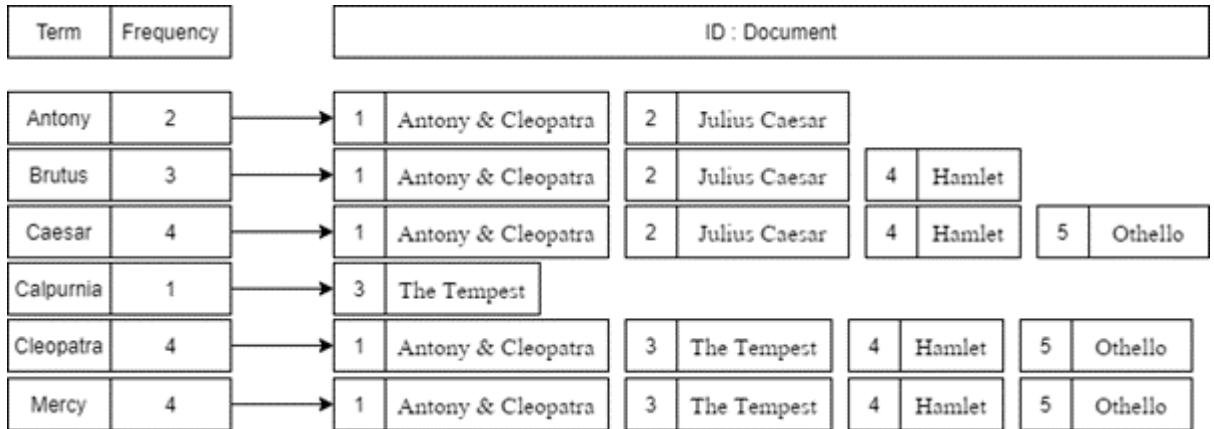
Natural language processing คือกระบวนการที่ใช้ในทางปัญญาประดิษฐ์ซึ่ง เป็นกระบวนการที่ทำการวิเคราะห์ทางด้านภาษาซึ่งเอาไปประยุกต์ทำให้ปัญญาประดิษฐ์ (AI) สามารถทำให้คอมพิวเตอร์เข้าใจภาษาและตอบกลับได้ใกล้เคียงกับมนุษย์มากขึ้น โดยในปัจจุบันนี้จะใช้มาช่วยในการหาคำสำคัญของหนังสือ และบทความต่าง ๆ เพื่อช่วยให้การค้นหาบทความมีประสิทธิภาพมากขึ้น

2.2.3.1 Information retrieval

Information retrieval คือ เทคโนโลยีการเก็บข้อมูลอย่างนึงโดยจะมีทั้งหมด 2 ลักษณะ ลักษณะที่ 1 คือ Boolean Retrieval เป็นการสร้างโครงสร้างข้อมูลในรูปแบบ Matrix ที่มีค่าเพียงแค่ 0,1 โดยที่ 0 คือไม่มีคำ (Term) ในหนังสือนั้น และ 1 คือมีคำ (Term) อยู่ภายในหนังสือนั้นหรือเรียกได้วาเป็น Term-Document Incidence Matrix ดัง ตารางที่ 2.1 โดยที่ถ้าเราพิจารณาในรูปแบบแคนเรางจะได้ Vector ของ Term นั้นที่ประกอบอยู่ในหนังสือ ไหนบ้าง แต่การเก็บในรูปแบบ Boolean Retrieval เมื่อมีหนังสือ เยอะขึ้นจะทำให้เกิดค่า 0 ที่ไม่มีประโยชน์มากขึ้นจึงมีลักษณะที่ 2 คือโครงสร้างแบบ Inverted index เป็นการเก็บเพียง Term นั้นอยู่ภายใต้หนังสือ ไหนบ้างเพื่อจะเก็บแต่เพียงชื่อข้อมูลสำคัญเอาไว้ดัง ตารางที่ 2.2 โดย คำ (Term) จะผ่านกระบวนการเตรียมข้อมูลตัวอักษร ประกอบไปด้วย Tokenization (การตัดคำจากประโยค), Normalization (การจัดการคำย่อ), Stemming (การแปลงคำให้อยู่รูปแบบเดียวกัน), Stop words (จัดการคำที่ไม่มีความหมาย) เพื่อเป็นการจัดรูปของคำให้อยู่ในรูปแบบเดียวกันก่อนที่จะนำไปใช้งาน ซึ่งการเก็บชื่อข้อมูลแบบ Information retrieval (IR) จะทำให้การค้นหาข้อมูลภายในฐานข้อมูลได้อย่างรวดเร็วและมีประสิทธิภาพ

ตารางที่ 2.1 Information retrieval ในลักษณะ Boolean Retrieval

	Antony & Cleopatra	Julius Ceasar	The Tempest	Hamlet	Othello
Antony	1	1	0	0	0
Brutus	1	1	0	1	0
Ceasar	1	1	0	1	1
Calpurnia	0	0	1	0	0
Cleopatra	1	0	1	1	1
Mercy	1	0	1	1	1



รูปที่ 2.4: Information retrieval ในลักษณะ Index Retrieval

2.2.3.2 TF-IDF

เป็นเทคนิคในการคัดแยกคำตามความสำคัญผ่านการให้น้ำหนักในแต่ละคำ โดยแบ่งเป็นสองส่วนนั้นก็คือ TF (Term Frequency) เป็นการดูว่าคำนี้ หรือว่า Term นี้ปรากฏขึ้นภายใน document มากน้อยเพียงไหน และ IDF (Inverse Document Frequency) คือการหาความผกตันในความถี่ของหนังสือโดยคะแนนความผกตันที่ทำให้รู้ว่าคำนี้เป็นคำที่มีความสำคัญเฉพาะภายนอกหนังสือนี้ แต่เมื่อจากการดูความถี่ของ IDF เพียงอย่างเดียวไม่สามารถบอกได้ว่า Term นั้นเป็นคำสำคัญ จึงจำเป็นต้องนำคำ TF มาคูณกับ IDF เป็นคำ TF-IDF เพื่อถูกความสำคัญของ Term นั้น ในส่วนการคำนวณนี้เพื่อนำไปใช้ในการค้นหาแบบ Cosine Similarity ต่อไป โดยที่ TF จะใช้เป็น Log normalization โดยคำนวณได้จากสมการ 2.1 ซึ่ง $f_{t,d}$ คือความถี่ของคำ (Term) ที่ปรากฏขึ้นภายใน Document ส่วน IDF จะคำนวณจากสมการ 2.2 ซึ่ง N คือจำนวน Document ที่มีภายในระบบ และ n_t คือ จำนวนของ document ที่มีคำ (term) นั้นอยู่ เมื่อหาค่าทั้ง TF และ IDF ได้แล้วก็จะหาค่าของ TF-IDF ได้จากสมการ 2.3

$$tf = \log(1 + f_{t,d}) \quad (2.1)$$

$$idf = \log \frac{N}{n_t} \quad (2.2)$$

$$TF - IDF = tf * idf \quad (2.3)$$

2.2.3.3 Cosine Similarity

เป็นหน่วยวัดความคล้ายคลึงกันระหว่างข้อมูลสอง Vector โดยวัดจากมุม cosine ของจาก Vector ทั้งสองโดยคำนวณได้จากสมการ 2.4 โดยที่ $\|x\|, \|y\|$ คือ สมการของ Euclidean norm ของ Vector x, y ดังสมการ 2.5 โดยในโปรเจคนี้เราได้นำค่าน้ำหนักของ TF-IDF มาเป็นน้ำหนักในการคิดคำ Cosine Similarity โดยนำประโยชน์ที่จะค้นหามาผ่านกระบวนการเตรียมข้อมูลตัวอักษร ก่อนที่จะนำมาค้นหา document ให้มีค่า relevance score (คะแนนความสัมพันธ์) เพื่อนำมาเรียงลำดับและแสดงเป็นผลลัพธ์การค้นหา

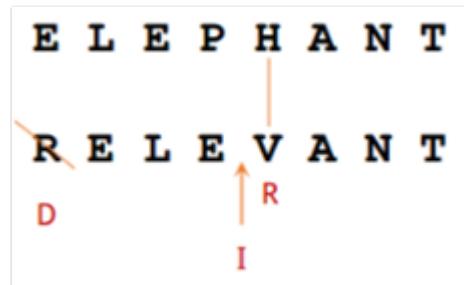
$$\sin(x, y) = \frac{x * y}{\|x\| \|y\|} \quad (2.4)$$

$$\|x\| = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2} \quad (2.5)$$

2.2.3.4 Minimum Edit Distance

เป็นหลักการที่หารระยะห่างที่สั้นที่สุดจากคำนึงไปสู่อีกคำนึงจะมีความแตกต่างกันเท่าไหร่ซึ่งหลักการเช็คความห่างของคำทั้งหมดสามรูปแบบ

- รูปแบบ Insert(I) จะเป็นการเพิ่มตัวอักษรลงในคำนั้น เพื่อคำดังเดิมของเราจะเปลี่ยนแปลงเป็นคำที่เราต้องการ
- รูปแบบ Delete(D) จะเป็นการลบตัวอักษรออกในคำนั้น เพื่อคำดังเดิมของเราจะเปลี่ยนแปลงเป็นคำที่เราต้องการ
- รูปแบบ Replace(R) จะเป็นการเปลี่ยนตัวอักษรนั้นให้เป็นตัวอักษรใหม่ เพื่อคำดังเดิมของเราจะเปลี่ยนแปลงเป็นคำที่เราต้องการ



รูปที่ 2.5: หลักการการเช็ค edit distance [1]

หลังจากมีรูปแบบการวัดระยะห่างของคำดังรูปภาพที่ 2.5 และ จะต้องทำการหาคำที่สั้นที่สุดผ่านรูปแบบของตารางดังรูปภาพที่ 2.6 ซึ่งการคำนวนผ่านตารางจะเป็นการนำกระทำก่ออนหน้ามาคำนวนเรื่อยๆ จนได้รูปการเปลี่ยนเป็นคำใหม่ที่ใช้การเปลี่ยนน้อยที่สุด

		E	L	E	P	H	A	N	T
	0	1	2	3	4	5	6	7	8
R	1	1	2	3	4	5	6	7	8
E	2	1	2	2	3	4	5	6	7
L	3	2	1	2	3	4	5	6	7
E	4	3	2	1	2	3	4	5	6
V	5	4	3	2	2	3	4	5	6
A	6	5	4	3	3	3	3	4	5
N	7	6	5	4	4	4	4	3	4
T	8	7	6	5	5	5	5	4	3

รูปที่ 2.6: ตัวอย่างตารางการทำ Minimum edit distance [1]

ซึ่งในโครงการของเราได้ดึงหลักการ Minimum edit distance มาใช้ในการตรวจสอบหาคำที่สะกดไม่ถูกต้องโดยมีเกณฑ์ตั้งไว้ว่าถ้าเกินที่กำหนดไว้จะถือว่าคำ ๆ นั้นสะกดไม่ถูกต้องแล้วถูกแก้ให้เป็นคำที่สะกดถูกต้อง

2.2.3.5 Spelling Corrector by Peter Norvig

หลักการแก้คำผิดของ Peter Norvig[7] เป็นการนำคำที่ถูกพิมพ์เข้ามาสร้างมาเป็นคำใหม่โดยการแบ่งคำ การลบคำ การกลับคำ การแทนที่คำ การเพิ่มคำ และนำคำที่สร้างใหม่ไปเช็คในพจนานุกรมภาษา่าว่ามีคำที่สร้างใหม่คำนั้นหรือไม่ ถ้ามีให้นำคำที่ได้ไปหาค่าความน่าจะเป็นของคำที่ไม่ออกเสียงเดียวกันมากที่สุด ซึ่งใน Library นี้ Norvig ได้นำไปหาโดยใช้ข้อมูลจาก Corpus ที่สร้างขึ้นจากคำที่ได้จาก Project Gutenberg[8], Wikitionary[9] และ British National Corpus[10] ที่จะรวมคำต่างๆอ้าไว จากนั้นคำนวนความน่าจะเป็นของคำที่เกิดขึ้นโดยใส่คำที่ต้องการหาแล้วนับว่าใน Corpus นั้นมีคำนั้นปรากฏขึ้นเท่าไร หากตัวย่อจำนวนคำของ Corpus ทั้งหมด ก็จะได้ความน่าจะเป็นอย่างมาก ซึ่งคำที่จะแก้ก็จะถือจากคำที่มีค่าความน่าจะเป็นมากที่สุด โดย library PyThaiNLP[11] และ Pyspellchecker[12] นั้นก็ใช้หลักการเดียวกันกับการแก้คำผิดของ Peter Norvig

2.2.4 RESTful Service

เป็นการสร้าง web service โดยเรียกใช้ผ่านทาง HTTP Method ทั้ง 4 ประเภท GET/POST/PUT/DELETE ส่งข้อมูลอุปกรณ์เป็นรูปของ XML ทำให้ปริมาณข้อมูลที่ส่งมากน้อยกว่าการใช้ Protocol SOAP โดยโครงสร้างของ HTTP Request ดังรูปภาพที่ 2.7 ประกอบด้วย

1. VERB: แสดง method ของ HTTP
2. URI: ตำแหน่งของข้อมูลที่ต้องการ
3. HTTP Version: เวอร์ชันของ HTTP
4. Request Header: Metadata ที่เก็บข้อมูลในรูปแบบ Key-Value ของ header
5. Request Body: ส่วนเก็บข้อมูลของเนื้อหา

HTTP Request



รูปที่ 2.7: แสดงลักษณะโครงสร้างของ HTTP Request [2]

HTTP Response ดังรูปภาพที่ 2.8 ประกอบด้วย

1. HTTP Version: เวอร์ชันของ HTTP

2. Response Code: รหัสผลลัพธ์ของการทำงานในระดับ HTTP เป็นเลข 3 หลัก
3. Response Header: Metadata ที่เก็บข้อมูลในรูปแบบ Key-Value ของ header
4. Response Body: ส่วนเก็บข้อมูลของเนื้อหา

HTTP Response



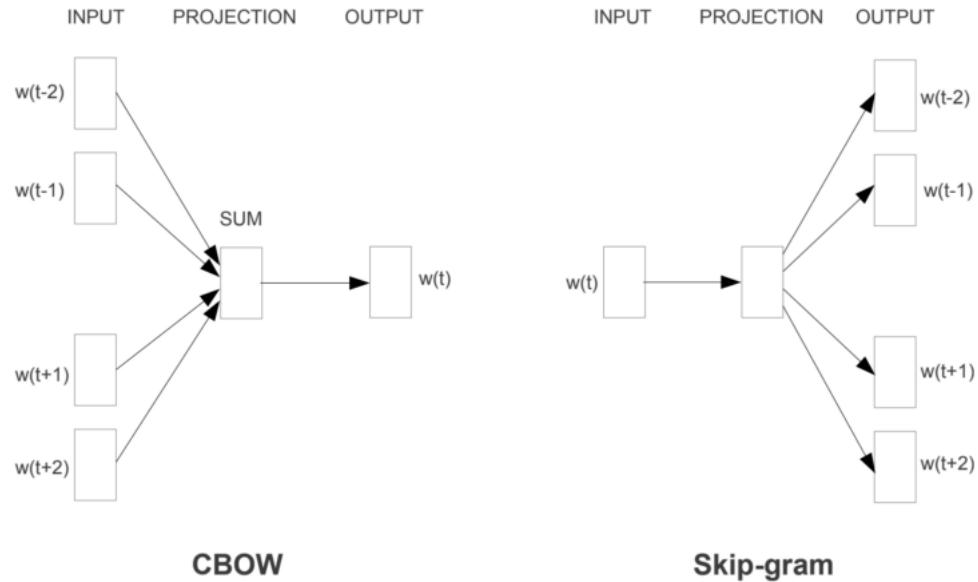
รูปที่ 2.8: แสดงถึงโครงสร้างของ HTTP Response [2]

2.2.5 Word Embedding

เป็นวิธีการที่จะเปลี่ยนคำปกติเป็น vector ที่อยู่ในหลักหลายมิติและขนาดเพื่อให้สามารถเปรียบเทียบคำต่าง ๆ ว่ามีความสัมพันธ์ใกล้เคียงกับคำไหนบ้างในระบบเพื่อ ที่ใช้สำหรับการทำคำที่มีความหมายใกล้เคียงกันโดยมีการทำ word embedding มากราฟมายไม่ว่าจะเป็น Word2Vec [13] [14] ที่ถูกสร้างโดยทีมวิจัยของ Google FastText [15] เป็น word embedding อีกหนึ่งตัวที่สร้างขึ้นจากทีมวิจัยของ facebook หรือจะเป็น ELMo [16] ที่เป็นรูปแบบการ word embedding ที่ดูรูปโดยรอบเป็นต้น

2.2.5.1 Word2Vec

ทฤษฎี Word2Vec[17] เป็นทฤษฎีที่ช่วยจัดการรับคำศัพท์ที่มีความหมายใกล้เคียงกัน หรือคำศัพท์ที่มีความสัมพันธ์อย่างตรงข้ามกันอย่างคำว่า พระราชา กับ กษัตริย์ ที่เป็นคำที่ความหมายใกล้เคียงกัน แต่ตรงข้ามกับคำว่า ราชินี โดยทั้งหมดนี้จะถูกจัดเก็บในรูปแบบของเวกเตอร์ แน่นอนว่าการที่จะรู้ว่าแต่ละคำมีความสัมพันธ์ได้นั้น เราจำเป็นต้องสร้างประเภทของคำจากรูปประโยคของคำทั้งสอง “การ เลี้ยง หมา เป็น สัตว์เลี้ยง ถือ เป็น การ ผ่อนคลาย” กับ “แมว ถือ เป็น สัตว์เลี้ยง ที่ สุขุม” จากทั้งสองประโยคจะเห็นว่าคำที่เหมือนกันจากทั้งสองประโยค จะเป็นคำว่า สัตว์เลี้ยง ซึ่งการทำ Word2Vec จะเป็นการดึงประเภทโดยเชื่อมต่อจากคำใกล้เคียงเพื่อจัดกลุ่ม ดังนั้นจากคำว่า สัตว์เลี้ยง ที่มีคำว่า หมา และ แมว เป็นสัตว์เลี้ยงเช่นเดียวกัน จะแสดงว่าคำว่า หมา และ แมว มีความหมายใกล้เคียงกันจากความสัมพันธ์ทั้งนี้การที่แต่ละคำจะมีสัมพันธ์ที่เข้มข้นกันได้มาก หรือน้อยขึ้นอยู่รูปแบบของคำ (Window Size) ยกตัวอย่างถ้ามีคำมาก คำว่าสัตว์เลี้ยง อาจจะมีคำที่ไม่เกี่ยวข้องเกิดขึ้นหากข้อมูลไม่เพียงพอ เช่น ตุ๊กตา หมา แมว เป็นสัตว์เลี้ยง แต่ถ้าหากคำน้อยก็จะไม่ถูกนับ เป็นประเภทสัตว์เลี้ยงแทน นอกจากส่วนการใช้คำข้างเคียงมาช่วยในการทำ Word2Vec แล้วยังมีส่วนของรูปแบบการสร้างโมเดลเพื่อหาคำที่มีความสัมพันธ์กันอีกด้วย โดยจะมีสองรูปแบบคือ CBOW (continuous bag of words) จะเป็นรูปแบบของการใช้คำหลาย ๆ คำ เพื่อมาหาคำเพียงคำเดียว และ Skip-Gram ที่จะตรงข้ามกับรูปแบบที่แล้วคือ การที่ใช้คำหนึ่งคำเพื่อหาคำหลายคำอีกมา



รูปที่ 2.9: ภาพแสดงโมเดล Skip-gram และ CBOW [3]

2.2.5.2 Skip-gram

Skip-gram[3] เป็นโมเดลที่จะใช้ความสัมพันธ์ของคำ โดยจะนำคำที่เข้าไปเป็นคำศักริบบ์โดยรอบ โดยการที่โมเดลจะทำการอ่านประโยค เข้าไปและเรียนรู้จากประযุก โดยการพิจารณาบริบทโดยรอบของคำ อย่างเช่น Thou shalt not make a machine in the likeness of human a mind โมเดลก็จะทำการรับคำ 1 คำเข้าไปและพิจารณาคำรอบๆ เริ่มจากว่า not ก็จะพิจารณาสองคำก่อนหน้าและสองคำหลัง (ตามจำนวน window size ในที่นี้กำหนด window size เป็น 2) ซึ่งจะได้มี 4 คำ คือ Thou, shalt, make และ a ดังรูปที่ 2.10 ซึ่งโมเดล ก็จะทำการเรียนรู้คำไปเรื่อยๆ เพื่อ weight น้ำหนักภายนอกในโมเดล

Thou shalt not make a machine in the likeness of a human mind

input word	target word
not	thou
not	shalt
not	make
not	a
make	shalt
make	not
make	a
a	machine
a	not
a	make
a	machine
a	in
machine	make
machine	a
machine	in
machine	the
in	a
in	machine
in	the
in	likeness

รูปที่ 2.10: แสดงการทำงานของ Skip-gram [3]

2.2.5.3 CBOW

CBOW[3] เป็นโมเดลที่ใช้หาความสัมพันธ์ของคำแบบจะพิจารณาคำหล่ายๆ คำเพื่อหาคำหนึ่งคำ ดังตัวอย่างในรูปที่ 2.11 โดยการที่ไม่เดล จะทำการอ่านประโยคเข้าไปและเรียนรู้จากประโยคเหมือนโมเดล Skip-gram อย่างเช่น Thou shalt not make a machine in the likeness of human a mind ไม่เดลก็จะทำการอ่านคำตามจำนวน window size ถ้า window size เป็น 2 ก็จะเป็นการกำหนดว่าอ่าน input เข้าไป 2 คำและคำต่อจาก 2 คำนั้นในประโยคจะเป็นผลลัพธ์ที่ได้จากการอ่าน input อย่างเช่นถ้า input อย่างเช่น thou,shalt เข้าไปก็จะได้คำว่า not ออกมาก ซึ่งจะสร้างโมเดลนี้ก็จะต้องใส่ข้อมูลประโยคจำนวนมากเพื่อทำการ weight ค่าน้ำหนักภายนอกในโมเดล ให้มีประสิทธิภาพคล้ายกับโมเดล skip-gram

Thou shalt not make a machine in the likeness of a human mind

Sliding window across running text									Dataset		
input 1	input 2	output									
thou	shalt	not	make	a	machine	in	the	...			
thou	shalt	not	make	a	machine	in	the				
thou	shalt	not	make	a	machine	in	the				
thou	shalt	not	make	a	machine	in	the				
thou	shalt	not	make	a	machine	in	the				

รูปที่ 3: รูป CBOW ในการสร้างชุดข้อมูล [2]

รูปที่ 2.11: แสดงการทำงานของ CBOW [3]

2.3 ภาษาคอมพิวเตอร์และเทคโนโลยี

2.3.1 Open source Computer Vision (OpenCV)

เป็นซอฟต์แวร์ที่เกี่ยวกับการประมวลผลภาพที่มีการสนับสนุนการพัฒนาจาก Intel Corporation โดยที่ตัว OpenCV นั้นเป็น Library Open Source โดยมีจุดประสงค์เพื่อให้นำไปต่อยอดการพัฒนาโปรแกรมในด้าน การรับรู้ของเห็นของคอมพิวเตอร์ (Computer Vision) ให้เข้าใจไม่ว่าจะเป็นภาพนิ่ง (Image) หรือจะเป็นภาคเคลื่อนไหว (Video) โดยภายในโปรเจคนี้ได้นำ OpenCV มาเป็นตัวทำงานโดยใช้ข้อมูลรูปภาพ โดยที่นำรูปภาพที่ได้มาจากการถ่ายภาพ / หนังสือ มาทำการปรับปรุงคุณภาพรูปภาพให้เหมาะสมกับการทำส่วน Optical Character Recognition (OCR) ให้มีความแม่นยำมากยิ่งขึ้น เช่นการลบรูปภาพ การลบสิ่งที่ลับกวน การลบกรอบตาราง การหมุนรูป

2.3.2 Tesseract OCR

เป็นหนึ่งใน Library ที่เกี่ยวกับการทำ Optical Character Recognition (OCR) ที่ถูกพัฒนาโดย Google โดยเป็น Library Open Source ที่ใช้ในการทำเกี่ยวกับ Text Detection โดยสามารถเรียกใช้งานได้ผ่าน Command line หรือจะเป็นการเรียก API ภายในโปรแกรมที่ทำได้ต่างกัน Tesseract เวอร์ชัน 5.0.0 beta มีการใช้ Convolutional Neural Network (CNN) [18] ร่วมกันกับ Long Short-Term Memory (LSTM) เพื่อให้การทำนายผลได้ดีขึ้นโดยเราจะนำตัว Tesseract มาทำเป็น OCR ภายในโปรเจคนี้

2.3.3 DeepCut

เป็น Library ในภาษา Python ที่สร้างมาจาก True Corporation โดยมีลักษณะเด่นที่ใช้ CNN (Convolutional Neural network) [18] มาช่วยทำให้ผลลัพธ์ที่ได้ออกมา มีความแม่นยำที่ค่อนข้างสูง ซึ่งโปรเจกของเราต้องการ DeepCut เพื่อที่จะสามารถไปตัดคำจากรูปประโยคภาษาไทยที่มีความซับซ้อน และไม่แบ่งแยกขัตเจนเหมือนภาษาอังกฤษ

2.3.4 ReactJS

เป็นหนึ่งใน Library หรือจะเรียกว่าเป็น Framework ที่ Facebook เป็นคนสร้างขึ้นโดยที่มีหน้าที่เป็นการสร้าง UI โดยมีความคิดมากจากรูปแบบ MVC [19] (Model View Controller) หรือก็คือเป็นตัวจัดการกับ Model กับ View ของตัวเว็บไซต์ โดยในโปรเจคนี้ได้เลือกใช้ ReactJS เป็น Front End สำหรับการทำ platform Web Application

2.3.5 Python

Python เป็นภาษาทางโปรแกรมที่ใช้เป็นภาษาทางคอมพิวเตอร์ระดับสูงที่ออกแบบมาให้เกล้าคึ่งกับภาษาамนุษย์มากที่สุดเพื่อให้สามารถเข้าใจได้ร่ายมากที่สุด ซึ่งในโปรเจกมีข้อมูลที่ต้องประมวลในแต่ละครั้งมีขนาดใหญ่อาจจะทำให้เกิดความล่าช้าในแต่ละการประมวล ทางผู้จัดทำจึงเลือกใช้ Python เนื่องจากรองรับในส่วนของการทำ Thread รวมถึงนำมาใช้ในการทำ Data preparation ทั้งการเตรียมข้อมูลรูปภาพ และการเตรียมข้อมูลต่างๆ หลังจากการทำ OCR นอกจากนี้ยังใช้ในการทำ Web Server อีกด้วย

2.3.5.1 Django

เป็น REST Framework ที่ใช้ภาษา Python เป็นฐาน โดยในโปรเจคนี้เราจะนำมารีใช้ REST API เพื่อใช้ในการใช้ Library อย่างเช่น DeepCut หรือ OCR ที่สามารถใช่วิ่งการแบ่ง Multi Thread ได้อย่างมีประสิทธิภาพ และยังสามารถจัดการข้อมูลใน database สำหรับโปรเจคนี้

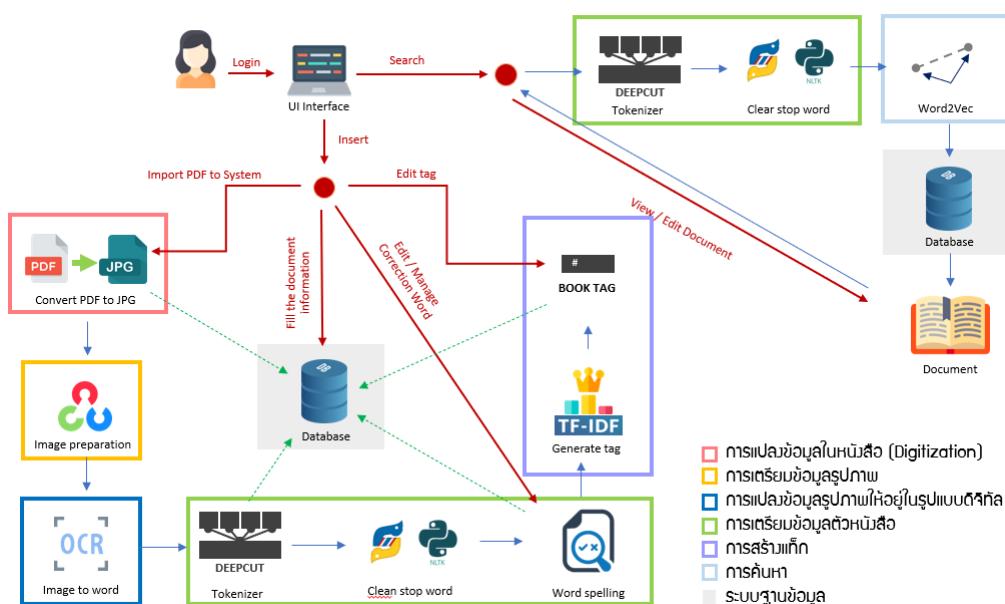
2.3.6 NodeJS

NodeJS เป็นสมมือนแพลตฟอร์มที่ใช้ภาษา JavaScript ที่มี Library สำหรับใช้จัดการกับฝั่ง Server ซึ่ง NodeJS นั้นมีความยืดหยุ่นสูงที่สำหรับการจัดการ Web Server โดย Library ที่นำมาใช้คือ Express เป็น Web Server ที่เป็น RESTful API ได้

บทที่ 3 การออกแบบและระบบวิจัย

3.1 ภาพรวมของระบบ

บทนี้จะกล่าวถึงภาพรวมของระบบโดยแสดงเป็นโครงสร้างแบบตัวอย่างรูปที่ 3.1 ซึ่งประกอบไปด้วยการออกแบบระบบฐานข้อมูล ระบบการตัดคำ ระบบการประมวลรูปภาพ และการออกแบบ User Interface สำหรับการใช้งาน



รูปที่ 3.1: ภาพรวมของระบบ

3.2 การออกแบบการทดลอง

3.2.1 การแปลงข้อมูลในหนังสือ (Digitization)

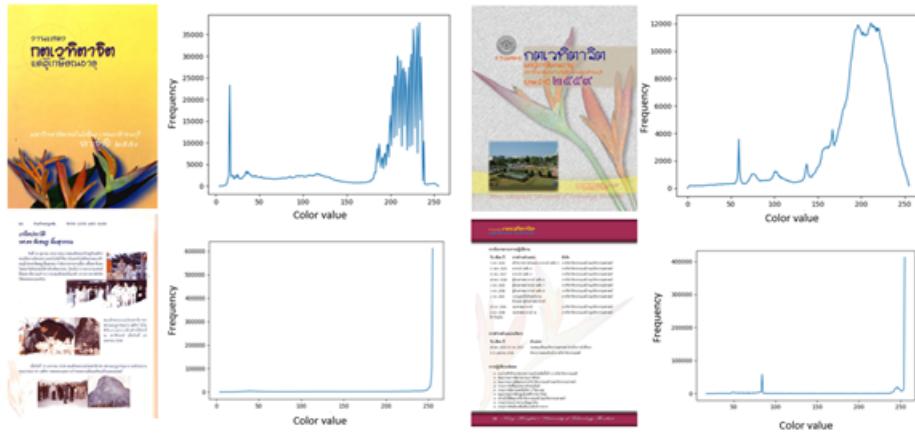
สำหรับการแปลงข้อมูลในหนังสือ (Digitization) ผู้ใช้จะต้องทำการอัปโหลดไฟล์ PDF ของหนังสือเข้าสู่ระบบหลังจากนั้นจะระบบจะทำการแปลงแต่ละหน้าเป็นรูปภาพ JPG เพื่อนำไปใช้ต่อในขั้นตอนต่อไปและนำไปแสดงภายใน web application

3.2.2 การเตรียมข้อมูลรูปภาพ

ในส่วนของการจัดการรูปก่อนที่จะทำการ OCR ซึ่งรูปภาพนำมา OCR นั้นมาจากการสแกนทำให้ภาพส่วนใหญ่อยู่ในสภาพเดิมก็ยังคงมี noise และมีความผิดพลาดจากการสแกน เช่น ภาพเอียง หรือตัวอักษรไม่ชัดเกิดจากการขยับในระหว่างการสแกน หรือมีพื้นหลังสีที่ทำให้ OCR ไม่มีประสิทธิภาพ ดังนั้นจึงต้องมีการเตรียมข้อมูลรูปภาพก่อนที่จะไปทำการ OCR ซึ่งในการเตรียมข้อมูลรูปภาพ นั้นทางคณะผู้จัดทำได้ออกแบบไว้ว่าจะทำการแยกระหว่างรูปและตัวอักษรออกจากกัน โดยการใช้คอนทัว (Contour) เข้ามาช่วยในการคัดแยกรูปออกจากตัวอักษร โดยดูจากพื้นที่สีเหลี่ยมที่ของค่อนทัว (Contour) กับพื้นที่ค่อนทัว (Contour) ว่ามีความต่างขนาดและความแตกต่างมากเท่าไร หรือใช้ขนาดความกว้างและยาวมาดูว่ามีขนาดเกินเท่าไหร่จะตัดให้เป็นรูปภาพ นอกจากนี้ออกแบบการหมุนโดยสร้างค่อนทัว (Contour) บรรทัดและวัดความเอียงของแต่ละบรรทัดว่าเอียงเท่าไรจากนั้นจึงหมุนกลับในองศาตรงข้าม

3.2.2.1 การคัดเลือกข้อมูล

เนื่องจากข้อมูลหนังสือที่ทางคณะผู้จัดทำนำมาใช้นั้นมีความหลากหลาย และบางเล่มมีหน้าหนังสือพื้นหลังสีที่ส่งผลให้การทำ OCR ได้ผลลัพธ์ที่ไม่ดีด้วยนั้นจึงทำการคัดเลือกข้อมูลหนังสือเพื่อลดโอกาสที่จะเกิดคำผิดที่จะเกิดขึ้น โดยการทำการคัดเลือกข้อมูลที่เป็นหน้าสีน้ำเงิน เราได้นำค่าความถี่ของหน้าที่มีพื้นหลังสีมาเพื่อตัดภายนอกต่างๆ ให้เหลือแค่หน้าที่มีพื้นหลังสี หลายสีนั้นจะมีค่าความถี่กระจายอยู่ทั่วไปในขณะที่เราเป็นหน้าที่มีพื้นหลังสีเดียวมีค่าความถี่ในช่วงนั้นสูงตั้งรูป 3.3



รูปที่ 3.2: ภาพแสดงความถี่ของภาพพื้นหลังสีและภาพพื้นหลังขาวดำ

ขั้นตอนหลักการทำงานของการคัดเลือกข้อมูล

1. รับไฟล์รูปภาพมาแปลงเป็น Gray Scale และทำการปรับตัวแปรที่เป็นรูปภาพแบบ Gray Scale ให้เป็นอาร์เรย์มิติเดียว
2. ทำการนับความถี่ของค่าสีว่าแต่ละค่าสีนั้นมีจำนวนเท่าไหร่
3. นำค่าความถี่ของค่าสีที่มีความถี่สูงสุดมาคำนวณว่ามีค่ามากกว่า 10 เปอร์เซ็นต์ของจำนวนพิกเซลทั้งหมดหรือไม่ ถ้าน้อยกว่าให้ข้ามการทำ OCR หน้าไป

```
def skipPage(image):
    #skip image or difficult bg
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    flat = gray.flatten().tolist()
    (unique, counts) = np.unique(flat, return_counts=True)
    if max(counts)/len(flat)*100 < 10:
        return True
    return False
```

รูปที่ 3.3: ภาพแสดงขั้นตอนการคัดเลือกข้อมูล

3.2.2.2 การหมุนรูป

ในส่วนของการหมุนรูปนั้นจัดทำขึ้นมาเพื่อแก้ไขข้อมูลรูปภาพที่เกิดความผิดพลาดจากการสแกนส่งผลให้รูปที่ได้นั้นมีความเอียงและทำให้เกิดความผิดพลาดในการทำ OCR เพื่อข้อมูลภาพให้อยู่ในรูปแบบที่เหมาะสมกับการทำ OCR มากที่สุด โดยจะมีวิธีการหาค่าองศาของแต่ละประ年之久คัดวิธีการนำจุด 2 ที่อยู่ในแนวระนาบเดียวกันมาทำการหา Arctan เพื่อหาองศาที่ทำให้บรรทัดนั้นตรง และนำค่าองศาแต่ละบรรทัดที่อยู่ในย่อหน้าเดียวกันมาหางศาสตร์เพื่อที่จะใช้การหมุนทั้งย่อหน้าให้ตรง

ขั้นตอนการหมุน

- เริ่มจากนำรูปภาพมาทำเป็นสองส่วนคือการทำขยายภาพ (Dilation) และกร่อนภาพ (Erosion) โดยที่รูปภาพที่ถูกขยายจะได้รูปที่มีการจับกลุ่มบรรทัดของอ่าน้ำ และรูปที่ถูกกร่อนจะได้รูปภาพที่มีการแยกบรรทัดข้อความกันอย่างชัดเจน และนำไปหาคุณทัว (Contour) โดยที่รูปภาพของการจับกลุ่มบรรทัดจะได้ผลลัพธ์ดังรูปภาพที่ 3.5 ทางด้านซ้าย และการแยกบรรทัดจะได้ผลลัพธ์ดังรูปภาพที่ 3.5 ทางด้านขวา

```
#use dilate for create externalCnt | erode for
kernalDilate = cv2.getStructuringElement(cv2.MORPH_RECT,(11,5))
kernalErode = cv2.getStructuringElement(cv2.MORPH_RECT,(21,3))
dilate = cv2.dilate(imageOCR,kernalDilate,iterations=3)
erode = cv2.erode(dilate,kernalErode,iterations=3)
h,w,c = image.shape

externalCnts = findContours(dilate, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
internalCnts = findContours(erode, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
```

รูปที่ 3.4: ภาพแสดงการทำกร่อนภาพ (Erosion) และการขยายภาพ (Dilation)



รูปที่ 3.5: ภาพแสดงการเปรียบเทียบการทำกร่อนภาพ (Erosion) และการขยายภาพ (Dilation)

- ทำการแยกรูปภาพและบรรทัดข้อความ ภายในคุณทัว (Contour) ที่ถูกกร่อน โดยการกำหนดค่าความสูงที่สุดและต่ำที่สุดไว้ ค่าอัตราส่วนระหว่างความสูงและความกว้าง โดยค่าที่กำหนดไว้เราได้มาจากการทำ Page dewarp ของ Matt Zucker [20] และนำมาระบุเงื่อนไขแล้วแยกเป็นกลุ่มของข้อความและรูปภาพดังรูปภาพที่ 3.6

```
## Range of values
TEXT_MIN_WIDTH = 15      # min reduced px width of detected text contour
TEXT_MIN_HEIGHT = 2       # min reduced px height of detected text contour
TEXT_MIN_ASPECT = 1.5     # filter out text contours below this w/h ratio
TEXT_MAX_THICKNESS = 10   # max reduced px thickness of detected text contour
TEXT_MAX_HEIGHT = 100
```

รูปที่ 3.6: ภาพแสดงเกณฑ์การวัดบรรทัดของตัวอักษร

```
#find height and width to check that cnt is picture or text
height,width,xlow,ylow = findDistance(box,w,h)
if height < TEXT_MIN_HEIGHT or width < TEXT_MIN_WIDTH or width < TEXT_MIN_ASPECT*height or height > TEXT_MAX_HEIGHT:
    if height > TEXT_MAX_HEIGHT:
        internalNotText.append([(x+5,y+5),(x+w-5,y+5),(x+w-5,y+h-5),(x+5,y+h-5)])
        cv2.rectangle(imgNotText, (x+10, y+10), (x + w-10, y + h-10), (255,0,12),3)
    continue
internalText.append([box,xlow,ylow])
```

รูปที่ 3.7: ภาพแสดงการคัดแยกค่อนทัว (Contour) ที่ไม่ใช้ตัวอักษร

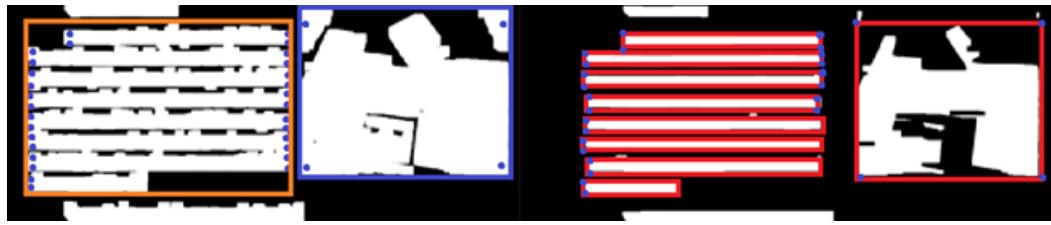
3. ทำการหา Mask เพื่อใช้ในการลบรูปภาพและคำนวณองศาของข้อความแต่ละบรรทัดภายในค่อนทัว (Contour) ของภาพที่ถูกขยาย (Dilation) นั้น โดยการลบรูปภาพจะทำโดยการระบุรูปภาพอยู่ภายในค่อนทัว (Contour) ใหม่ เพื่อจะนำมาเป็น mask สำหรับการลบรูปภาพออกด้วยวิธีการดูว่ามุมทั้ง 4 มุมของค่อนทัว (Contour) ที่ถูกแยกเป็นรูปภาพนั้นอยู่ภายในค่อนทัว (Contour) ภาพที่ถูกขยาย (Dilation) ครบทั้งสี่มุมหรือเปล่า ถ้าใช้แสดงว่าค่อนทัว (Contour) ภาพที่ถูกขยาย (Dilation) คือ mask ของรูปภาพที่จะต้องถูกลบ ให้สร้างค่อนทัว (Contour) ภาพที่ถูกขยาย (Dilation) นั้นลงไปบนรูปแต่ถ้าไม่ใช่ก็จะนำไปหาองศาของแต่บรรทัดข้อความและนำมาเฉลี่ยเป็นองศาที่ต้องหมุนสำหรับค่อนทัว (Contour) ภาพที่ถูกขยาย (Dilation) นั้น

```
for exCnt in externalCnts:
    val=False
    boundaryBox = cv2.boundingRect(exCnt)
    rect = cv2.minAreaRect(exCnt)
    box = np.int8(cv2.boxPoints(rect))
    polygon = Polygon(box)
    for index, notText in enumerate(internalNotText):
        p0 = Point(notText[0])
        p1 = Point(notText[1])
        p2 = Point(notText[2])
        p3 = Point(notText[3])
        val = p1.within(polygon) and p2.within(polygon) and p3.within(polygon) and p0.within(polygon)
        if val:
            cv2.drawContours(imgNotText,[box],-1,(255,255,255),-1)
            internalNotText.pop(internalNotText.index(notText))
            break
```

รูปที่ 3.8: ภาพแสดงการทำ Mask ในส่วนที่ไม่ใช้ตัวอักษร

```
if not val:
    externalBox.append(box)
    externalCntBox.append(boundaryBox)
    angle = []
    avgAngle=0
if len(internalText) == 0:
    angleBox.append(avgAngle)
for indexText, textBox in enumerate(internalText):
    p10 = Point(textBox[0][0])
    p11 = Point(textBox[0][1])
    p12 = Point(textBox[0][2])
    p13 = Point(textBox[0][3])
    val1 = p11.within(polygon) and p12.within(polygon) and p13.within(polygon) and p10.within(polygon)
    if val1:
        angleSingle = findAngle(textBox[1][0],textBox[2][0],textBox[1][1],textBox[2][1])
        angle.append(angleSingle)
        cv2.arroWnedLine(picture,(textBox[1][0],textBox[2][0]),(textBox[1][1],textBox[2][1]),(0,255,0),2)
    if len(internalText)-1 == indexText:
        if(len(angle) != 0):
            avgAngle = findAverageAngle(angle)
            angleBox.append(avgAngle)
```

รูปที่ 3.9: ภาพแสดงการคัดตัวอักษรเพื่อนำไปหาองศาในการหมุน



รูปที่ 3.10: ภาพแสดงจุดของค่อนทัว (Contour) เล็กในค่อนทัว (Contour) ใหญ่

- นำ Mask ที่เป็นรูปภาพนำกลับออกโดยเมื่อได้ mask มา ก็จะนำไปลบออกจากรูปใหญ่ในฟังก์ชัน removePicture ในรูปที่ 3.11 ซึ่งจะนำเอาก่อนทัว (Contour) ภาพที่ถูกขยาย (Dilation) ที่เป็นตัวอักษร (กรอบสีเขียว) มาลบออกจาก mask เดิม(กรอบสีแดง)ที่ได้สร้างไว้เพื่อกันการลบพื้นที่เป็นตัวอักษรดัง 3.12 ด้วยการใช้ฟังก์ชัน drawContour ก่อนจะนำ mask ที่ได้มาลบรูปออกทำให้ภาพในแต่ละหน้าหายไป เป็นผลลัพธ์ออกมาดัง 3.13

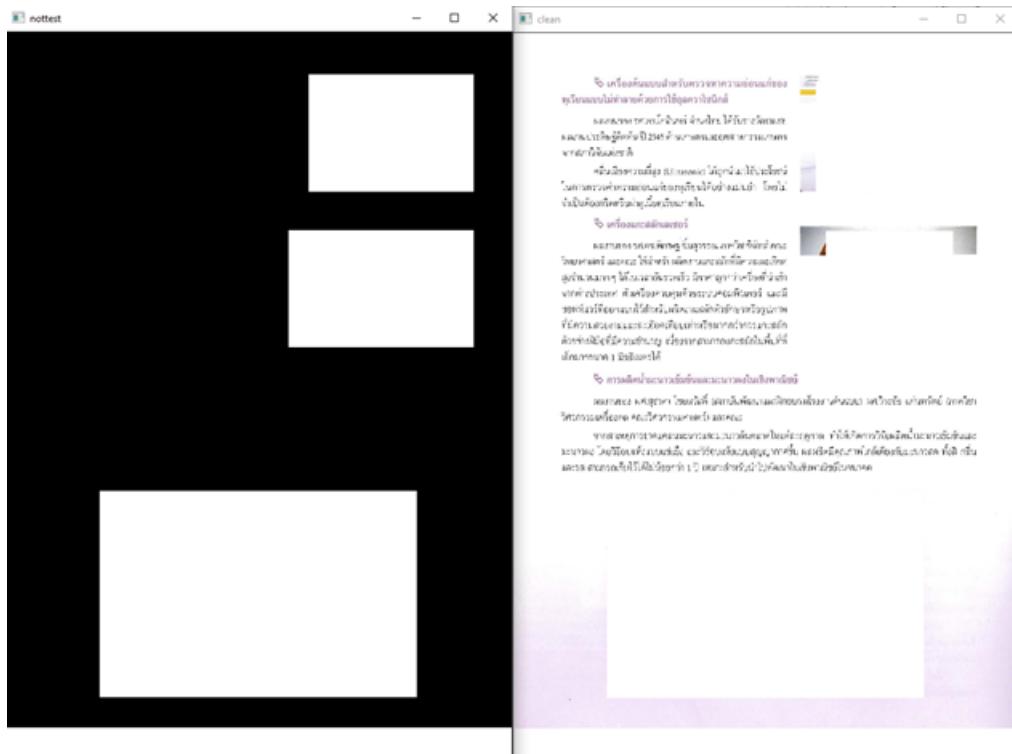
```
def removePicture(externalBox, imgNotText, image):
    for exBox in externalBox:
        cv2.drawContours(imgNotText,[exBox],-1,(0,0,0),-1)

    imgNotText = cv2.cvtColor(imgNotText, cv2.COLOR_BGR2GRAY)
    deleteImage = findContours(imgNotText, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)

    for delIm in deleteImage:
        cv2.drawContours(image,[delIm],-1,(255,255,255),-1)

    return image
```

รูปที่ 3.11: ภาพแสดงฟังก์ชันการลบรูปภาพออกจากหนังสือ



รูปที่ 3.12: ภาพแสดงการสร้าง Mask เพื่อลบรูปภาพ



รูปที่ 3.13: ภาพแสดงการสร้าง Mask โดยวิธีที่ตัวอักษร

5. การหาค่าเฉลี่ยองศาในแต่ละรูปภาพที่ถูกขยาย (Dilation) ทำโดยการนำค่อนทัว (Contour) ของตัวอักษรแต่ละบรรทัดที่ได้มาเข้าสู่กระบวนการรัดมุมและหมุนภาพ โดยการนำจุดสี่จุดของค่อนทัว (Contour) เล็กมาเฉลี่ย องศาในการหมุน เนื่องจากค่าจุดที่ได้จากการหา Minimum Area Rectangle นั้นอาจจะมีบางครั้งที่ค่าจุดที่ส่งมาไม่ได้เริ่มจากซ้ายบน ดังนั้นจุดและเว้นที่หามาได้นั้นอาจจะทำให้ได้องศาในแนวตั้งมากได้ จึงต้องทำการกรองว่าองศาที่ได้ว่าตั้งจากหรือไม่ตั้งจากถ้าเป็นองศาสั้งจากให้ทำการอธิบายจากแกน 90 องศา และถ้าไม่ใช้ก็เทียบจากแกนแนวโน้ม 0 องศาหรือ 180 องศา

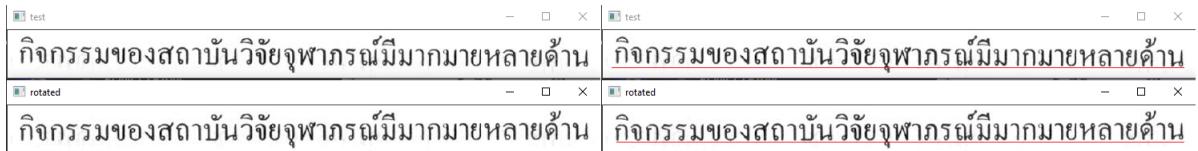
```

def findAngle(x1,y1,x2,y2):
    angleCal = math.degrees(math.atan2(y2-y1,x2-x1))
    return angleCal

def findAverageAngle(angle):
    for key,val in enumerate(angle):
        if val > 135.0:
            angle[key] = 180-val
        elif 135> val > 45:
            angle[key]=90-val
        elif val<-135:
            angle[key]=180+val
        elif val<-45:
            angle[key]=90+val
        else:
            angle[key]=val
    angleAvg = (sum(angle)/len(angle))
    return angleAvg

```

รูปที่ 3.14: ภาพแสดงการหาองศาในการหมุน



รูปที่ 3.15: ภาพแสดงผลลัพธ์ที่ได้จากการหมุน

3.2.2.3 การลบพื้นหลัง

ในส่วนของการลบพื้นหลังนั้นเป็นฟังก์ชันที่สร้างขึ้นมาเพื่อลบรายละเอียดที่ไม่รูปหรือลักษณะ ซึ่งมีผลต่อการแยกแยะและจัดจำตัวอักษร ซึ่งเป็นฟังก์ชันทางเลือกที่ทำขึ้นมาแก้ไขผลลัพธ์ที่ได้จากการจัดการลบรูปภาพ

ขั้นตอนการลบพื้นหลัง

1. แปลงรูปภาพสีให้กลายเป็นขาวดำจะทำให้ขั้นสีเหลือเพียงขั้นเดียวจากเดิมที่มี 3 ขั้นสีเป็น RGB เหลือเป็นค่า Gray scale ที่มีค่า 0-255 โดยค่าเข้าใกล้ 0 คือสีดำและเข้าใกล้ 255 คือสีขาว
2. นำรูปภาพมาผ่านกระบวนการขยาย (Dilation) โดยกำหนดเป็นสี่เหลี่ยมขนาด 5x5 จะได้ผลลัพธ์ดังรูปภาพที่ 3.19
3. นำรูปภาพ gray scale มาหารกับค่าที่ถูกขยาย (Dilation) มาโดยใช้ scale อูฐที่ 0 – 255 จะทำให้พื้นหลัง หายไปเนื่องจากจะเห็นได้ว่าถ้าส่วนไหนของรูปภาพถูกขยาย (Dilation) ออกไปจะไม่ถูกลบเมื่อนำค่าพิกเซลล์(pixel) นั้นมาหารแต่ถ้าพื้นที่นั้นไม่ถูกขยาย (Dilation) ออกไปจะทำให้เมื่อนำค่าพิกเซลล์(pixel) มาหารจะทำให้พิกเซลล์(pixel) นั้นกลับเป็นสีขาวดังรูปภาพที่ 3.20
4. นำรูปภาพที่ไม่มีพื้นหลัง ไปทำ threshold ให้รูปเหลือเพียงสีขาวกับสีดำ โดยเราจะทำเป็น THRESH_BINARY_INV ที่ทำให้ตัวอักษรเป็นสีขาวและพื้นหลังเป็นสีดำเพื่อนำไปใช้ในการหาcontour (Contour) ต่อ

```
def removeBG(picture):
    gray = cv2.cvtColor(picture, cv2.COLOR_BGR2GRAY)
    # apply morphology
    kernel = cv2.getStructuringElement(cv2.MORPH_RECT, (5, 5))
    morph = cv2.morphologyEx(gray, cv2.MORPH_DILATE, kernel)
    # divide gray by morphology image
    division = cv2.divide(gray, morph, scale=255)
    # threshold
    return cv2.threshold(division, 0, 255, cv2.THRESH_OTSU + cv2.THRESH_BINARY_INV)[1]
```

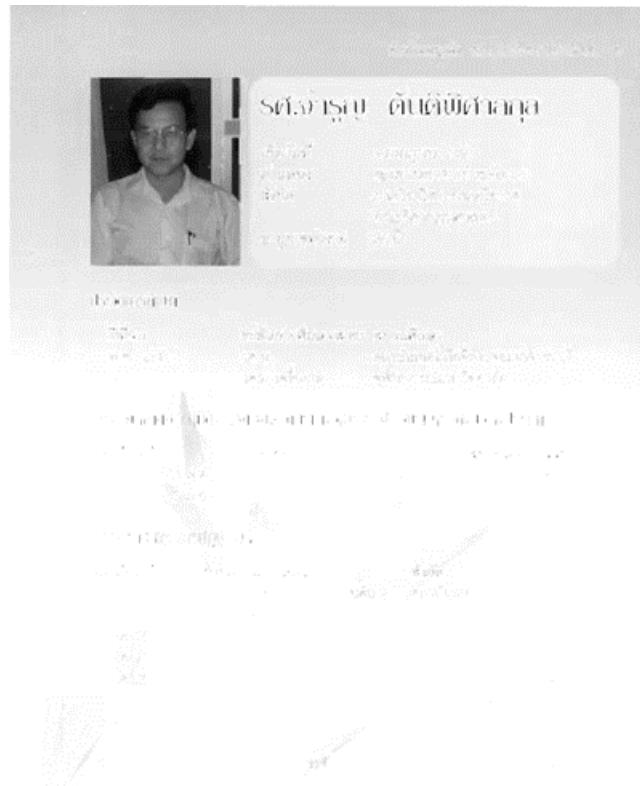
รูปที่ 3.16: ภาพแสดงขั้นตอนในการลบพื้นหลังสี



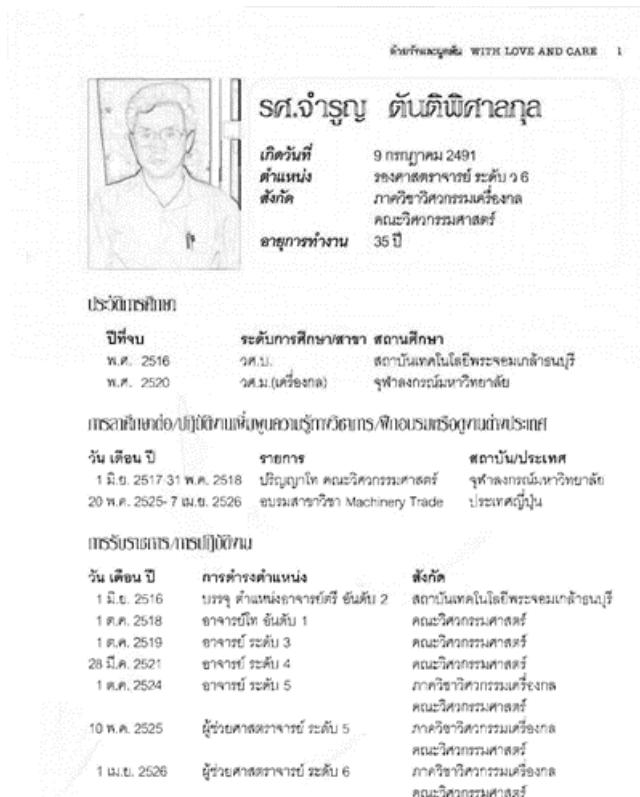
ຮູບທີ່ 3.17: ຮູ່ປາພສຶກອ່ອນຄຸນນຳເຂົາມາທຳກາຣລົບພື້ນຫລັງ



ຮູບທີ່ 3.18: ຮູ່ປາພກການແປລ່ງພາພສຶກເປັນ gray scale



รูปที่ 3.19: รูปภาพที่ผ่านการทำการขยาย โดยใช้รูปแบบสี่เหลี่ยมขนาด 5x5

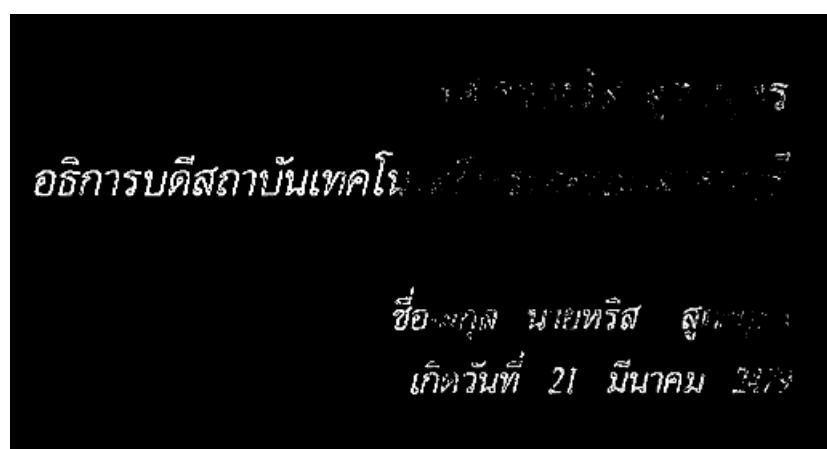


รูปที่ 3.20: รูปภาพที่ผ่านการลบพื้นหลัง

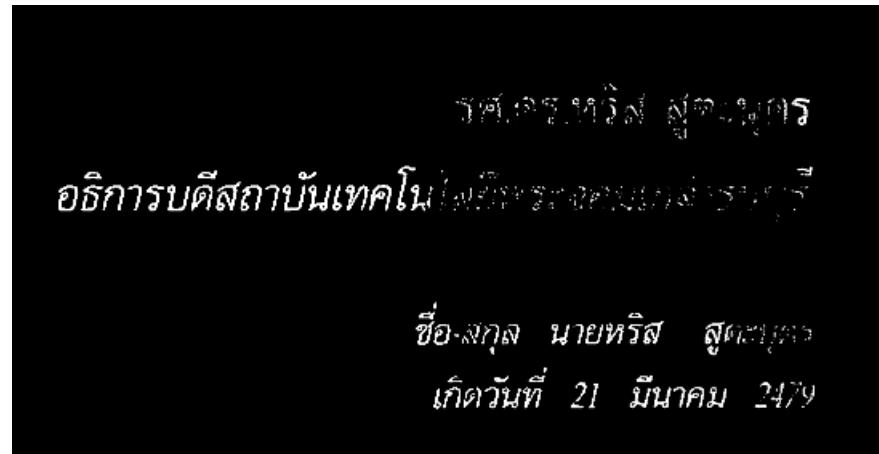


ຮູບທີ 3.21: ຮູບພາບທີ່ໄໜ້ນທໍາການ threshold ແບບ THRESH_BINARY_INV

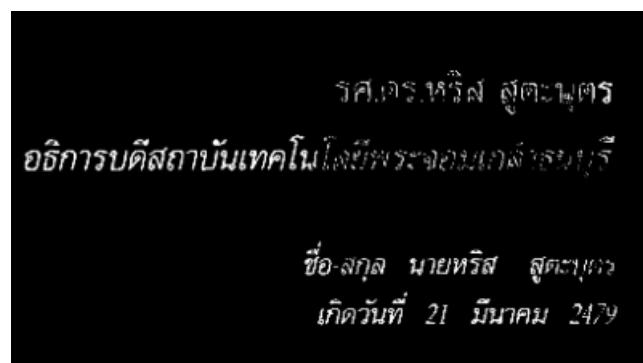
ໃນກາຣລັບພື້ນທັງໝາຍັງຜູ້ຈັດທໍາໄດ້ທົດລອງໃໝ່ຂາດໄຊ່ສ່ວນຂອງ kernel ໃນຂາດດ່າງໆ ໂດຍເປີຍເບເທີບ 3×3 , 5×5 , 7×7 ແລະ 15×15 ຈຶ່ງຈາກກາຣທົດລອງພບວ່າຂາດ kernel 3×3 ນັ້ນຈະທໍາໄດ້ຕັ້ງອັກຊ່ຣທີ່ໄດ້ມີຮູ່ຮຽງກລາງ ດັ່ງກາພທີ່ [3.22](#) ແລະ ພອເພີ່ມຂາດເປັນ 5×5 7×7 ແລະ 15×15 ຈະເຫັນໄດ້ວ່າຍຶ່ງໃໝ່kernel ຂາດໃໝ່ ຈະທໍາໄດ້ຕັ້ງອັກຊ່ຣດັ່ງນີ້ ແລະ [3.26](#) ຈຶ່ງຕັ້ງ kernel ຂາດ 3×3 ແລະ 25×25 ທັ້ງໄປ ເໜື້ອ 5×5 , 7×7 ແລະ 15×15 ຈຶ່ງຈາກຜລັບຮົບທີ່ໄດ້ເຫັນຢ່າງຍິ່ງຂຶ້ນ ແລະ [3.23](#) [3.24](#) ນັ້ນໄໝ່ຕ່າງກັນມາເຮົາຈຶ່ງເລືອກໃໝ່ຂາດ 5×5 ກີ່ເພີ່ມພອ ສ່ວນ kernel 15×15 ນັ້ນ ມີຂາດທີ່ຄ່ອນໜ້າງໃໝ່ ທໍາໄຫ້ຄາດວ່າອາຈະທໍາໄດ້ເກີດກາຣຕັດປະໂຍຄມີຄວາມຜິດພາດມາຍິ່ງຂຶ້ນດັ່ງກາພ [3.27](#) ດັ່ງນັ້ນຈາກກາຣທົດລອງກັບຮູ່ປັບປຸງຈຶ່ງໃໝ່ຜູ້ຈັດທໍາເລືອກໃໝ່ຂາດ 5×5



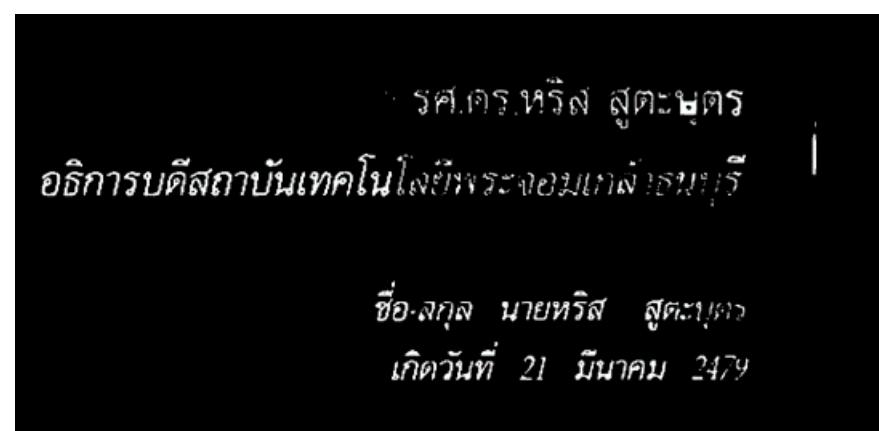
ຮູບທີ 3.22: ຮູບພາບແສດກກາຣໃໝ່ກາຣລັບພື້ນທັງດ້ວຍ kernel ຂາດ 3×3



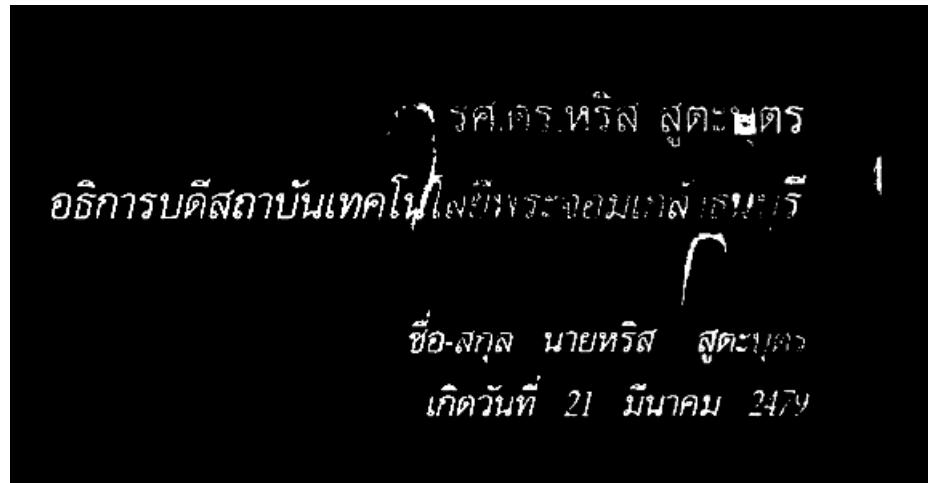
รูปที่ 3.23: รูปภาพแสดงการใช้การลบพื้นหลังด้วย kernel ขนาด 5*5



รูปที่ 3.24: รูปภาพแสดงการใช้การลบพื้นหลังด้วย kernel ขนาด 7*7



รูปที่ 3.25: รูปภาพแสดงการใช้การลบพื้นหลังด้วย kernel ขนาด 15*15



รูปที่ 3.26: รูปภาพแสดงการใช้การลับพื้นหลังด้วย kernel ขนาด 25×25

▪ Sektorisasi-KMUTT	96
▪ Sektorisasi-MULTI Membership	98
▪ GearUp Publishing	98
▪ กิจกรรมนักศึกษาและกิจกรรม	99
ผู้นำและผู้บริหารระดับสูงของมหาวิทยาลัย	102
▪ ผู้นำสถาบันและผู้บริหารระดับสูงของมหาวิทยาลัย	103
▪ ผู้บริหารและผู้อำนวยการฝ่ายต่างๆ ของมหาวิทยาลัย 1 Flagships (Block 1, 2 และ 3)	111
▪ ภาควิชาและศูนย์ฝึกอบรม	115
▪ ผู้บริหารและผู้อำนวยการฝ่ายต่างๆ ของมหาวิทยาลัย	116
▪ ผู้บริหารและผู้อำนวยการฝ่ายต่างๆ ของมหาวิทยาลัย	123
▪ ผู้บริหารและผู้อำนวยการฝ่ายต่างๆ ของมหาวิทยาลัย	126
▪ Management System Development (MSD)	128
▪ ผู้บริหารและผู้อำนวยการฝ่ายต่างๆ	129
▪ ผู้บริหารและผู้อำนวยการฝ่ายบริหารและวางแผน (行政及戰略管理處)	133
▪ ผู้บริหารและผู้อำนวยการฝ่ายอบรม (Training Roadmap)	136
▪ ผู้บริหารและผู้อำนวยการฝ่ายบริการ	137
▪ ผู้บริหารและผู้อำนวยการฝ่ายบริการสนับสนุนและบริการนักศึกษา	139
▪ ผู้บริหารและผู้อำนวยการฝ่ายบริการ	141
▪ ผู้บริหารและผู้อำนวยการฝ่ายบริการ	142
▪ ผู้บริหารและผู้อำนวยการฝ่ายบริการ	143
ผู้นำและผู้บริหารระดับสูงของมหาวิทยาลัยในอดีต	144
▪ ผู้บริหารและผู้อำนวยการอดีต	145
▪ ผู้บริหารและผู้อำนวยการอดีต	145
▪ ผู้บริหารและผู้อำนวยการอดีต	149
▪ ผู้บริหารและผู้อำนวยการอดีต	154
▪ ผู้บริหารและผู้อำนวยการอดีต	158
▪ ผู้บริหารและผู้อำนวยการอดีต	161
ผู้นำและผู้บริหารระดับสูงของมหาวิทยาลัยในปัจจุบัน	165
▪ ผู้บริหารและผู้อำนวยการปัจจุบัน	166
▪ ผู้บริหารและผู้อำนวยการปัจจุบัน	166
▪ ผู้บริหารและผู้อำนวยการปัจจุบัน	167
▪ ผู้บริหารและผู้อำนวยการปัจจุบัน	168
▪ ผู้บริหารและผู้อำนวยการปัจจุบัน	169
▪ ผู้บริหารและผู้อำนวยการปัจจุบัน	170
ผู้นำและผู้บริหารระดับสูงของมหาวิทยาลัย	171
ผู้บริหารและผู้อำนวยการฝ่ายต่างๆ	177
▪ ผู้บริหารและผู้อำนวยการ	177
▪ ผู้บริหารและผู้อำนวยการ	178
▪ ผู้บริหารและผู้อำนวยการ	179
▪ ผู้บริหารและผู้อำนวยการ	180
▪ ผู้บริหารและผู้อำนวยการ	182
▪ ผู้บริหารและผู้อำนวยการ	184
▪ ผู้บริหารและผู้อำนวยการ	184
▪ ผู้บริหารและผู้อำนวยการ	185
▪ ผู้บริหารและผู้อำนวยการ	186
▪ ผู้บริหารและผู้อำนวยการ	187
▪ Premises August 2552	190

รูปที่ 3.27: รูปภาพแสดงการใช้การลับพื้นหลังด้วย kernel ขนาด 15×15 ที่กินพื้นที่รอบข้าง

3.2.3 การแปลงข้อมูลรูปภาพให้อยู่ในรูปแบบดิจิทัล

สำหรับการแปลงข้อมูลรูปภาพให้อยู่ในรูปแบบดิจิทัลจะใช้ Tesseract OCR โดยจะใช้รูปภาพที่ผ่านกระบวนการเตรียมข้อมูลรูปภาพและประยุกต์ที่แปลงอุปกรณ์ได้จัดเก็บไว้ใช้งานต่อไป

3.2.4 การเตรียมข้อมูลตัวอักษร

สำหรับการเตรียมข้อมูลตัวอักษรจัดทำเพื่อให้ข้อมูลเหมาะสมสำหรับการค้นหาเพื่อสร้างคำสำคัญของหนังสือ โดยเราจะนำหลักการของการทำ Tokenizer มาใช้ ซึ่งหลักการนี้จะมีรูปแบบแตกต่างกันออกเป็นแต่ละภาษาที่จะใช้ โดยภาษาที่เราจะใช้มีภาษาไทย และภาษาอังกฤษ อันดับแรกของกระบวนการทำคือการตัดคำ จากรูปประโยคซึ่งระหว่างแต่ละภาษาจะมีรูปแบบต่างกัน เราจึงต้องนำอัลกอริทึมที่เรียกว่า Deepcut เข้ามาช่วยในการตัดแบ่งคำ หลังจากการตัดคำเราต้องจัด การกับคำที่สื่อความหมายใกล้เคียงกันและรูปแบบการเขียนที่แตกต่างกันอย่าง รูปแบบตัวใหญ่ตัวเล็ก คำที่อยู่ในรูปนาม กวิยา กรรมแต่สื่อความหมายถึงสิ่งเดียวกัน การจัดการคำที่ไม่มีความหมายแต่เป็นคำสำคัญรูปแบบการพูดเท่านั้น การแก้คิดสำคัญที่สแกนมาผิดพลาดให้กลับเป็นคำที่ถูกต้อง ซึ่งการเช็คคำในส่วนนี้จะ ไม่สามารถเช็คคำเฉพาะได้อย่างเช่น ชื่อคน เราจึงต้องมีการเช็คคำเฉพาะอีกรอบ แล้วสุดท้ายผู้ใช้งานจะมีสิทธิในการเช็คคำอีกรอบในส่วนนี้เพื่อลดการแก้คิดที่ไม่สามารถแก้ได้ทั้งหมด

ขั้นตอนการตัดแบ่งคำและการแก้ไขคิด

1. นำแต่ละประโยคในหน้านั้นมาทำการตัดคำโดยใช้อัลกอริทึม Deepcut
2. การจัดการตัวอักษรที่ไม่ต้องการใช้และถือเป็นตัวอักษรที่ผิดพลาดที่เกิดจากการทำ OCR อย่างเช่น รูปแบบตัวอักษราติน ตัวอักษรที่เป็นรูปแบบสัญลักษณ์พิเศษ การจัดการรูปตัวอักษรตัวเล็กตัวใหญ่
3. การค้นหาคำเฉพาะโดยการคุยกับบริบทของคำ กรณีภาษาอังกฤษจะเป็นคำที่มีตัวอักษรแรกเป็นตัวใหญ่ และในกรณีของการใช้จุด “.” ของทั้งสองภาษาจะถือว่าเป็นคำเฉพาะ เช่นเดียวกัน
4. นำคำที่ถูกแยกออกจากกรุปประโยคมาแก้ไขจากคำที่ผิดให้เป็นคำที่ถูกต้อง ซึ่งก่อนแก้ไขต้องระบุว่าคำที่จะแก้ไขเป็นภาษาชนิดใด โดยภาษาไทยจะมีวิธีการตรวจสอบโดยที่คำนั้นจะต้องมีทูกตัวอักษรเป็นภาษาไทยทั้งหมดจึงจะนับว่าเป็นภาษาไทย นอกจากนี้หากนั้นจะนับเป็นภาษาอังกฤษทั้งหมด
5. นำคำที่ถูกแก้คำแล้ว มาแก้คำเฉพาะอีกรอบนึงเนื่องจากจะมีบางคำที่ไม่สามารถแก้ได้อย่าง ชื่อคนสำคัญ ชื่อมหาวิทยาลัย โดยใช้หลักการ Minimum edit distance
6. นำผลลัพธ์ที่ได้ส่งไปยังขั้นตอนให้ผู้ใช้ตรวจสอบเพื่อเช็คคำผิดอีกรอบ
7. นำผลลัพธ์ที่ผู้ใช้งานตรวจสอบและแก้ไขมาทำการลบคำที่เป็น stop word
8. นำผลลัพธ์ที่ได้จากการทำข้อที่ 7 เข้าสู่ระบบ

3.2.5 การคำนวณค่าความสำคัญของคำตามที่ปรากฏในเอกสาร

หลังจากการเตรียมข้อมูลตัวอักษร จะนำผลลัพธ์ที่ได้จะนำเข้ากระบวนการคำนวณค่าความสำคัญของคำตามที่ปรากฏในเอกสาร โดยอัลกอริทึม TF-IDF ซึ่งจะถูกเก็บในรูปแบบของ Inverted index ซึ่งในกรณีหากที่กล่าวจะเป็นรูปแบบการสร้างคำสำคัญ ของระบบที่สร้างเอง แต่จะมีในส่วนของการสร้างคำสำคัญ ที่ผู้ใช้งานสามารถเพิ่มเข้าไปในระบบเองได้

ขั้นตอนการคำนวณค่าความสำคัญของคำตามที่ปรากฏในเอกสาร

1. นำคำที่ได้จากการเตรียมข้อมูลตัวอักษร มาทำการคำนวณผ่านอัลกอริทึม TF-IDF เพื่อสร้างค่าคะแนนขึ้นมา

2. นำคำสำคัญผลลัพธ์ที่ได้มาให้ผู้ใช้ตรวจสอบใน 10 อันดับ และสามารถ เพิ่ม ลด แก้ไขได้ โดยที่ระบบจะทำการปรับแก้คำแนะนำของคำสำคัญที่ผู้ใช้งานจัดการในหนังสือ สามารถถูกค้นหาได้ง่ายกว่าคำสำคัญที่ระบบเป็นสร้างขึ้นมาเอง

3.2.5.1 การอัพเดทคำแนะนำ TF-IDF

เนื่องจากการที่มีหนังสือเพิ่มเข้ามาในระบบจะทำให้ผลลัพธ์คำแนะนำ TF-IDF เปลี่ยนทั้งระบบจึงทำให้จำเป็นต้องมีการคำนวณใหม่เพื่อความแม่นยำในการค้นหาแต่เนื่องจาก ถ้าระบบมีหนังสือจำนวนมากยิ่งขึ้นทางผู้จัดทำอาจปั๊บเปลี่ยนการอัพเดทคำแนะนำเป็น 1 ครั้งต่อวันโดยที่จะคำนวณคำแนะนำในช่วงเวลาถูกต้องคืนเพื่อไม่ให้ส่งผลกระทบกับผู้ใช้งาน และในส่วนการเพิ่มข้อมูลเข้ามาใหม่คำที่อยู่ภายใต้หนังสือนั้นจะถูกคำนวณและอัพเดตคำใหม่ทันทีส่วนคำอื่นนั้นจะต้องรอเวลาที่กำหนดเพื่อที่จะอัพเดทคำ TF IDF และ TF-IDF

3.2.6 การค้นหา

ในส่วนของระบบการค้นหานั้น จะมีระบบการกรองไว้ให้ผู้ใช้สามารถเลือกได้ว่าจะค้นหาหนังสือในหัวข้อไหน ซึ่งสามารถกรองหัวข้อได้มากกว่า 1 ครั้ง ต่อการค้นหา 1 รอบ โดยระบบการค้นหาจะถูกแบ่งออกเป็นสองส่วน ในส่วนแรกจะเป็นค้นหาในรูปแบบของ Cosine Similarity ซึ่งจะถูกทำ Tokenizer เพื่อให้การค้นหาของผู้ใช้งานตรง กับฐานข้อมูลมากที่สุด แต่จะไม่ทำการแก้ไขคำเพื่อไม่ให้จุดประสงค์ของการค้นหาถูกเปลี่ยนไปจากเดิม หลังจากนั้นการค้นหาในระบบยังสามารถค้นหาคำใกล้เคียงที่สื่อ ความหมายแบบเดียวกันจากคำเดิม เพื่อที่จะได้ผลลัพธ์ที่ครอบคลุมหนังสือมากขึ้น และส่วนที่สองจะเป็นการกรองผลลัพธ์ที่ให้ผู้ใช้สามารถลดผลลัพธ์ที่ไม่เกี่ยวข้อง จำนวนมากให้ลดน้อยลงเพื่อจ่ายต่อการค้นหา เมื่อผู้ใช้งานได้ทำการค้นหาเสร็จสิ้นระบบจะทำการส่งหนังสือที่เกี่ยวข้องกับมาให้ผู้ใช้งาน

ขั้นตอนการทำงานของระบบการค้นหา

1. แบ่งแยกประเภทการค้นหาของผู้ใช้ว่ามีค้นหา และการกรองหัวข้อมูลก่อน
2. นำรูปประโยค หรือคำ จากรูปแบบการค้นหาที่ได้จากผู้ใช้ไปทำการ ตัดคำผ่าน Deepcut และหาคำที่มีความเหมือนจาก Word2Vec
3. จากผลลัพธ์จะได้คำที่สามารถนำไปหาคำแนะนำเพื่อเข้ากระบวนการคิดแบบ Cosine Similarity ดังสมการ 3.1 และสามารถได้หนังสือที่เกี่ยวข้องพร้อมอันดับความเกี่ยวข้องของหนังสือนั้น ๆ
4. จากผลลัพธ์จะได้หนังสือที่เกี่ยวข้อง ดังนั้นหากผู้ใช้ต้องการกรองผลลัพธ์เหล่านี้ระบบจะต้องนำหนังสือไปตรวจสอบว่าหนังสือได้บ้างที่ตรงตามเงื่อนไข
5. นำผลลัพธ์ที่ได้ส่งคืนให้กลับผู้ใช้งาน

$$sim(\vec{q}, \vec{d}) = \frac{\sum_{i=1}^{|v|} q_i d_i}{\sqrt{\sum_{i=1}^{|v|} q_i^2} \times \sqrt{\sum_{i=1}^{|v|} d_i^2}} \quad (3.1)$$

```

const calculateRelevanceScore = (scores, LenTokens) => {
  let sigmaScore = 0
  for (let i = 0; i < scores.length; i += 1) {
    sigmaScore += scores[i]
  }
  let sigmaSqr2Score = 0
  for (let i = 0; i < scores.length; i += 1) {
    sigmaSqr2Score += scores[i] ** 2
  }
  const sqrRootScore = Math.sqrt(sigmaSqr2Score)
  const partUnder = Math.sqrt(LenTokens * sqrRootScore)
  if (partUnder === 0) return 0
  const scoreResult = sigmaScore / partUnder
  return scoreResult
}

```

รูปที่ 3.28: รูปภาพแสดงการคำนวณ Cosine Similarity

3.2.6.1 การทำโมเดล Word2vec

โดยเราได้เตรียมข้อมูลที่จะนำมาสร้างโมเดลเป็นข้อมูลหนังสือกเทศทิพยและรายงานประจำปีของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี จำนวนทั้งหมด 44 เล่ม โดยจะเป็นการนำข้อมูลที่ถูกกระบวนการ OCR และการเตรียมข้อมูลตัวอักษร มาตัดแบ่งคำเว้นช่องว่าง และขั้นบรรทัดใหม่ โดยจะใช้ Library genism ในสร้างโมเดลโดยมีการกำหนด window size เท่ากับ 2 และค่าต้องมีกล่าวถึงมากกว่า 5 ครั้งโดยทำเป็นลักษณะ Skip-gram

ขั้นตอนการสร้างโมเดล

1. เตรียมข้อมูลตัวอักษรโดยการนำแท็ลประโภคเข้ากระบวนการเตรียมข้อมูลตัวอักษร ในการตัดคำในรูปประโยค
2. นำคำไปจัดการตัวอักษรที่เราไม่ต้องการใช้ และตรวจสอบแก้ไขคำผิดที่เกิดขึ้น
3. ทำการเว้นช่องว่างระหว่างคำและขั้นบรรทัดใหม่เมื่อขึ้นประโภคใหม่เพื่อที่เอาไว้ใช้งานสำหรับการสร้างโมเดล
4. ทำการเขียนเพิ่ลงไปในไฟล์ corpus
5. กำหนด parameter
6. เตรียมโมเดล

3.2.7 การจัดการหนังสือดิจิทัล

ในการจัดการข้อมูลหนังสือภายในระบบจะแบ่งทั้ง 3 ส่วนนั้นคือ 1. การเพิ่มหนังสือเข้าสู่ระบบ 2. การแก้ไขหนังสือภายในระบบ 3. การลบหนังสือออกจากระบบ ส่วนที่ 1. ในการเพิ่มหนังสือเข้าสู่ระบบ ผู้ใช้งานจะต้องอัปโหลดไฟล์หนังสือในรูปแบบ PDF และกรอกรายละเอียดของหนังสือเพื่อเข้าสู่กระบวนการแปลงข้อมูลในหนังสือ (Digitization) ต่อไปจะเป็นการเตรียมข้อมูลรูปภาพ ก่อนที่จะนำมาทำการแปลงภาพเป็นตัวอักษรเพื่อที่จะได้ข้อมูลดิจิทัลจากหนังสือที่ผู้ใช้เพิ่มเข้าสู่ระบบหลังจากนั้นจะเป็นการเตรียมข้อมูลตัวอักษร และให้ผู้ใช้งานได้ตรวจสอบคำอ่านนึงครั้งก่อนที่นำคำเหล่านี้ไปผ่านกระบวนการสร้างคำสำคัญ เพื่อหาคำสำคัญในหนังสือโดยให้ผู้ใช้งานได้ตรวจสอบแก้ไข

หรือเพิ่มเติมก่อนจะสั่งสุกดารบาร์เพิ่มหนังสือเข้าสู่ระบบ ในส่วนที่ 2 การแก้ไขหนังสือภายในระบบผู้ใช้งานสามารถค้นหาหนังสือภายในระบบ เพื่อนำมาแก้ไขรายละเอียดที่ผู้ใช้งานกรอกเท่านั้นแต่มาสามารถแก้ไขคำที่ถูกแปลงอ错มาเป็นดิจิทอลีครรังได้ และส่วนสุดท้ายการลบหนังสือในระบบผู้ใช้งานสามารถลบหนังสือภายในระบบได้โดยการค้นหาหนังสือที่ต้องการและกดลบหนังสือนั้นออกจากระบบโดยเมื่อมีการลบหนังสือออกก็จะลบคำที่มีอยู่ในหนังสือออกไปจากระบบเช่นกัน

3.2.8 Login

ผู้ใช้งานสามารถเข้าสู่ระบบเพื่อใช้งานฟังก์ชันต่าง ๆ ภายในระบบโดยเมื่อผู้ใช้งานทำการเข้าสู่ระบบด้วยชื่อผู้ใช้งานและรหัสผ่านแล้วจะได้รับ “Token” เพื่อที่จะใช้สำหรับการยืนยันตัวในการใช้งานฟังก์ชันต่าง ๆ ภายในระบบและผู้ใช้งานจะสามารถออกจากระบบได้

3.3 System requirements

ผู้ใช้งาน

ใช้งานได้บนระบบ web browser

- Google Chrome เวอร์ชัน 84.0 ขึ้นไป
- Microsoft Edge เวอร์ชัน 83.0 ขึ้นไป
- Firefox เวอร์ชัน 75.0 ขึ้นไป

ผู้เชื่อมต่อ

ทางด้านฮาร์ดแวร์

- CPU: Intel or AMD processor with 64-bit โดยที่ต้องมี 4 Core ขึ้นไป
- GPU: NVIDIA 1050ti or higher
- Disk Storage: 10 GB
- RAM: 8GB or higher

ทางด้าน Software แบ่งเป็น 2 ส่วนคือ Python และ JavaScript

1. Python Backend

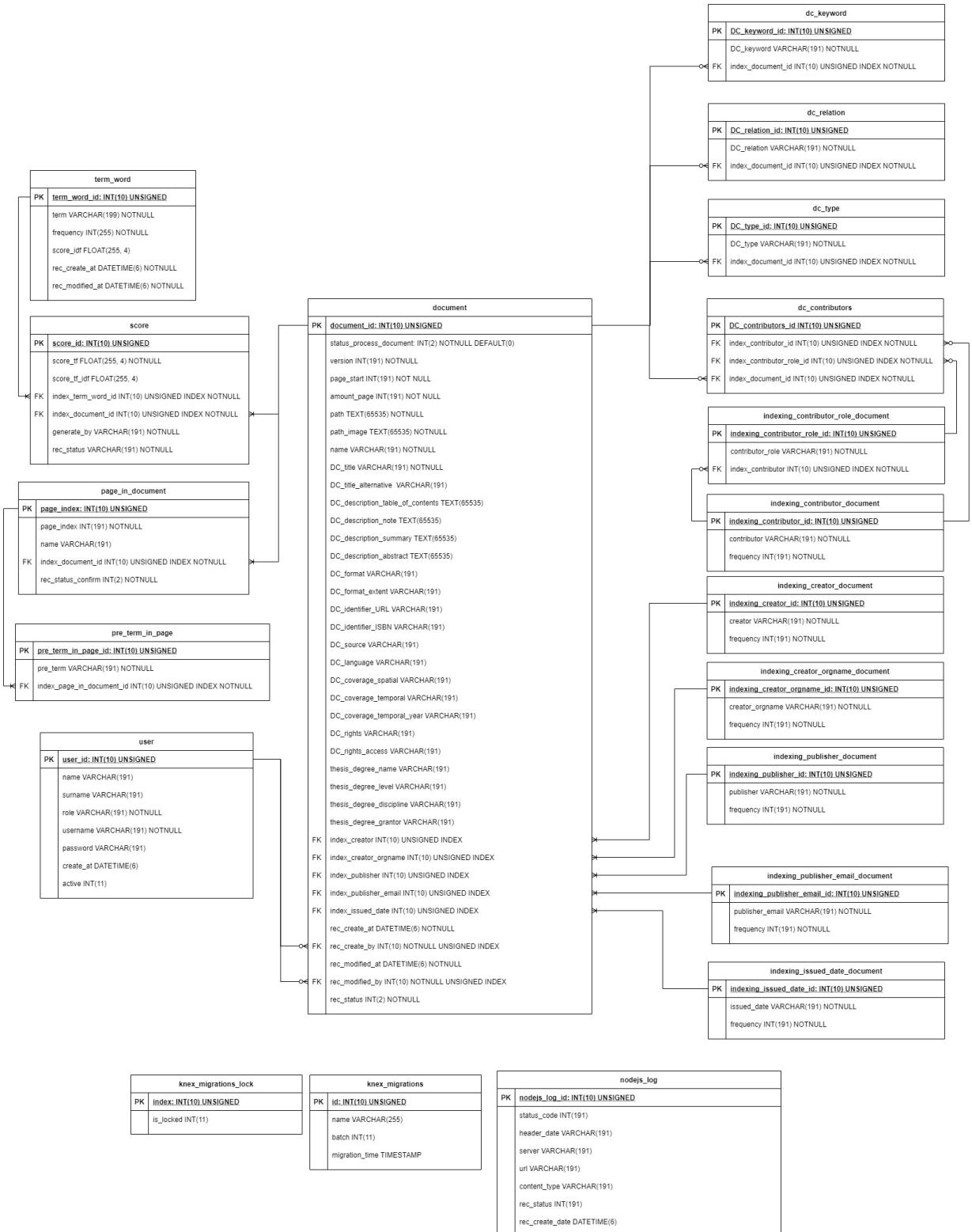
- Python เวอร์ชัน 3.7.5
- Tensorflow เวอร์ชัน 2.3.1
- DeepCut เวอร์ชัน 0.7
- Django เวอร์ชัน 3.1.3
- Djangorestframework เวอร์ชัน 3.12.2
- Django-cors-headers เวอร์ชัน 3.5.0

- PyThaiNLP เวอร์ชัน 2.2.5
- Pyspellchecker เวอร์ชัน 0.5.5
- nltk เวอร์ชัน 3.5.0
- mysqlclient เวอร์ชัน 2.0.1
- pillow เวอร์ชัน 8.0.1
- shapely เวอร์ชัน 1.7.1
- pytesseract เวอร์ชัน 5.0.0 beta
- opencv-python เวอร์ชัน 4.4.0.46
- pdf2image เวอร์ชัน 1.14.0
- scipy เวอร์ชัน 1.5.4

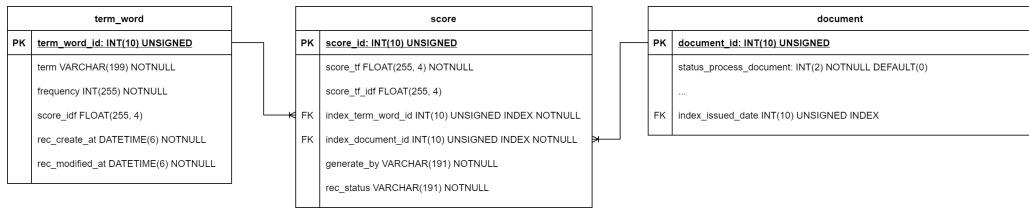
2. JavaScript Backend and Frontend

- nodejs เวอร์ชัน 12.16.3
- apollo-server-express เวอร์ชัน 2.19.0
- axios เวอร์ชัน 0.20.0
- cors เวอร์ชัน 2.8.5
- dotenv เวอร์ชัน 8.2.0
- express เวอร์ชัน 4.17.1
- graphql เวอร์ชัน 15.4.0
- jsonwebtoken เวอร์ชัน 8.5.1
- knex เวอร์ชัน 0.21.5
- morgan เวอร์ชัน 1.10.0
- mysql2 เวอร์ชัน 2.2.1
- password-hash เวอร์ชัน 1.2.2
- react เวอร์ชัน 16.13.1
- react-hook-form เวอร์ชัน 6.3.1
- react-router-dom เวอร์ชัน 5.2.0
- styled-components เวอร์ชัน 5.1.1
- props-types เวอร์ชัน 15.7.2

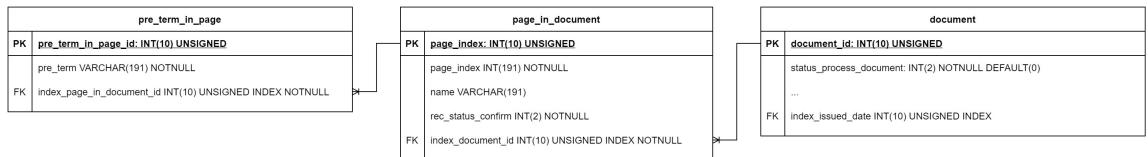
3.4 โครงสร้างฐานข้อมูล



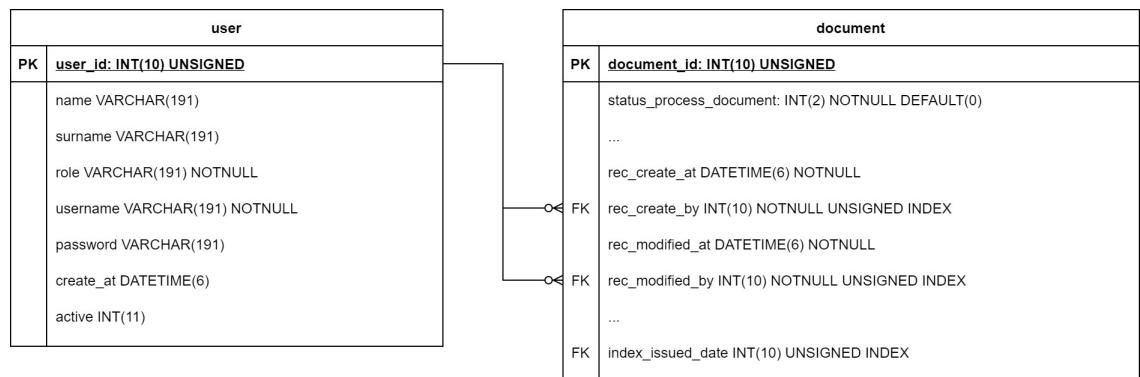
รูปที่ 3.29: แสดง ER Diagram ของฐานข้อมูล



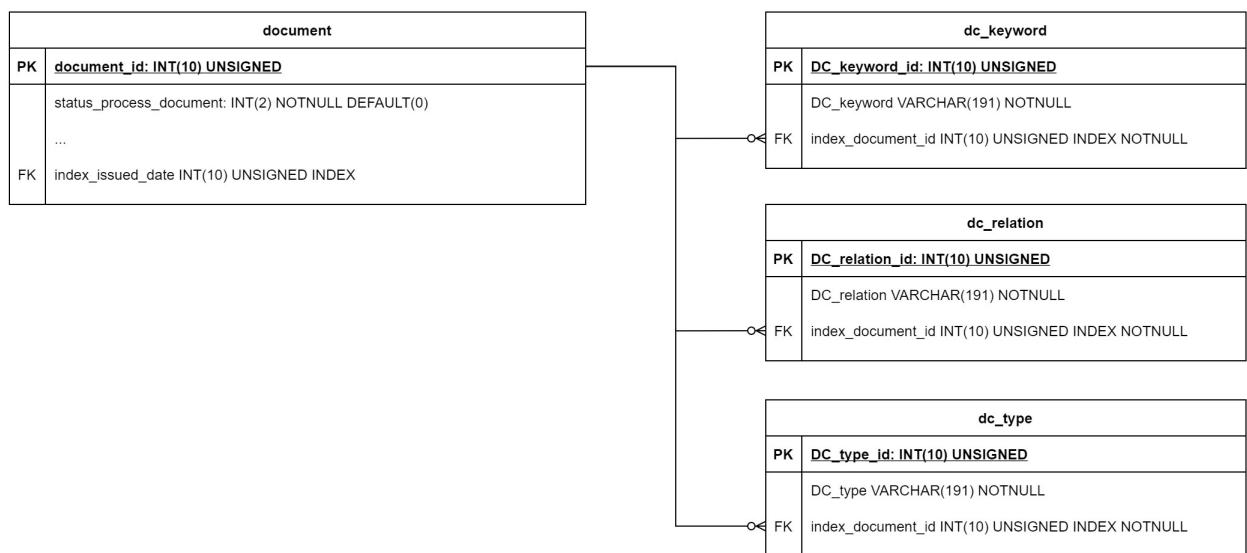
รูปที่ 3.30: แสดง ER Diagram ส่วนของคิร์เริร์ดและคงแหนนความสำคัญในระบบ



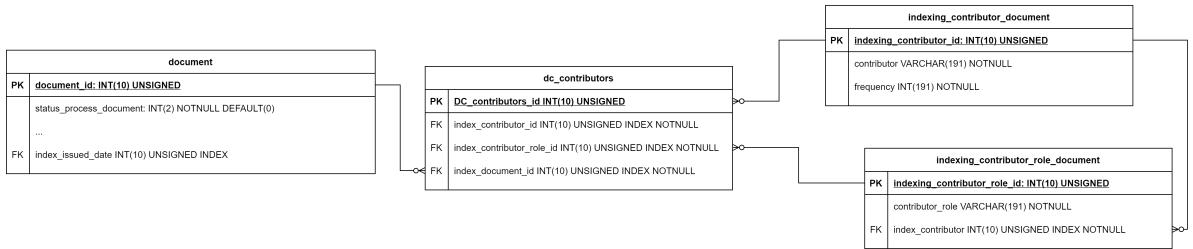
รูปที่ 3.31: แสดง ER Diagram ส่วนของการเก็บคำจากแต่ละหน้าที่แปลงมาจากหนังสือ



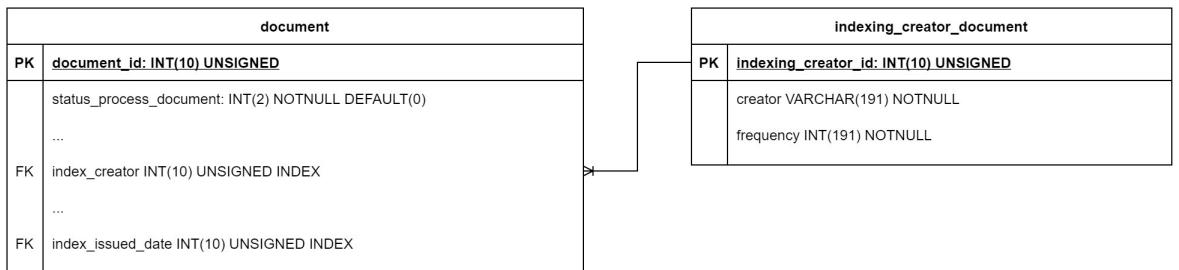
รูปที่ 3.32: แสดง ER Diagram ส่วนของประวัติของผู้ใช้งานมีการสร้างหรือแก้ไขหนังสือ



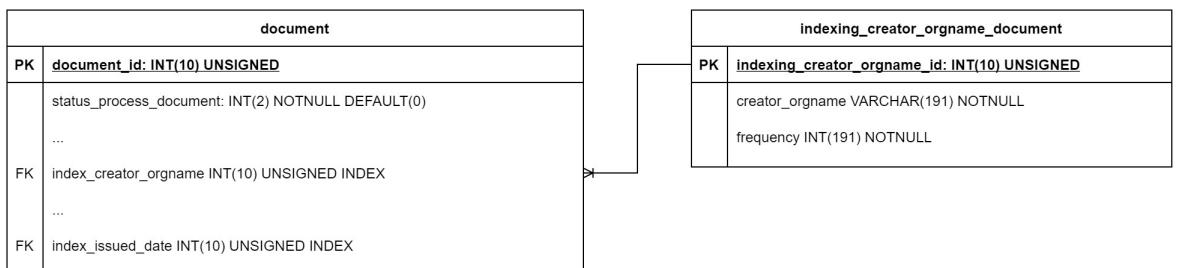
รูปที่ 3.33: แสดง ER Diagram ส่วนของการเก็บข้อมูล keyword, relation, type ของหนังสือ



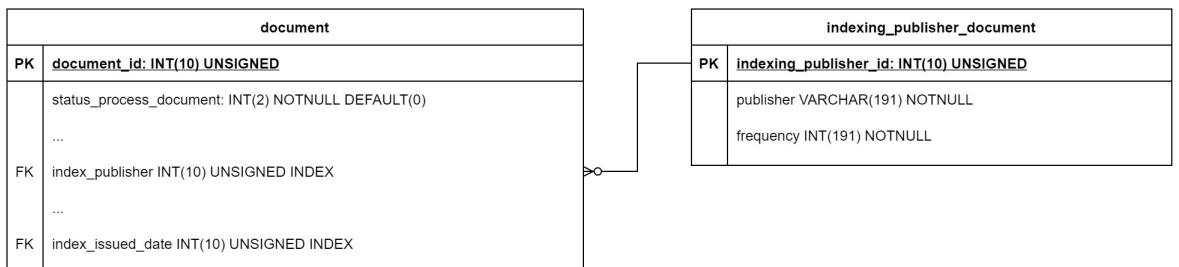
รูปที่ 3.34: แสดง ER Diagram ส่วนของการเก็บข้อมูล Contributors ว่ามีความเกี่ยวข้องกับหนังสือไหนบ้าง



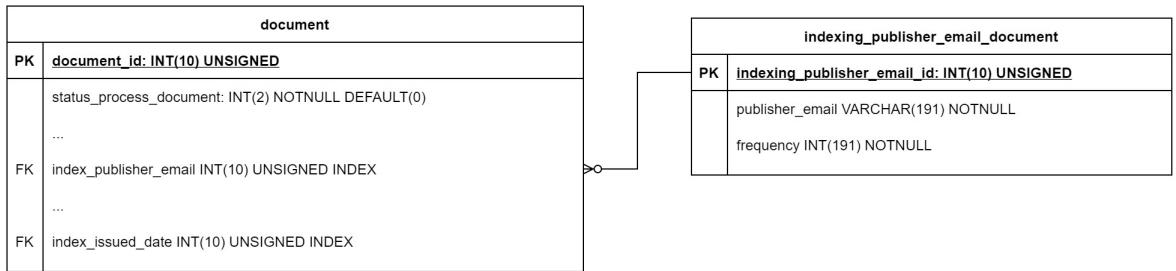
รูปที่ 3.35: แสดง ER Diagram ส่วนของ Creator มีความเกี่ยวข้องกับหนังสือไหนบ้าง



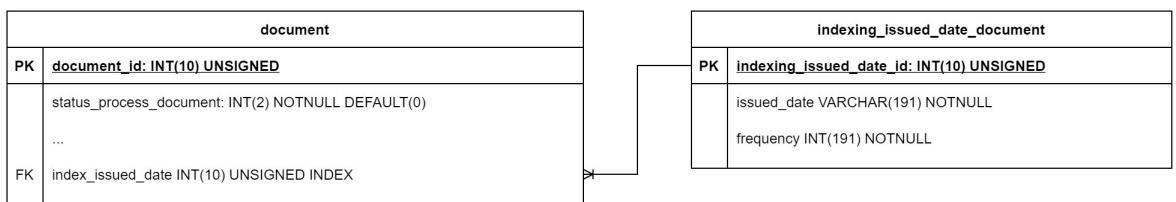
รูปที่ 3.36: แสดง ER Diagram ส่วนของ Creator Organized Name มีความเกี่ยวข้องกับหนังสือไหนบ้าง



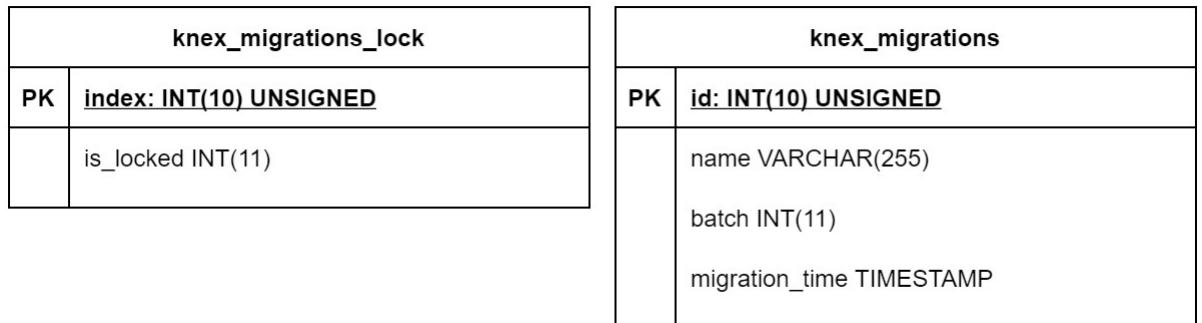
รูปที่ 3.37: แสดง ER Diagram ส่วนของ Publisher มีความเกี่ยวข้องกับหนังสือไหนบ้าง



รูปที่ 3.38: แสดง ER Diagram ส่วนของ Publisher Email มีความเกี่ยวข้องกับหนังสือในบ้าง



รูปที่ 3.39: แสดง ER Diagram ส่วนของ Issued Date มีความเกี่ยวข้องกับหนังสือในบ้าง



รูปที่ 3.40: แสดง ER Diagram ส่วนของ Knex module ที่ใช้สำหรับ Migration ฐานข้อมูล

nodejs_log	
PK	<u>nodejs_log_id: INT(10) UNSIGNED</u>
	status_code INT(191) header_date VARCHAR(191) server VARCHAR(191) url VARCHAR(191) content_type VARCHAR(191) rec_status INT(191) rec_create_date DATETIME(6)

รูปที่ 3.41: แสดง ER Diagram ส่วนของการเก็บประวัติการ HTTP Request NodeJS ไปยัง Django

3.4.1 Database Structure

รูปที่ 3.29 แสดงฐานข้อมูลของทั้งระบบโดยจะมีหลัก ๆ ทั้งหมดสามส่วน ทางด้านฝั่งขวาของตาราง document จะเป็นตารางที่เก็บข้อมูลเพิ่มเติมจากตาราง document และส่วนทางด้านฝั่งซ้ายของตาราง document สำหรับการเก็บข้อมูลในด้านของการทำระบบการเก็บคำจากหนังสือที่ถูกใส่ลงมาในระบบ ระบบการแปลงคำเป็นคีย์เวิร์ดและคะแนน TF-IDF ที่นำมาใช้สำหรับการค้นหาหนังสือ ระบบจัดการฐานข้อมูลผู้ใช้งาน และการตรวจสอบความผิดพลาดที่มีโอกาสจากการสร้างคีย์เวิร์ด และส่วนสุดท้ายที่เป็นตารางที่ไม่มีการเชื่อมโยงกับตารางใด ๆ จะมีไว้สำหรับการทำระบบฐานข้อมูล และระบบตรวจสอบ HTTP Request ของทาง NodeJS

รูปที่ 3.30 จะเป็นส่วนของคีย์เวิร์ด และคะแนนเพื่อนำมาใช้สำหรับการค้นหาหนังสือของระบบนี้ โดยจะมีทั้งหมดสามตาราง document, term_word, score ตาราง document จะเป็นตารางที่เก็บข้อมูลของหนังสือไว้ ส่วนตาราง term_word จะเป็นการเก็บคีย์เวิร์ด และคะแนน IDF สำหรับการลดความสำคัญของคีย์เวิร์ดนั้น ๆ ไว้ซึ่งทั้งสองตารางนี้จะเป็นความสัมพันธ์แบบ one to many กับตาราง score ที่จะมีคะแนนสำหรับระบบการค้นหาเก็บเอาไว้ ที่มีความสัมพันธ์แบบนี้เนื่องจากในแต่ละคีย์เวิร์ดมีโอกาสพบได้ในหลายหนังสือ และหนังสือเองก็สามารถมีได้หลายคีย์เวิร์ด เนื่องจากแต่ละคีย์เวิร์ดที่อยู่ต่างหนังสือกันจะมีคะแนนไม่เท่ากัน

รูปที่ 3.31 จะเป็นส่วนของการเก็บคำที่แปลงมาจากหนังสือไว้โดยเริ่มที่ตาราง document จะที่สามารถบอกได้ว่าหนังสือไหน ที่จะมีความพันธ์ one to many ไปยังตาราง page_in_document ที่จะเป็นตารางที่บอกร่องหน้าต่าง ๆ ในหนังสือนั้น และยังมีความสัมพันธ์ one to many ต่อไปยังตาราง per_term_in_page ที่จะมีค่าต่าง ๆ เก็บเอาไว้ ดังนั้นจะเป็นความสัมพันธ์ที่หนังสือนั้นจะสามารถมีได้หลายหน้าแล้วแต่ละหน้าเองก็จะมีค่าต่าง ๆ ที่แปลงออกมากถูกเก็บเอาไว้

รูปที่ 3.32 จะเป็นความสัมพันธ์ของบัญชีผู้ใช้กับหนังสือ โดยจะมีตาราง user ที่จะเก็บข้อมูลของผู้ใช้งานที่มีความสัมพันธ์แบบ one to many ไปยังตาราง document ที่จะเก็บต้องเก็บข้อมูลของผู้ใช้ไว้ผู้ใช้คนไหนเป็นคนสร้าง หรือแก้ไขหนังสือนี้ ซึ่งบัญชีผู้ใช้สามารถสร้างหรือแก้ไขหนังสือได้หลายหนังสือ

รูปที่ 3.33 จะเป็นส่วนของข้อมูลของตาราง Document เมื่อกันแต่เนื่องจากข้อมูลมีมากกว่าหนึ่งทำให้ต้องสร้างความสัมพันธ์แบบ one to many กับตาราง dc_keyword, dc_relation, dc_type ซึ่งจะเป็นข้อมูลคีย์เวิร์ด ความสัมพันธ์ และประเภทของหนังสือตามลำดับ

รูปที่ 3.34 จะเป็นการเก็บข้อมูลของ Contributor โดยจะมีตารางแยกเพื่อเก็บของสัมพันธ์ของทั้งสองด้านเนื่องจากในเอกสารสามารถมี contributor ได้หลายคน และ contributor สามารถมีหลายเอกสารเช่นกัน โดยที่ contributor จะมี role เป็นของตัวเองซึ่งสามารถมี

略有 role เช่น กันทำให้ต้องมีตาราง รองรับเพิ่ม แต่เนื่องจากในเอกสารเล่มนึงนั้น contributor จะสามารถมี role ได้แค่อย่างเดียวเท่านั้น ดังนั้นจะเห็นว่ามีการเขียน role อีกครั้งเพื่อระบุให้แต่ละเอกสารอย่างเฉพาะเจาะจง

รูปที่ 3.35 จะเป็นส่วนของการเก็บความสัมพันธ์ระหว่าง Creator กับหนังสือ เนื่องจาก Creator สามารถมีเดิ้ลลายหนังสือทำให้ตาราง indexing_creator_document จะเป็นความสัมพันธ์แบบ one to many กับตาราง document

รูปที่ 3.36 จะเป็นส่วนของการเก็บความสัมพันธ์ระหว่าง Creator orgname กับหนังสือเนื่อง จากCreator orgname สามารถมีเดิ้ลลายหนังสือทำให้ตาราง indexing_creator_orgname_document จะเป็นความสัมพันธ์แบบ one to many กับตาราง document

รูปที่ 3.37 จะเป็นส่วนของการเก็บความสัมพันธ์ระหว่าง Publisher กับหนังสือ เนื่องจาก Publisher สามารถมีเดิ้ลลายหนังสือทำให้ตาราง indexing_publisher_document จะเป็นความสัมพันธ์แบบ one to many กับตาราง document

รูปที่ 3.38 จะเป็นส่วนของการเก็บความสัมพันธ์ระหว่าง Publisher Email กับหนังสือ เนื่องจาก Publisher Email สามารถมีเดิ้ลลายหนังสือทำให้ตาราง indexing_publisher_email_document จะเป็นความสัมพันธ์แบบ one to many กับตาราง document

รูปที่ 3.39 จะเป็นส่วนของการเก็บความสัมพันธ์ระหว่าง Issued Date กับหนังสือ เนื่องจาก Issued Date สามารถมีเดิ้ลลายหนังสือทำให้ตาราง indexing_issued_date_document จะเป็นความสัมพันธ์แบบ one to many กับตาราง document

รูปที่ 3.40 จะเป็นสองตารางที่บันทึกการจัดการฐานข้อมูลของเครื่องมือที่ชื่อว่า Knex ที่จะทำการจัดการสร้างฐานข้อมูล ด้วยคำสั่ง Migration เล้าหลังจากทำการคำสั่งเสร็จสิ้นจะเก็บบันทึกไว้

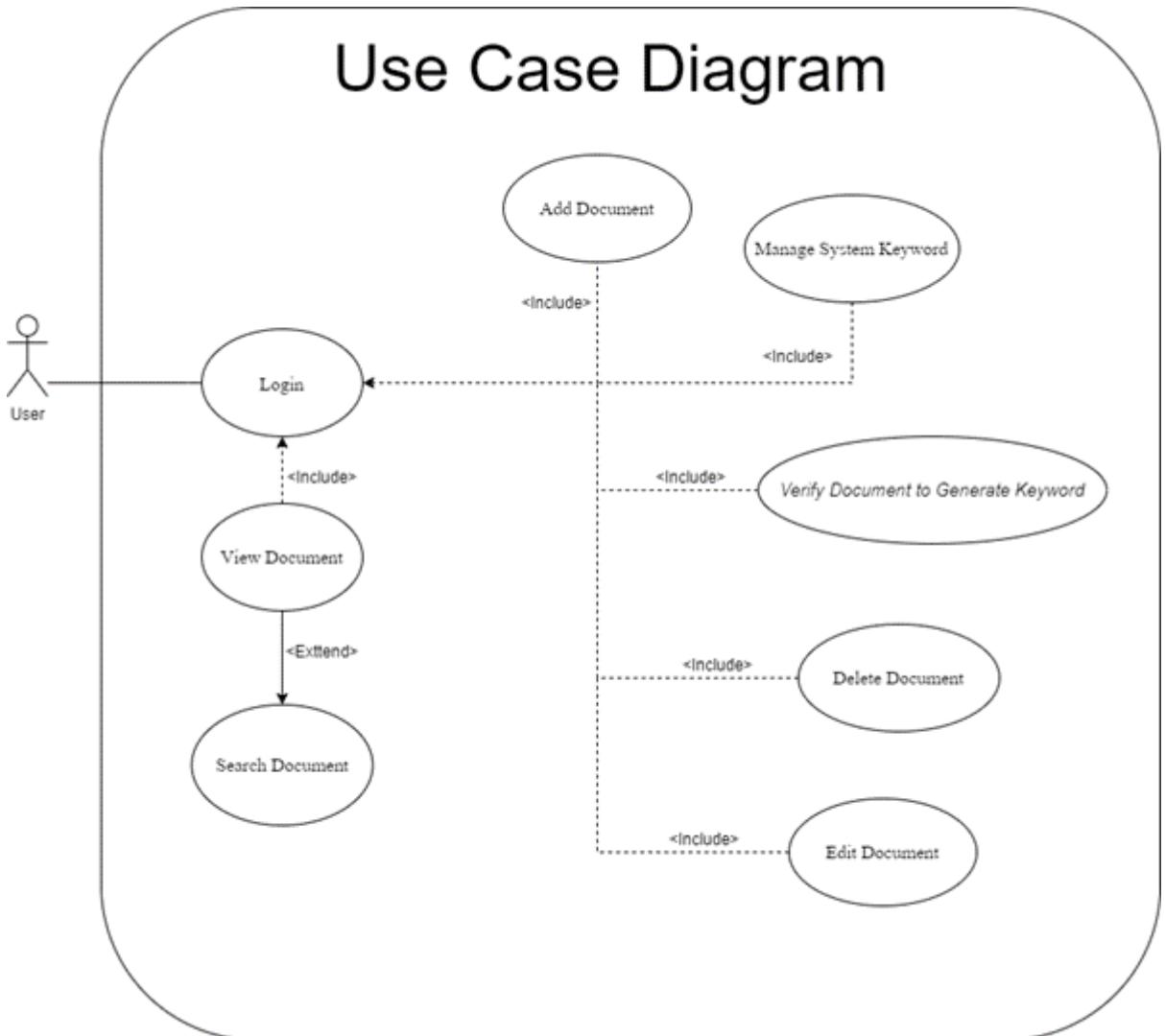
รูปที่ 3.41 จะเป็นตารางสำหรับการเก็บ HTTP Request จาก NodeJS ที่ส่งไปทางฝั่งของ Django ซึ่งจะถูกเก็บข้อมูลไว้ในตารางนี้

3.4.2 Database Dictionary

ในส่วนของการอธิบายถึงชื่อของคอลัมน์ ความหมายและลักษณะการเก็บข้อมูลภายใต้ฐานข้อมูลโดยที่ตารางมีทั้งหมด 18 ตารางจะอยู่ในส่วนของภาคผนวก

3.5 UML Design

3.5.1 Use case diagram



รูปที่ 3.42: Use case diagram

3.5.2 Sequence diagram

3.5.2.1 Use case Add Document

Scenario 1: เพิ่มหนังสือ/หนังสือเข้าสู่ระบบ

Goal: เพิ่มข้อมูลของหนังสือเข้าไปอยู่ในระบบ

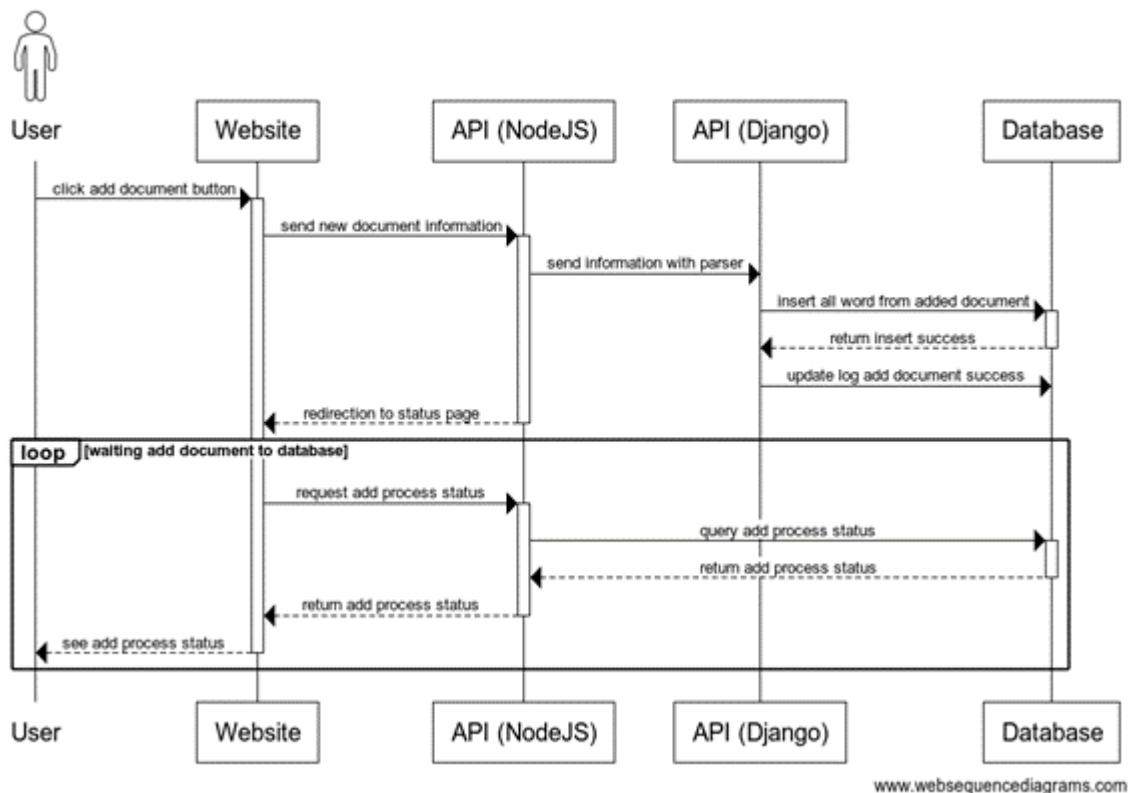
Precondition: กดไปที่หัวข้อ INSERT BOOK ใน Web Application

Main success scenario:

1. อัปโหลดหนังสือ/หนังสือเลือกหน้าที่จะให้เริ่มต้นการแปลง

2. กรอกข้อมูลรายละเอียดที่ต้องการลงในระบบ
3. แสดงสถานะของการเพิ่มข้อมูล
4. เพิ่มหนังสือ/หนังสือเข้าสู่ระบบ

Use case Add Document



รูปที่ 3.43: แสดง Scenario 1 เพิ่มหนังสือเข้าระบบ

3.5.2.2 Use case Manage word in document

Scenario 2: การตรวจสอบและแก้ไขคำก่อนนำเข้าสู่ระบบ

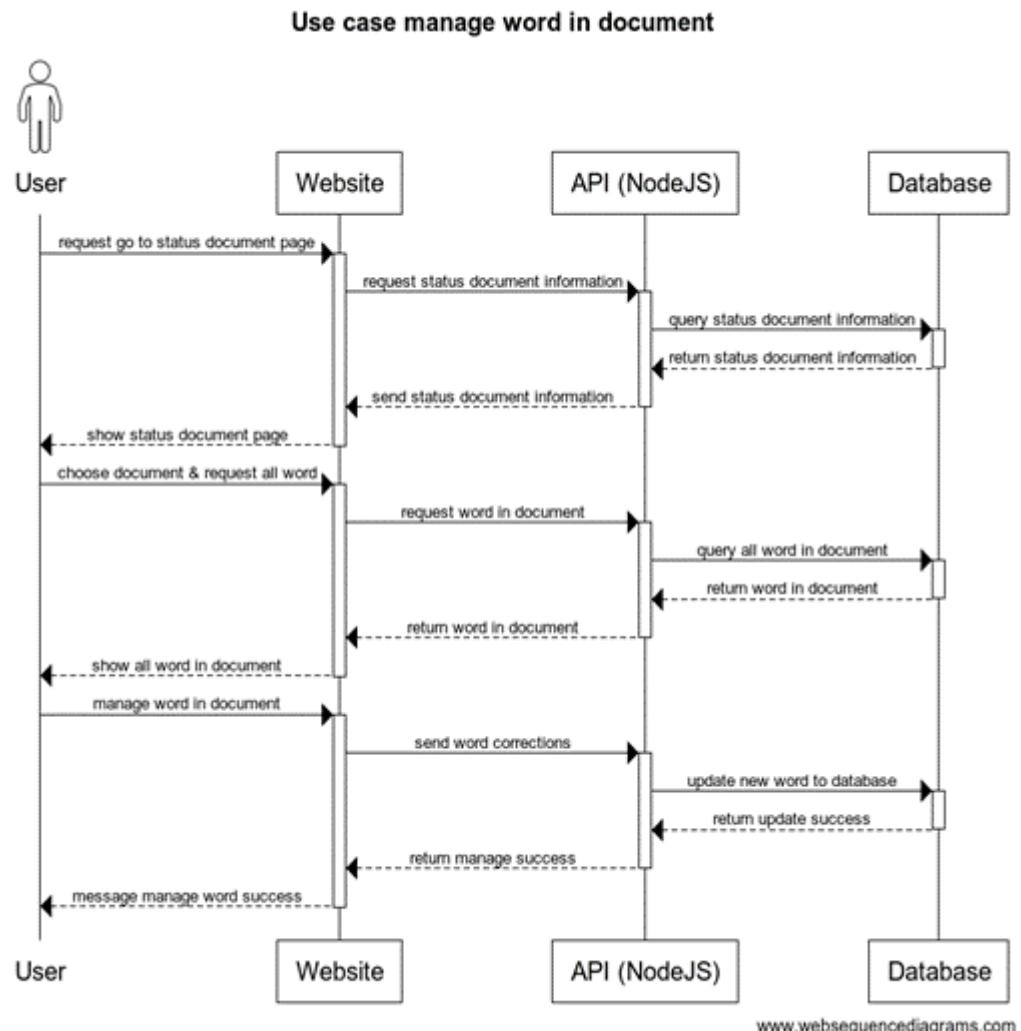
Goal: ผู้ใช้งานเห็นคำที่จะถูกแปลงเป็นดิจิทัลแล้วสามารถจัดการคำเหล่านั้นได้

Precondition: อยู่ภายใต้ขั้นตอนการเพิ่มหนังสือ/หนังสือลงในระบบ

Main success scenario:

1. ผู้ใช้เข้าไปบันทึกหน้าสถานะการเพิ่มหนังสือ
2. ผู้ใช้เลือกหนังสือที่อยู่ในสถานะตรวจสอบคำ
3. ระบบแสดงคำทั้งหมดที่ถูกแปลงมาได้จากหนังสือแต่ละหน้า
4. ผู้ใช้ตรวจสอบ แก้ไขคำที่แสดงขึ้นมา

5. ยืนยันขั้นตอนการตรวจสอบและแก้ไขคำ



รูปที่ 3.44: แสดง Scenario 2 การจัดการคำที่ถูกเก็บได้จากหนังสือในระบบ

3.5.2.3 Use case Verify Document to Generate Keyword

Scenario 3: ยืนยันว่าพร้อมสำหรับการถูกนำไปสร้างคีย์เวิร์ด

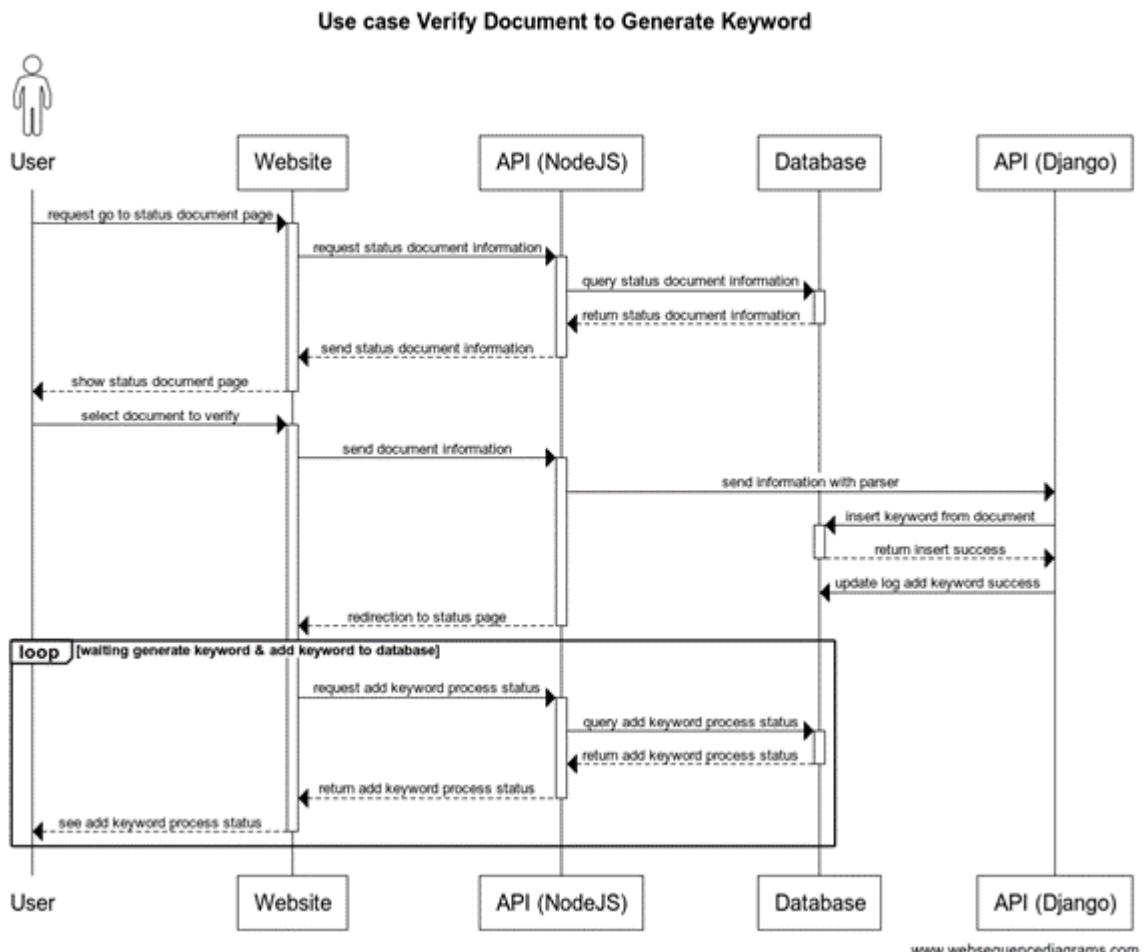
Goal: หนังสือถูกยืนยันพร้อมกับสร้างคีย์เวิร์ดเพื่อเพิ่มเข้าไปในระบบ

Precondition: ใบยังหน้าสถานะของหนังสือแล้วกดไปยังปุ่มยืนยันหนังสือถูกต้อง

Main success scenario:

1. ผู้ใช้เข้าไปยังหน้าสถานะการเพิ่มหนังสือ
2. ระบบแสดงสถานะหนังสือว่าหนังสือไหนอยู่สถานะได้แล้วบ้าง
3. ผู้ใช้กดยืนยันว่าหนังสือถูกต้อง

4. ระบบย้ายไปหน้าสถานะหนังสืออีกครั้งเพื่อรอผลการทำงาน
5. ระบบแสดงการยืนยันหนังสือ และถูกเชื่อมต่อคีย์เวิร์ดเสร็จสิ้น



รูปที่ 3.45: แสดง Scenario 3 ยืนยันว่าพร้อมสำหรับการถูกนำไปสร้างคีย์เวิร์ด

3.5.2.4 Use case Edit Document

Scenario 4: การแก้ไขรายละเอียดของหนังสือ/หนังสือที่อยู่ภายใต้ระบบ

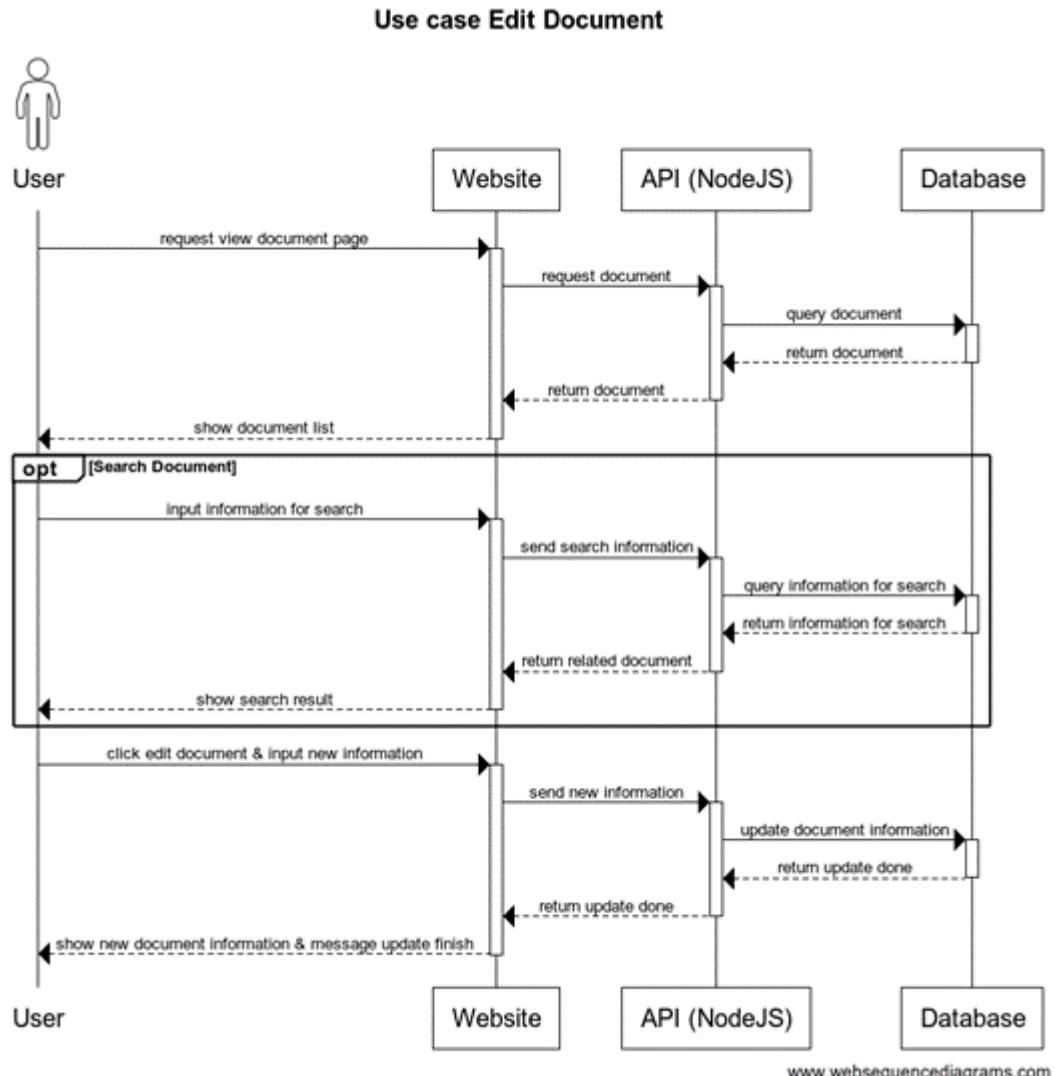
Goal: รายละเอียดหนังสือถูกแก้ไขตามผู้ใช้งานต้องการ

Precondition: กดไปที่หัวข้อ MANAGE BOOK ใน Web Application

Main success scenario:

1. ผู้ใช้ค้นหาหนังสือที่ต้องการแก้ไขรายละเอียด
2. แสดงผลลัพธ์ในการค้นหาหนังสือ/หนังสือ
3. เลือกหนังสือ/หนังสือที่ต้องการแก้ไขรายละเอียด

4. แก้ไขรายละเอียดที่ต้องการ
5. กดบันทึกข้อมูลลงในระบบ



รูปที่ 3.46: แสดง Scenario 4 แก้ไขข้อมูลหนังสือ

3.5.2.5 Use case Delete Document

Scenario 5: ลบหนังสือ/หนังสือภายในระบบ

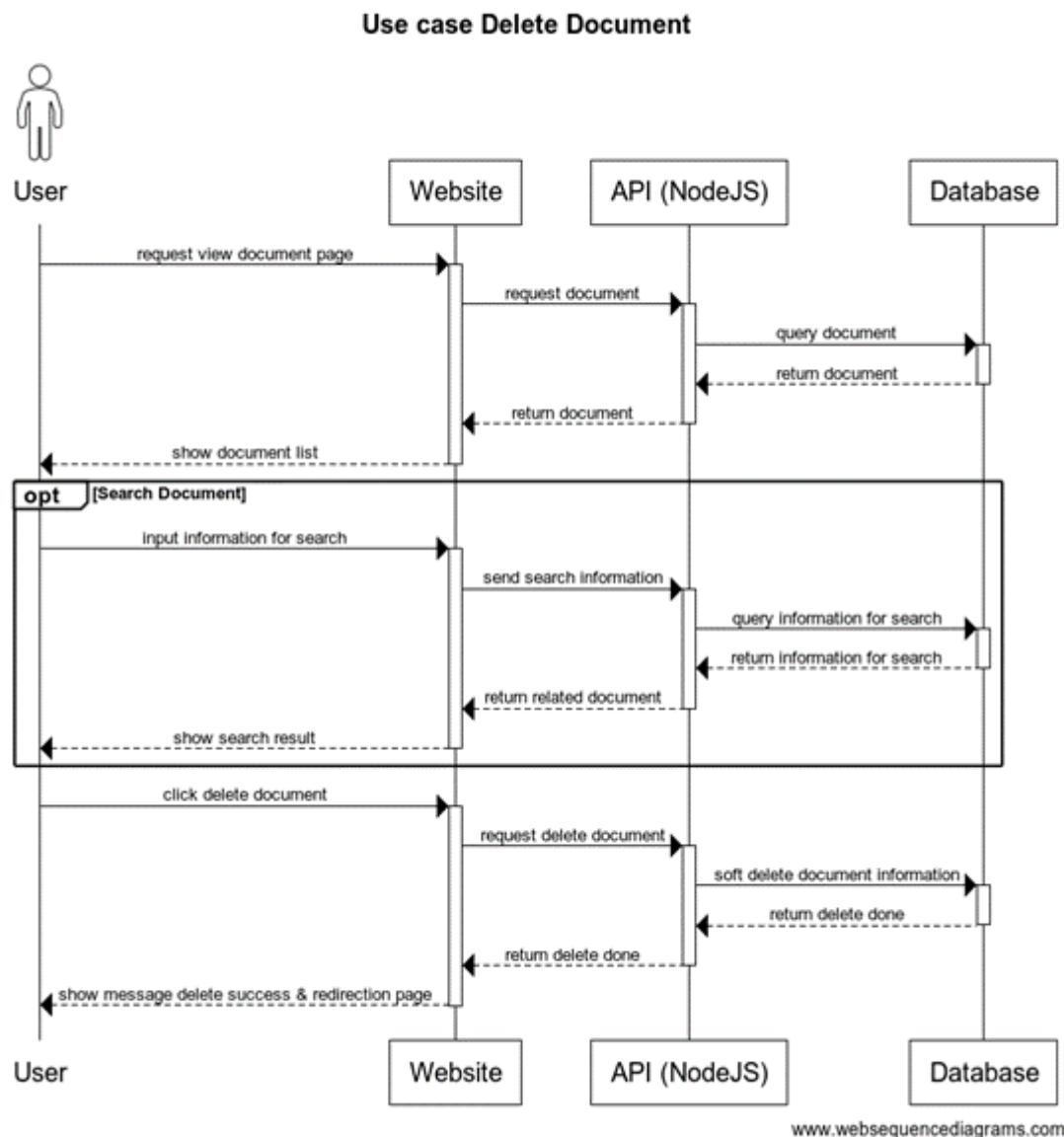
Goal: หนังสือ/หนังสือถูกนำออกจากระบบ

Precondition: กดเลือกหัวข้อ MANAGE BOOK ใน Web Application

Main success scenario:

1. ผู้ใช้ทำการค้นหาหนังสือที่ต้องการจะลบออกจากระบบ

2. แสดงผลลัพธ์ในการค้นหาหนังสือ/หนังสือ
3. กดลบหนังสือ/หนังสือที่ต้องการ
4. กดยืนยันคำสั่งลบเพื่อบันทึกลงระบบ



รูปที่ 3.47: แสดง Scenario 5 ลบหนังสือ

3.5.2.6 Use case View Document & Search Document

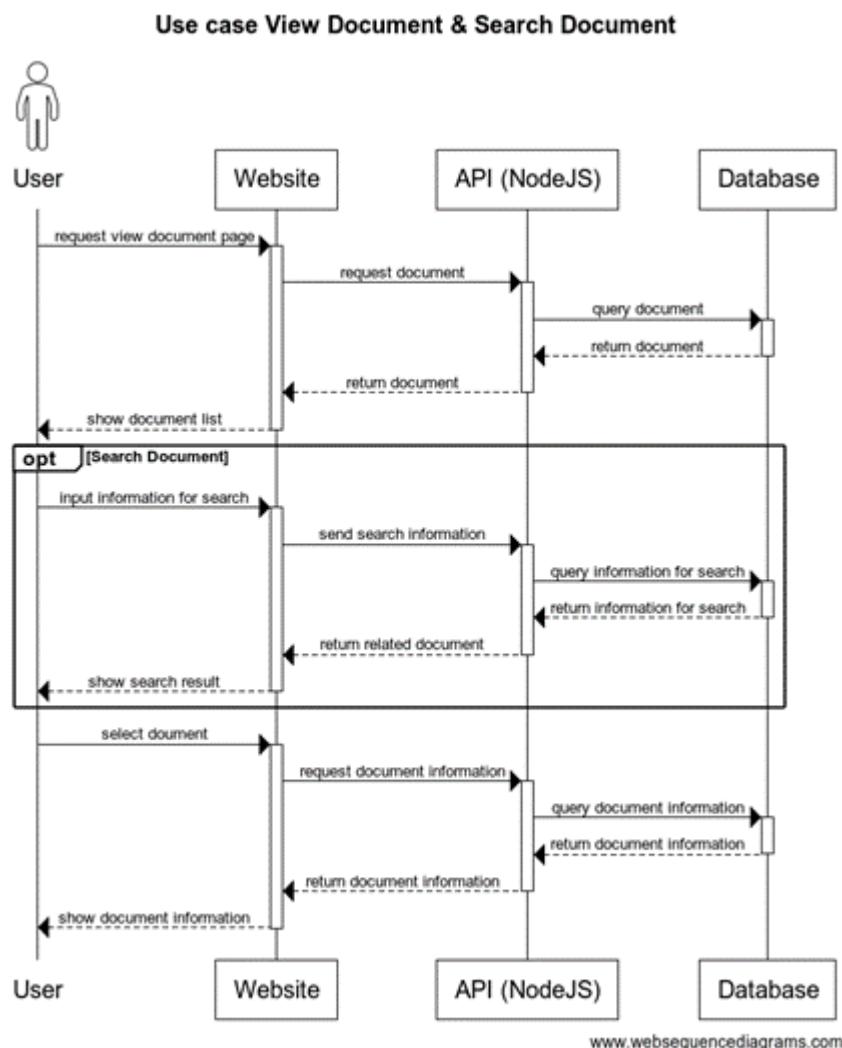
Scenario 6: ดูข้อมูลหนังสือ และการค้นหาหนังสือ

Goal: ผู้ใช้เจอหนังสือที่ต้องการ

Precondition: กดไปที่หัวข้อ SEARCH ใน Web Application

Main success scenario:

1. กรอกรายละเอียดข้อมูลที่ต้องการจะค้นหา
2. แสดงผลลัพธ์ในการค้นหา
3. ผู้ใช้เลือกหนังสือที่ต้องการที่จะดูข้อมูล
4. ระบบย้ายไปยังหน้าแสดงข้อมูลหนังสือที่ถูกเลือก



รูปที่ 3.48: แสดง Scenario 6 ดูข้อมูลหนังสือ และการค้นหาหนังสือ

3.5.2.7 Use case Login

Scenario 7: ระบบล็อกอิน

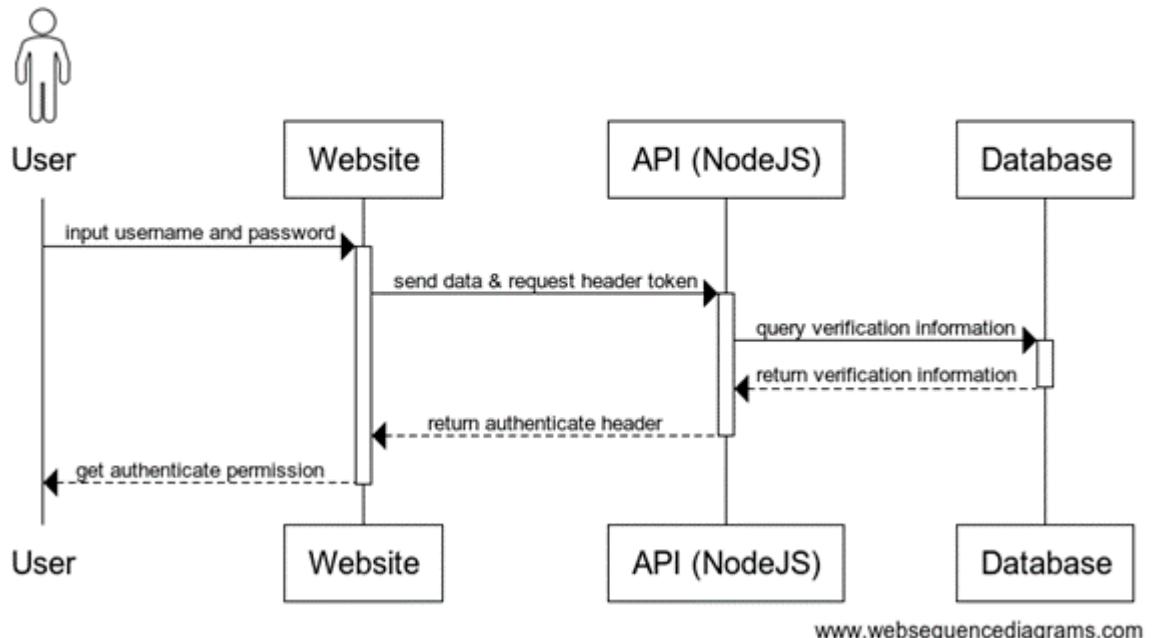
Goal: เพื่อเข้าสู่ระบบให้สามารถใช้งานได้ภายใน Web Application เพิ่มเติมได้

Precondition: กดหัวข้อ LOGIN ใน Web Application

Main success scenario:

1. ผู้ใช้กรอกชื่อผู้ใช้งานและรหัสผ่าน
2. กดเข้าสู่ระบบ
3. เข้าสู่ระบบสำเร็จ ส่งผู้ใช้กลับไปสู่ Homepage
4. สามารถเข้าใช้งานฟังก์ชันของ Web Application ได้

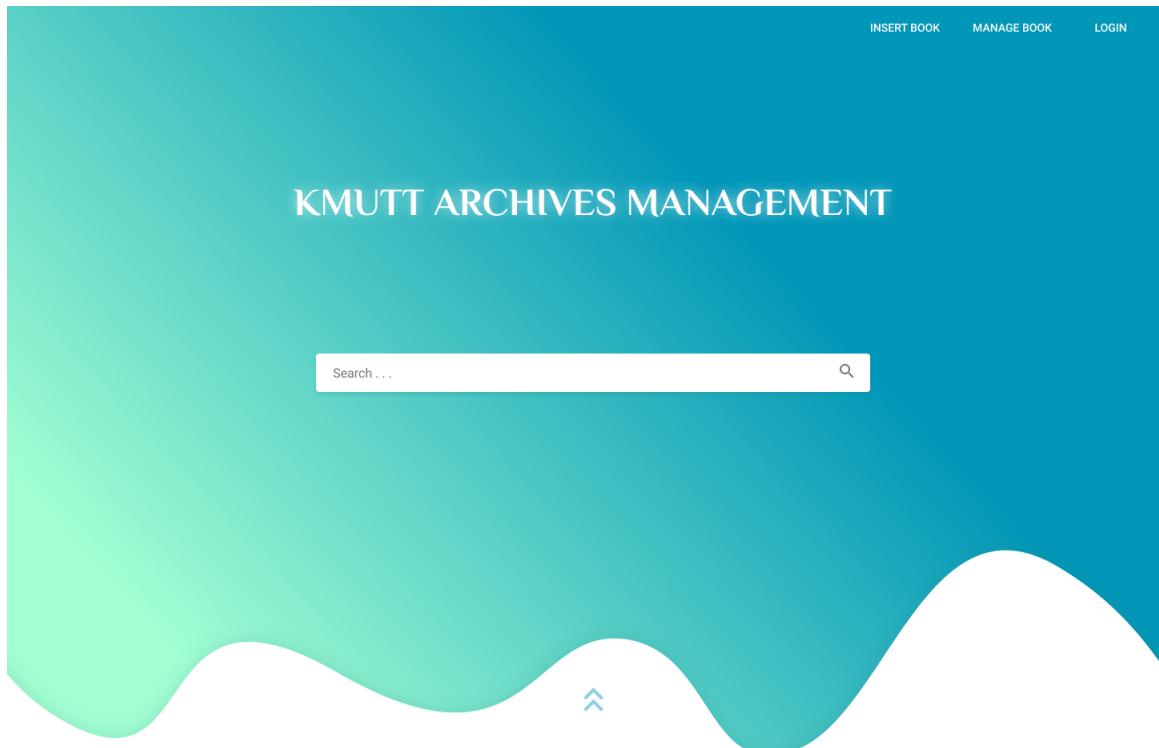
Use case Login



รูปที่ 3.49: แสดง Scenario 7 ระบบตีกอกอิน

3.6 GUI Design

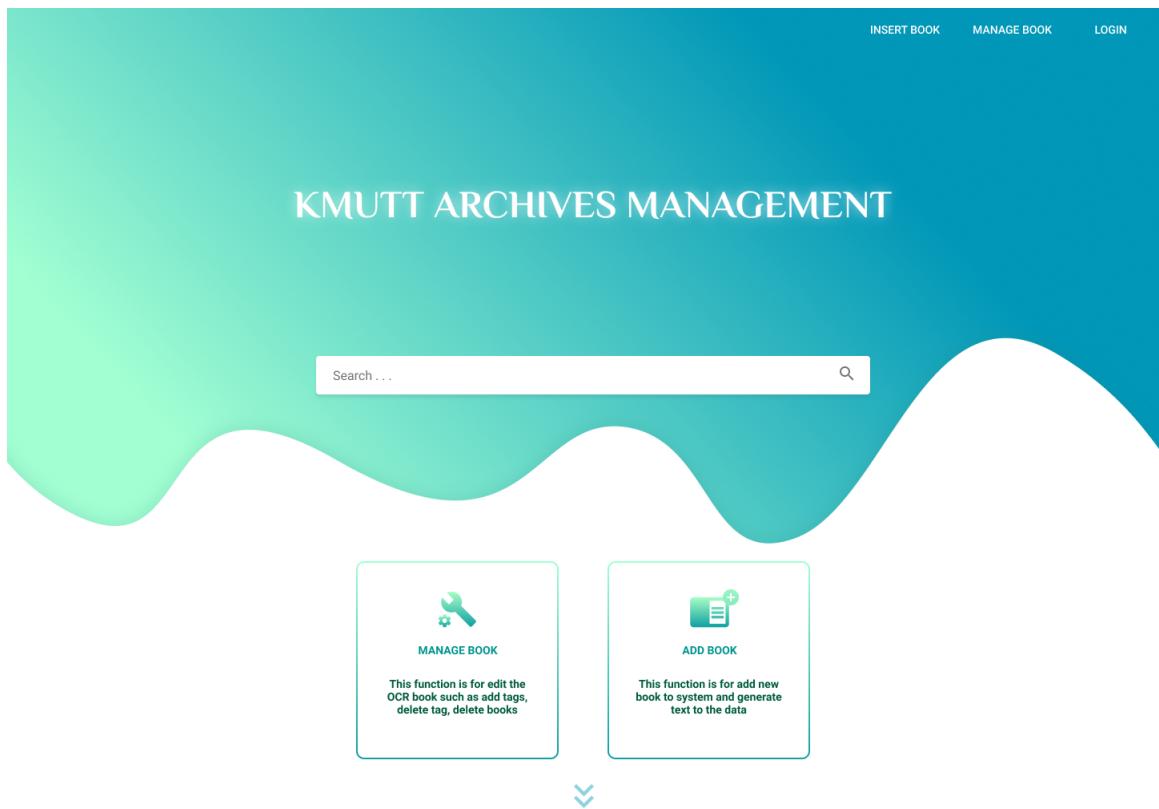
3.6.1 Homepage



รูปที่ 3.50: ภาพแสดงหน้าหลักของเว็บไซต์

หน้าหลักของเว็บไซต์จะเป็นหน้าที่เน้นการค้นหาเป็นหลัก ที่ผู้ใช้สามารถเข้าถึงเมนูการเพิ่มหนังสือ การจัดการ และการเข้าสู่ระบบได้ที่แถบ Navigation ด้านบนของเว็บไซต์ดังรูปที่ 3.50

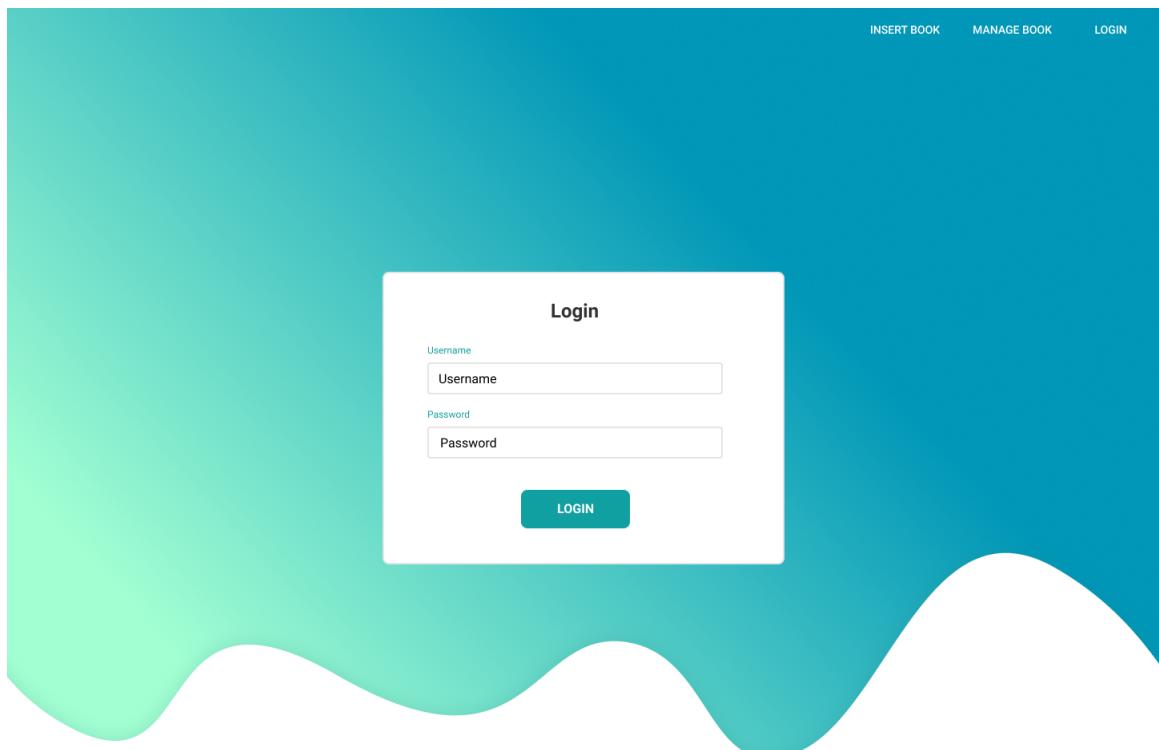
3.6.2 Homepage2



รูปที่ 3.51: ภาพแสดงหน้าหลักของเว็บไซต์หลังจากการกดเปิดเมนู

เมื่อกดปุ่มลูกศรที่ด้านล่างของรูป 3.50 จะมีเมนูเพิ่มเติมขึ้นมาภายเป็นรูปที่ 3.51 ซึ่งจะแสดงรายละเอียดในแต่ละฟังก์ชันเพิ่มเติม

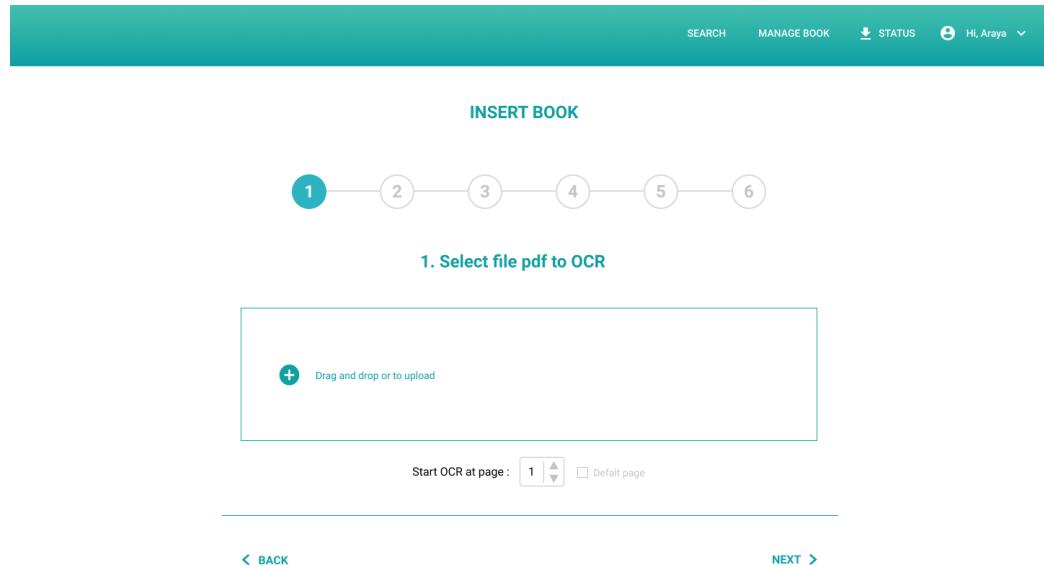
3.6.3 Login



รูปที่ 3.52: ภาพแสดงหน้าเข้าสู่ระบบ

ก่อนที่จะทำการเพิ่มหนังสือหรือจัดการกับหนังสือผู้ใช้นั้นจะต้องเข้าสู่ระบบก่อนเสมอ ถ้าเกิดกดเข้าฟังก์ชันการเพิ่มหนังสือหรือค้นหาโดยที่ยังไม่ได้เข้าสู่ระบบ ระบบจะบังคับให้ผู้ใช้เข้ามาในหน้าเข้าสู่ระบบตั้งรูป 3.25 เพื่อทำการเข้าสู่ระบบหรือจะเข้ามาโดยการกด log in ที่ปุ่มขวางานได้

3.6.4 Insert Book (1)



รูปที่ 3.53: ภาพแสดงขั้นตอนการเพิ่มหนังสือเข้าสู่ระบบขั้นเลือกไฟล์

หน้าเพิ่มหนังสือขึ้นและเป็นการเลือกไฟล์หนังสือที่ต้องการโดยที่จะมีส่วนของการเพิ่มไฟล์ที่อยู่รูปของ pdf เพื่อทำ OCR จากนั้นจะสามารถเลือกได้ว่าจะทำการ OCR ตั้งแต่หน้าไหนดังรูปที่ 3.26

3.6.5 Insert Book (2)

2. Fill the data

Title

Title:
Title alternative:

Creator

Creator name:
Creator Organization name:

Description

Table of contents:
Summary:

Publisher

Publisher:
Publisher E-mail:

Contributor

Contributor:
Contributor Role:

Date

Date: DD/MM/YY

Coverage

Coverage Spatial:
Coverage Temporal: YYYY

Rights

Rights:
Rights Access: Available

[BACK](#) [NEXT](#)

รูปที่ 3.54: ภาพแสดงขั้นตอนการเพิ่มหนังสือเข้าสู่ระบบขั้นกรอกข้อมูลขั้นที่ 1

หน้าเพิ่มหนังสือขั้นตอนที่ 2 เป็นหน้าที่ต้องใส่ข้อมูลที่จำเป็นของหนังสือ โดยที่จำเป็นต้องใส่จะมีสัญลักษณ์กำกับไว้หรือคือชื่อหนังสือดังรูป 3.27 โดยในหน้านี้จะมีกล่องใส่ข้อมูลที่ถูกกรอกบ่อยๆ สำหรับผู้ใช้(เจ้าหน้าที่)

3.6.6 Insert Book (3)

SEARCH **MANAGE BOOK** **STATUS** **Hi, Araya**

INSERT BOOK

1 2 3 4 5 6

3. Optional data

Identifier

Identifier URL
Input

Identifier ISBN
Input

Source

Source
Input

Relation

Relation **+ ADD**
Input

Thesis

Degree name
Input

Degree level
Input

Degree discipline
Input

Degree grantor
Input

Type

Type
Text

Language

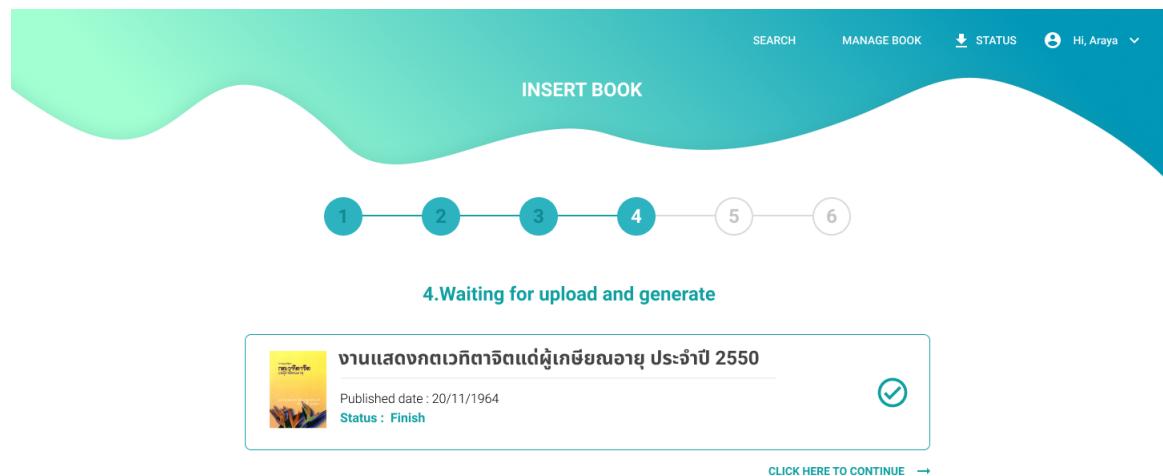
Language
Thai

< BACK **NEXT >**

รูปที่ 3.55: ภาพแสดงขั้นตอนการเพิ่มหนังสือเข้าสู่ระบบขั้นกรอกข้อมูลขั้นที่ 2

ในขั้นตอนที่ 3 จากรูปที่ 3.28 จะเป็นหน้าที่ใส่ข้อมูลที่ส่วนใหญ่ใช้จะไม่ค่อยกรอกมากนัก ซึ่งไม่มีกล่องข้อมูลใหม่จำเป็นที่ต้องกรอกผู้ใช้สามารถเข้ามายังไปขั้นตอนถัดไปได้เลย

3.6.7 Insert Book (4)



รูปที่ 3.56: ภาพแสดงขั้นตอนการเพิ่มหนังสือเข้าขึ้นໂ Holden ข้อมูลเข้าสู่ระบบ

หลังจากที่ทำการใส่ข้อมูลอกรมาทั้งหมดแล้วมาถึงหน้าที่เป็นหน้า Holden ข้อมูลดังรูป 3.29 ที่ระบบจะทำการ OCR และทำการเตรียมชุดข้อมูลที่ได้จากการ OCR โดยการนำคำมาตัดและเข็คคำผิด เมื่อ Holden ข้อมูลเสร็จแล้วระบบจะทำการเปลี่ยนสถานะการໂ Holden และขึ้นล็อกเพื่อเข้าสู่ขั้นตอนถัดไปได้

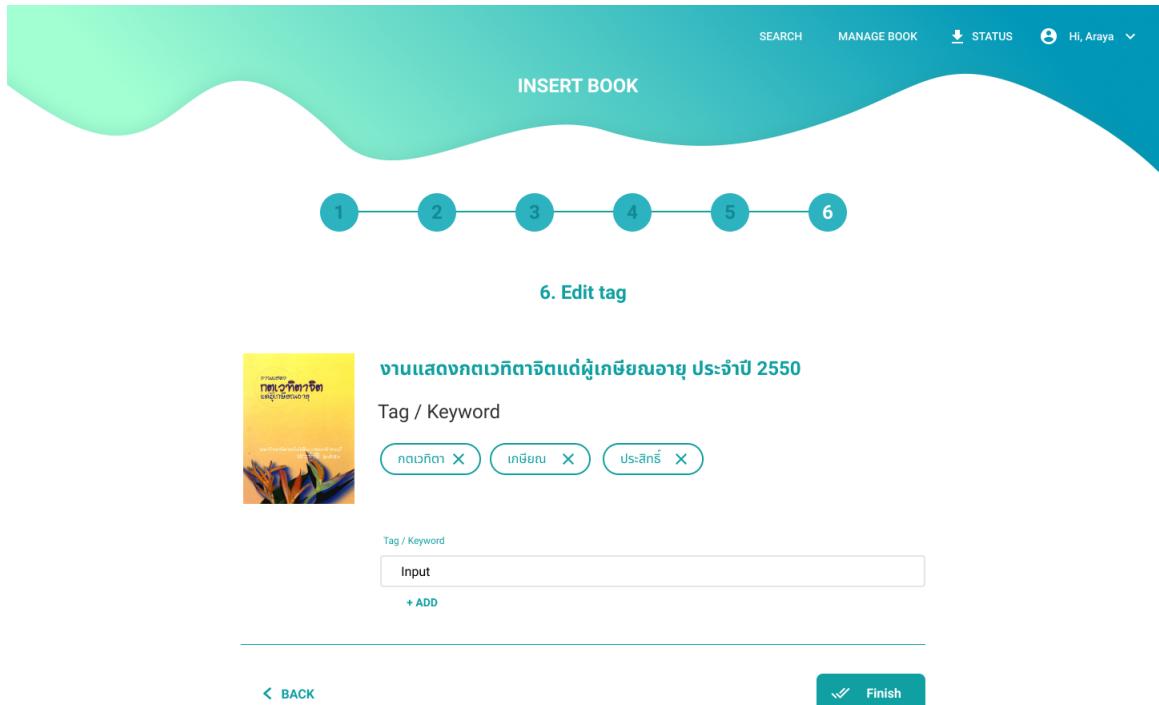
3.6.8 Insert Book (5)



รูปที่ 3.57: ภาพแสดงขั้นตอนการเพิ่มหนังสือเข้าสู่ระบบขั้นแก้ไขคำผิด

หลังจากโหลดและเตรียมข้อมูลเรียบร้อยแล้ว ระบบจะทำการแสดงข้อมูลที่ถูกแปลงมาโดยที่ผู้ใช้สามารถแก้ไขได้ดังรูป 3.30 หรือสามารถข้ามได้โดยเชื่อม กดโดยเมื่อคลิกไปที่กล่องข้อความจะขึ้นให้แก้แต่ละคำและเมื่อเปลี่ยนหน้าจะทำการเก็บข้อมูลที่เปลี่ยนไว้ และจะบันทึกการแก้ไขข้อมูลทั้งหมดที่แก้เมื่อข้ามไปขั้นตอนถัดไป

3.6.9 Insert Book (6)



รูปที่ 3.58: ภาพแสดงขั้นตอนการเพิ่มหนังสือเข้าสู่ระบบขั้นแก้ไขและเพิ่มคำสำคัญ

หน้าสุดท้ายของการเพิ่มหนังสือจะเป็นหน้าที่ให้ผู้ใช้สามารถจัดการรับ Keyword ได้ดังรูปที่ 3.31 โดยเมื่อผู้ใช้ต้องการใส่คำสำคัญเพิ่มสามารถกด ADD เพื่อเพิ่มคำที่ต้องการใส่ได้ และสามารถลบเมื่อคลิกที่ปุ่มลบหากต้องการลบคำที่ไม่ต้องการ หลังจากใส่เสร็จแล้วสามารถกดปุ่ม Finish เพื่อทำการบันทึกข้อมูล

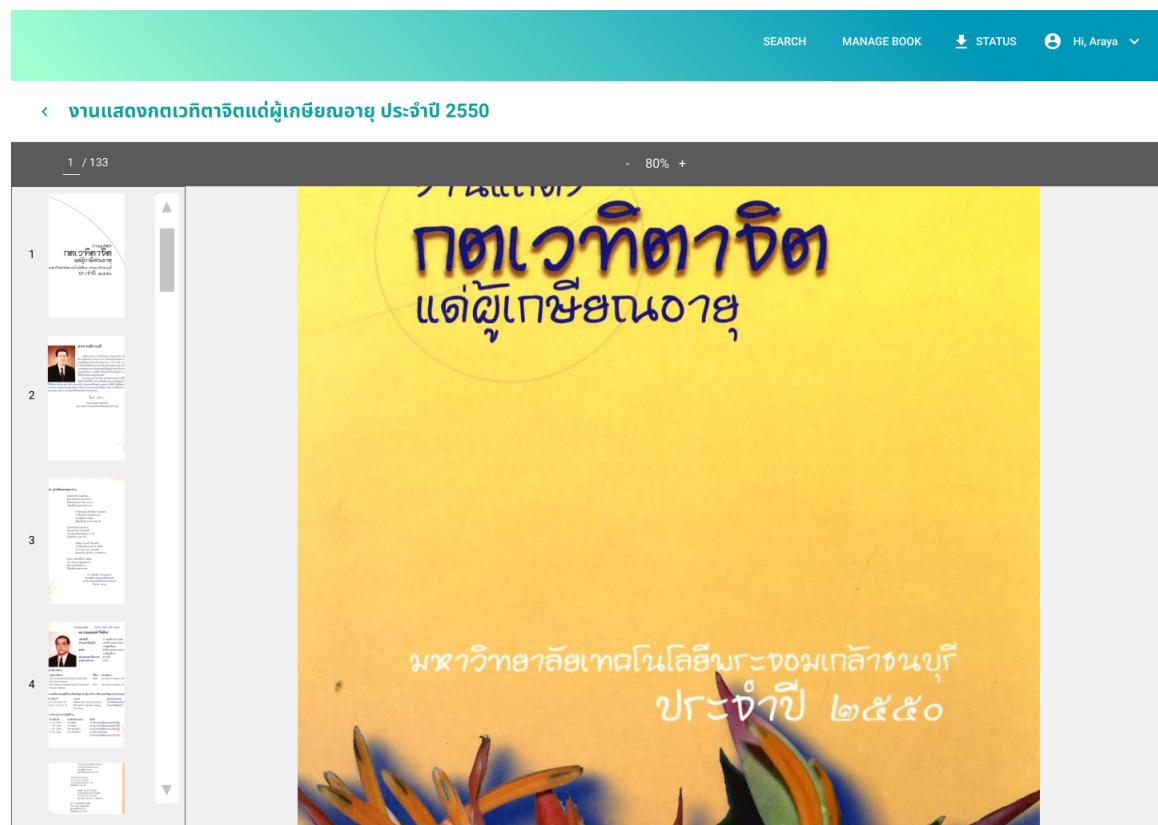
3.6.10 Search

The screenshot shows a library search interface with a teal header bar. On the right side of the header are buttons for 'EDIT BOOK', 'MANAGE BOOK', 'STATUS' (with a download icon), and a user profile 'Hi, Araya'. Below the header is a search bar with placeholder text 'Search ...' and a magnifying glass icon. To the right of the search bar is a 'Filter' section with several checkboxes and a 'Creator' input field containing 'Creator'. Other filter options include 'Creator Organization Name', 'Contributor', 'Contributor Role', 'Issued Date', 'Publisher', and 'Publisher Email'. A blue 'APPLY' button is located at the bottom right of the filter section. The main content area displays four search results, each consisting of a small thumbnail image, the title 'งานแสดงกตเวกิตาจิตแด่ผู้เกียรติยศ ประจำปี 2550', the creator 'Joe, Bryan', the coverage temporal '1998', and the tag 'กิตติมศักดิ์ เทศกาล 1964'.

รูปที่ 3.59: ภาพแสดงหน้าค้นหาข้อมูล

หน้าแสดงข้อมูลการค้นหาเมื่อทำการค้นหาข้อมูลจากหน้าแรก (รูปที่ 3.23 หรือ 3.24) จะทำการแสดงข้อมูลหนังสือที่ตรงกับ Keyword โดยเรียงลำดับของหนังสือที่เกี่ยวข้องกับคำค้นามากที่สุดดังรูปที่ 3.32 เมื่อกดเข้าไปที่รายชื่อหนังสือจะทำการนำทางผู้ใช้ไปยังหน้าดูหนังสือดังรูปที่ 3.33

3.6.11 Document View



รูปที่ 3.60: ภาพแสดงหน้าดูหนังสือ

เมื่อเราค้นหาและเลือกหนังสือ ก็จะมีหน้าหนังสือ (รูปที่ 3.33) ขึ้นมาให้ดูเนื้อหาภายในโดยที่ผู้ใช้สามารถปรับขนาดภาพและสามารถเลือกหน้าที่ต้องการจะเปิดได้และสามารถย้อนหลับไปยังหน้าเดิมได้ที่บุํลูกศรทางด้านซ้ายบน

3.6.12 Manage book

Search results : KMUTT

Image	Title	Creator	Coverage temporal	Tag	Action
	งานแสดงกตเวกิตาอิตเด่อผู้เกี้ยวนอยุ ประจำปี 2550	Creator : Joe, Bryan	Coverage temporal : 1998	Tag : กตเวกิตา เกี้ยวนอยุ 1964	DELETE Edit
	งานแสดงกตเวกิตาอิตเด่อผู้เกี้ยวนอยุ ประจำปี 2550	Creator : Joe, Bryan	Coverage temporal : 1998	Tag : กตเวกิตา เกี้ยวนอยุ 1964	DELETE Edit
	งานแสดงกตเวกิตาอิตเด่อผู้เกี้ยวนอยุ ประจำปี 2550	Creator : Joe, Bryan	Coverage temporal : 1998	Tag : กตเวกิตา เกี้ยวนอยุ 1964	DELETE Edit
	งานแสดงกตเวกิตาอิตเด่อผู้เกี้ยวนอยุ ประจำปี 2550	Creator : Joe, Bryan	Coverage temporal : 1998	Tag : กตเวกิตา เกี้ยวนอยุ 1964	DELETE Edit

รูปที่ 3.61: ภาพแสดงหน้าการจัดการหนังสือที่เพิ่มเข้าสู่ระบบ

ในหน้าของการจัดการหนังสือดังรูปที่ 3.34 จะมีลักษณะคล้ายกับหน้าการค้นหาเพียงแต่ว่าจะมีฟังก์ชันสำหรับการแก้ไขเนื้อหนังสือภายในที่ผู้ใช้เคยกรอกไว้ตอน OCR หนังสือมา เมื่อกดปุ่มลบจะมีหน้าต่างแจ้งเตือนเพื่อถามความแนใจในการลบหนังสือ หรือกดปุ่ม Edit เพื่อทำการเข้าสู่การแก้ไขข้อมูลของหนังสือนั้นๆดังรูปที่ 3.35 - 3.37

3.6.13 Edit Book

งานแสดงผลตัวบ่งชี้ผลผลิตผู้เชี่ยวชาญอาชีวะฯ ประจำปี 2550

Title

Title:

Title Alternative:

Creator

Creator:

Creator Organization name:

Description

Table of contents:

Summary:

Abstract:

Note:

Publisher

Publisher:

Publisher Email:

Contributor

Contributor:

Contributor Role:

Date

Date:

Coverage

Coverage Spatial:

Coverage Temporal:

Rights

Rights:

Rights Access:

Buttons

< BACK

SAVE

NEXT >

รูปที่ 3.62: ภาพแสดงขั้นตอนการแก้ไขหนังสือขั้นที่ 1

SEARCH MANAGE BOOK STATUS Hi, Araya ▾

INSERT BOOK



งานแสดงกตเวกิตาอิตแล่ญูกเซย์ลนวาตุ ประจำปี 2550

Creator : Joe, Bryan
Coverage temporal : 1998
Tag : ไทย เข้มข้น 1964

Identifier

Identifier URL

Identifier ISBN

Source

Source

Relation

Relation + ADD

Thesis

Degree name

Degree level

Degree discipline

Degree grantor

Type

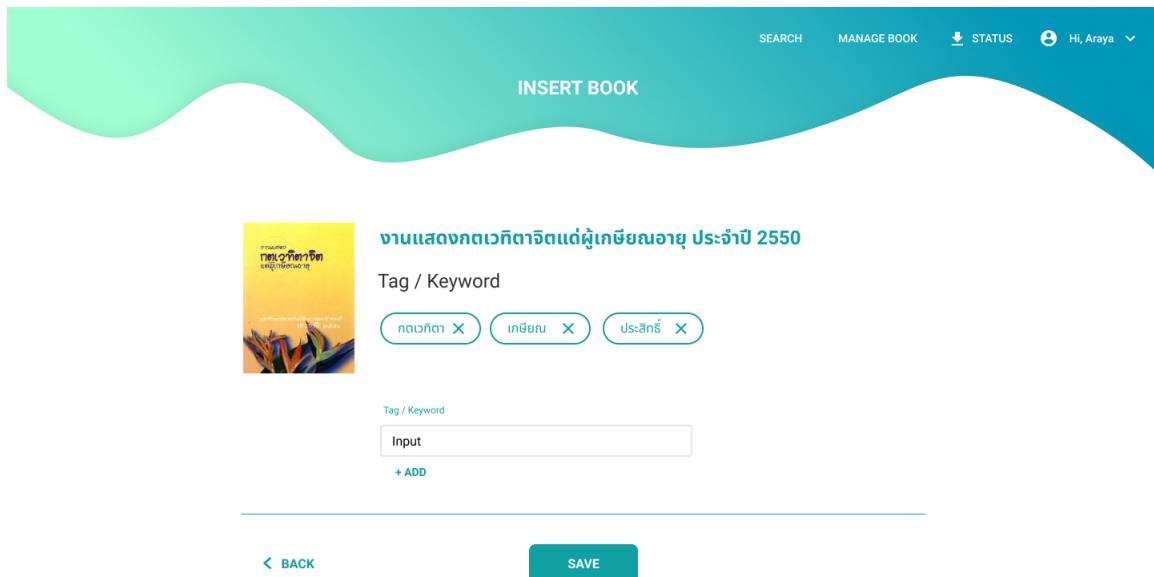
Type Text

Language

Language Thai

< BACK
SAVE
NEXT >

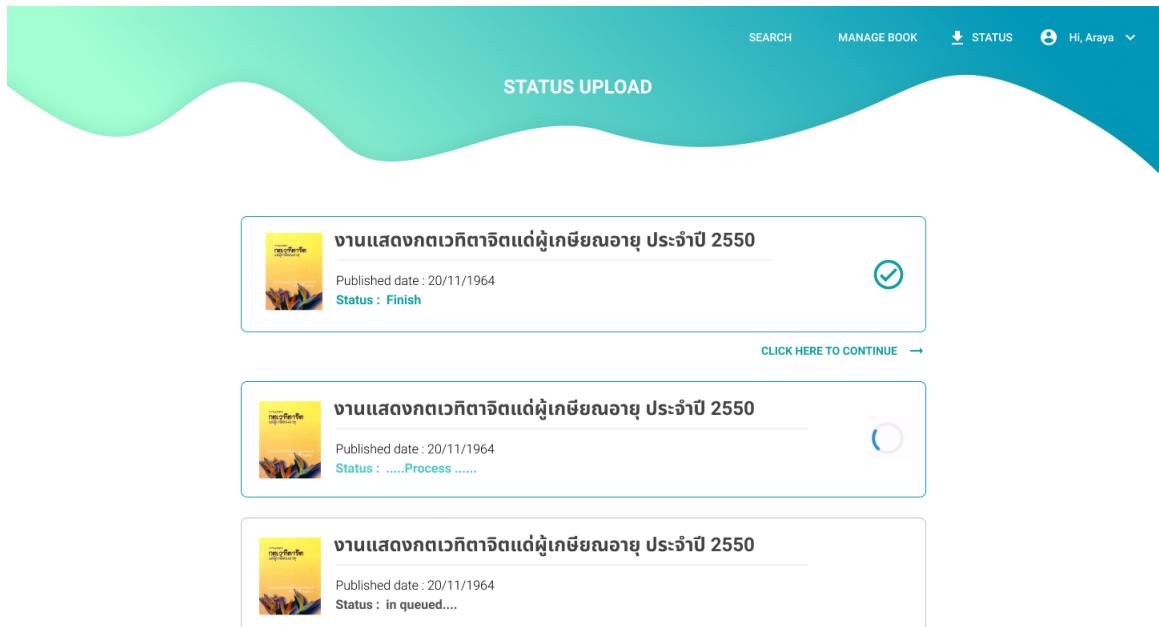
รูปที่ 3.63: ภาพแสดงขั้นตอนการแก้ไขหนังสือขั้นที่ 2



รูปที่ 3.64: ภาพแสดงขั้นตอนการแก้ไขหนังสือขั้นที่ 3

หน้าแก้ไขหนังสือแบบอ打เป็น 3 ขั้นตอนดังรูป 3.35 - 3.37 ซึ่งจะมีให้แก้ไข ข้อมูลที่เคยกรอกไว้ตอนเพิ่มหนังสือเข้ามา โดยจะมีรูปปิกหนังสือและชื่อหนังสืออยบอกรวบรวมกับแก้ไขหนังสือเล่นใหม่อยู่ และในทุกหน้าจะมีปุ่มสำหรับบันทึกในทุกหน้าเพื่อที่จะสามารถบันทึกได้ที่ไม่ต้องรอไปหน้าสุดท้ายเพื่อบันทึกข้อมูล

3.6.14 Upload Status Page



รูปที่ 3.65: ภาพแสดงหน้าการโหลดข้อมูล

จากรูป 3.38 สำหรับผู้ใช้ที่ทำการเพิ่มหนังสือเข้าสู่ระบบจะมีหน้าสำหรับโหลดกรณีที่กดออกมากลังจากผ่านขั้นตอนการเพิ่มหนังสือขึ้นตอนที่ 4 จะสามารถเข้ามาดูสถานะและทำการดำเนินการต่อให้โดยไม่ต้องผ่านการเพิ่มหนังสือเข้าสู่ระบบใหม่

3.6.15 Evaluate Process Design

ในส่วนของการประเมินผลการทำงานนั้นจะแบ่งออกเป็น 3 ส่วนคือการออกแบบ User Interface ส่วนของการเตรียมข้อมูลรูปภาพ จะช่วยให้การทำ OCR มีประสิทธิภาพมากเท่าไร และส่วนของระบบการค้นหา โดยในส่วนของ OCR จะทำการประเมินจากการเลือกเช็คคำจาก 2 หน้าของแหล่งหนังสือมาเช็คว่าแหล่งหน้ามีคำพิเศษเท่าไร โดยจะเลือกวัดหนังสือทั้งหมด 5 เล่มแบบสุ่มและเทียบการเตรียมข้อมูลรูปภาพ ว่าทำแบบไหนได้ผลลัพธ์แบบไหนมาก

ตารางที่ 3.1 ตารางประเมินการทำ OCR

ตารางประเมินการทำ OCR				
หนังสือ	หน้า	จำนวนคำทั้งหมด	คำที่ผิด(%)	คำเกิน(คำ)

ระบบการค้นหา จะเช็คโดยให้ผู้ใช้เป็นผู้ประเมินว่าได้รับหนังสือตรงตามที่ต้องการหรือไม่โดยจะให้เจ้าหน้าที่บรรณาธิการคัดเลือกหนังสือจำนวน 3 เล่มที่คาดหวังว่าจะขึ้นมาเมื่อค้นหาทั้งหมด 10 ครั้ง

ตารางที่ 3.2 ตารางประเมินระบบการค้นหา

ตารางประเมินระบบการค้นหา		
คำค้นหา	หนังสือที่คาดหวัง	การค้นหา
		<p>คะแนน 5 ระดับ</p> <p>5 = ค้นหาหนังสือได้ตรงตามที่ต้องการทั้งหมด และไม่มีหนังสือที่ไม่เกี่ยวข้องกับคำค้นหาขึ้นมา</p> <p>4 = ค้นหาหนังสือได้ตรงตามที่ต้องการทั้งหมด และมีหนังสือที่ไม่เกี่ยวข้องกับคำค้นหาขึ้นมาบ้าง</p> <p>3 = ค้นหาหนังสือได้ตรงตามที่ต้องการบางส่วน และมีหนังสือที่ไม่เกี่ยวข้องกับคำค้นหาขึ้นมาบ้าง</p> <p>2 = ค้นหาหนังสือได้ตรงตามที่ต้องการบางส่วน และมีหนังสือที่ไม่เกี่ยวข้องกับคำค้นหาขึ้นมาบ้าง</p> <p>1 = ไม่มีหนังสือที่ต้องการขึ้นมาในผลลัพธ์ หรือขึ้นหนังสือทุกเล่ม</p>

ตารางที่ 3.3 ตารางประเมินความพึงพอใจการออกแบบ UX/UI

ตารางประเมินการออกแบบ UX/UI					
	4	3	2	1	คะแนนที่ได้
ความสมบูรณ์ของข้อมูล	ข้อมูลมีความสมบูรณ์ชัดเจนทำให้เข้าใจความหมายที่ต้องการจะสื่อได้เป็นอย่างดี	มีข้อมูลที่ชัดเจน และแม่นยำในบางครั้ง และสามารถแสดงความหมายที่ต้องการจะสื่อได้บ้าง	ข้อมูลมีความแม่นยำและชัดเจนบ้าง	มีข้อมูลที่ไม่ชัดเจน ไม่ครบ สื่อความหมายได้ไม่ดี	
การออกแบบ	มีการออกแบบที่เน้นความสำคัญและจัดวางองค์ประกอบอย่างสวยงาม เช่น การเคลื่อนไหว(animation) ได้อย่างเหมาะสม	มีการจัดหน้า และองค์ประกอบที่ให้เห็นถึงความสำคัญของเนื้อหา มีการใช้การเคลื่อนไหว(animation) บ้าง	การวางแผน และการจัดต่อองค์ประกอบมีความไม่เหมาะสม มีการใช้การเคลื่อนไหว(animation) เข้ามาช่วยบ้าง	การวางแผนและการจัดองค์ประกอบมีความไม่เหมาะสม และไม่มีการใช้ การเคลื่อนไหว(animation) เข้ามาช่วยในการใช้งาน	
การใช้งาน	ผู้ใช้สามารถใช้งานบุ่มหรือเบี่ยงหน้าต่างๆ ได้อย่างง่ายดาย แต่มีลักษณะพิเศษ หน้าอย่างมากหนึ่งลักษณะไม่มีเลย	ผู้ใช้สามารถใช้งานบุ่มหรือเบี่ยงหน้าต่างๆ ได้อย่างง่ายดาย แต่มีลักษณะพิเศษ หน้าอย่างมากสองลักษณะไม่มีเลย	ผู้ใช้มีความสับสนในการใช้บุ่ม หรือการเบี่ยงหน้าต่างๆ บางครั้ง และมีลักษณะพิเศษหน้าอย่างมากสามลักษณะ	ผู้ใช้เกิดความสับสนในการใช้บุ่ม หรือลักษณะพิเศษที่เบี่ยงหน้าต่างๆ	
การใช้ภาษา	มีการใช้คำพิเศษหรือภาษาที่ไม่เหมาะสมอย่างมาก 1 จุด	มีการใช้คำพิเศษหรือภาษาที่ไม่เหมาะสมอย่างมาก 2 จุด	มีการใช้คำพิเศษหรือภาษาที่ไม่เหมาะสมอย่างมาก 3 จุด	มีการใช้คำพิเศษหรือภาษาที่ไม่เหมาะสมมากกว่า 4 จุด	

ตารางที่ 3.4 ตารางประเมินการทดสอบเว็บไซต์

ตารางประเมินเว็บไซต์			
เกณฑ์การประเมิน	ผลลัพธ์		หมายเหตุ
	ผ่าน	ไม่ผ่าน	
1. สามารถเข้าสู่ระบบและออกจาก ระบบได้			
2. สามารถเพิ่มนั้งสือเข้าสู่ระบบได้			
3. สามารถแก้ไขรายละเอียดหนังสือที่อยู่ในระบบได้			
4. สามารถตรวจสอบและแก้ไขคำที่เพิ่มเข้ามาในระบบในขั้นตอนเพิ่มนั้งสือได้			
5. สามารถลบหนังสือที่อยู่ในระบบได้			
6. สามารถค้นหาข้อมูลหนังสือภายในระบบได้			
7. สามารถเรียกดูหนังสือที่ต้องการได้			

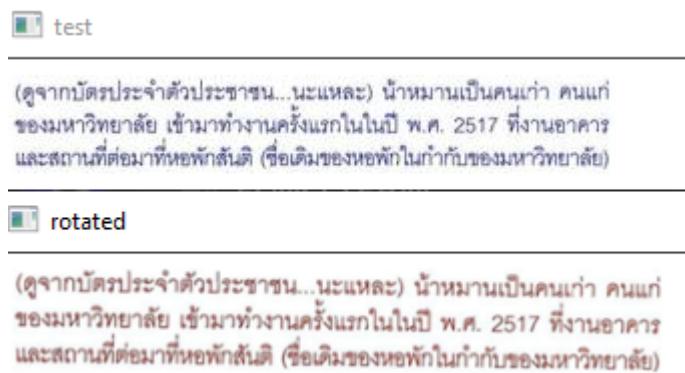
บทที่ 4 ผลการดำเนินงาน

การดำเนินงานของโปรเจคนี้จะแบ่งออกมาเป็นทั้งหมด 3 ส่วน โดยส่วนแรกคือส่วนของการจัดเก็บข้อมูลเข้าสู่ระบบโดยนำรูปภาพได้ที่ได้รับมาผ่านกระบวนการ การเตรียมข้อมูลรูปภาพ ก่อนจะนำไปผ่านกระบวนการ OCR และการเตรียมข้อมูลตัวอักษร ก่อนจะถูกเก็บข้อมูล ในระบบ ส่วนที่สองการค้นหาข้อมูล เป็นการค้นหาแบบ IR (Information retrieval) ที่จะนำໄไปโมเดล Word2Vec เข้ามาช่วยในการค้นหาคำที่มีความสัมพันธ์ใกล้เคียงกับคำค้นหา และนำคะแนน TF-IDF มาใช้เป็นคะแนนในการค้นหา และส่วนสุดท้ายคือส่วนของการทำแพลตฟอร์มเว็บไซต์ ซึ่งในการประเมินผลการดำเนินงานนั้นเราจะทำการประเมินในส่วนแรก โดยการประเมินความถูกต้องของการทำ OCR จะมีเจ้าหน้าที่ปรับปรุงรักษาภาระที่เกิดขึ้น ซึ่งเกณฑ์ที่กำหนดในส่วนของความถูกต้องในการทำ OCR อยู่ที่ 75 % และความแม่นยำในการค้นหาอยู่ที่ 75 %

4.1 ผลลัพธ์ที่ได้จากการแปลงข้อมูลรูปภาพให้เป็นข้อมูลดิจิทัล

4.1.1 ผลลัพธ์ที่ได้จากการหมุนรูปภาพ

ผลลัพธ์จากการหมุนภาพตัวอักษรทั้ง 978 ภาพ มีความคลาดเคลื่อนทั้งหมด 7.98% ที่ยังไม่สามารถหมุนภาพให้ตรงตั้งภาพที่ [4.1](#) และทำให้บางภาพหายไป เนื่องจากว่าบรรทัดตัวอักษรอาจจะมีสีระที่ไม่สามารถทำการอ่านให้ถูกต้องได้



รูปที่ 4.1: ภาพแสดงผลลัพธ์การหมุนรูปที่ผิดพลาด

4.1.2 ผลการเปรียบเทียบประสิทธิภาพในการทำ OCR ของ การทำการเตรียมข้อมูลรูปภาพ แต่ละแบบ

จากการทดสอบประสิทธิภาพของการทำการเตรียมข้อมูลรูปภาพทั้งสองแบบ พบร่วมกันว่า การทำการเตรียมข้อมูลรูปภาพ แบบแรกนั้นมีจำนวนคำผิดน้อยกว่า แต่เมื่อจำนวนคำที่ไม่สามารถแปลงเป็นดิจิทัลมากถึง 32.71% ตั้งตารางที่ [4.1](#) ซึ่งต่างจากการทำการเตรียมข้อมูลรูปภาพ แบบที่ 2 ที่มีค่าความถูกต้องของคำ 74.74 % ตั้งตาราง [4.2](#)

4.1.2.1 แบบที่ 1 การใช้การคัดเลือกข้อมูล, การหมุน, การลบรูปภาพ, การลบเส้น และการจัดกลุ่ม

ตารางที่ 4.1 ตารางประเมินการทำการเตรียมข้อมูลรูปภาพแบบที่ 1

หนังสือ	หน้า	จำนวนคำทั้งหมด	จำนวนคำพิเศษที่ต้องลบ	เปอร์เซ็นต์คำพิเศษที่ต้องลบ(%)	จำนวนคำเกิน	จำนวนคำที่ไม่สามารถแปลงเป็นตัวอักษร	เปอร์เซ็นต์คำที่ไม่สามารถแปลงเป็นตัวอักษร(%)
กตเวทิตาปี 2542	15	4	4	100 %	0	0	0 %
	29	252	14	5.56 %	46	2	0.79 %
กตเวทิตาปี 2556	15	242	33	13.64 %	2	1	0.41 %
	29	257	20	7.78 %	3	10	3.89 %
รายงานประจำปี 2544	15	47	3	6.38 %	2	34	72.34 %
	29	585	39	6.67 %	3	308	52.65 %
รายงานประจำปี 2553	15	68	0	0 %	0	68	100 %
	29	596	17	2.85 %	8	340	57.05 %
รายงานประจำปี 2549	15	155	53	34.19 %	42	45	29.03 %
	29	304	22	7.24 %	20	13	4.28 %
	total	2510	205	8.17 %	126	821	32.71 %

4.1.2.2 แบบที่ 2 ใช้การลบพื้นหลัง

ตารางที่ 4.2 ตารางประเมินการทำการเตรียมข้อมูลรูปภาพแบบที่ 2

หนังสือ	หน้า	จำนวนคำทั้งหมด	จำนวนคำผิดที่ตรวจพบ	เปอร์เซ็นต์คำผิดที่ตรวจพบ(%)	จำนวนคำเกิน	จำนวนคำที่ไม่สามารถแปลงเป็นดิจิทัล	เปอร์เซ็นต์คำที่ไม่สามารถแปลงเป็นดิจิทัล(%)
กตdevทิศปี 2542	15	4	4	100%	0	0	0%
	29	252	30	11.9%	6	9	3.57%
กตdevทิศปี 2556	15	242	42	17.36%	2	48	19.83%
	29	257	54	21.01%	2	62	24.12%
รายงานประจำปี 2544	15	47	27	57.45%	5	5	10.64%
	29	585	101	17.26%	23	0	0%
รายงานประจำปี 2553	15	68	30	44.12%	7	0	0%
	29	596	85	14.26%	30	0	0%
รายงานประจำปี 2549	15	155	57	36.77%	14	4	2.58%
	29	304	76	25%	7	0	0%
	total	2510	506	20.16%	96	128	5.1%

ซึ่งจำนวนคำต่างๆในตารางนั้นถูกบันทึกและตรวจพบด้วยคน สรุว่าจำนวนคำเกินนั้นจะเป็นคำที่คาดว่าเกิดจากภาษาอ่านพยัญชนะแยกกับสาระผิด หรือบางที่มีการอ่านบรรทัดเดิมมาเข้าทำให้มีตัวอักษร หรือคำอื่นๆที่ไม่มีอยู่ในรูปเกินอกรกมา สรุว่าจำนวนคำที่ไม่สามารถแปลงเป็นดิจิทัลได้นั้นจะเป็นคำที่มีอยู่ในรูปภาพแต่ไม่ผ่านการแปลงเป็นรูปแบบดิจิทัล เนื่องจาก OCR ไม่สามารถแปลงรูปหรือคำอกรกมาเป็นตัวหนังสือได้ หรือเกิดจากการทำ image processing หรือการเตรียมข้อมูลรูปภาพส่งผลให้มีคำบางคำหายไป จากผลลัพธ์ตารางที่ 4.1 และ 4.2 ทำให้ผู้จัดทำเลือกการทำ image processing แบบที่ 2 มาใช้ในการเตรียมรูปภาพก่อนนำไป OCR ถึงแม้ว่าจะมีจำนวนคำผิดมากกว่าในแบบที่ 1 แต่จำนวนคำที่ไม่ถูกอ่านในการทำ image processing แบบที่ 1 มีมากถึง 32.71 % ซึ่งจะทำให้ผู้ใช้งานจะมีภาระในการตรวจสอบคำมากกว่าแบบที่ 2

4.1.3 ผลการเปรียบเทียบข้อมูล training set 2 ชุดสำหรับการทำ OCR

โดยตอนที่เลือกข้อมูลที่ใช้การทำ OCR ทางผู้จัดได้พบว่ามีชุดข้อมูลที่ทาง Tesseract ได้ปล่อยออกมาในเว็บไซต์หลักซึ่งเป็นชุด training set ในปี 2016 เป็นชุดที่ 1 และมีข้อมูลชุด training set ที่ได้มีการอ้างอิงมาว่าเป็นชุดข้อมูลที่ดีที่สุดในปี 2019 ที่ได้รับการประเมินจาก Google เป็นชุดที่ 2 ซึ่งทำให้ผู้จัดทำนำชุดข้อมูล training set ทั้งสองชุดนี้มาราทำ การเปรียบเทียบว่าชุดไหนมีประสิทธิภาพมากกว่ากัน จากการเปรียบข้อมูลทั้งสองชุดระหว่าง ชุดข้อมูลปี 2016 (ชุดที่ 1) ดังตารางที่ 4.3 ชุดข้อมูลปี 2019 (ชุดที่ 2) ดังตารางที่ 4.4 กับ พบว่าประสิทธิภาพของข้อมูลชุดที่ 1 มีความถูกต้องอยู่ที่ 76.61 % ซึ่งมีจำนวนคำผิดสูงกว่าประมาณ 2% ความถูกต้องอยู่ที่ เมื่อเทียบกับข้อมูลชุดที่ 2 ที่มีความถูกต้องอยู่ที่ 77.41 % แต่ว่ามีจำนวนคำเกินที่ต่างกันเป็นเท่าตัว และมีจำนวนคำที่ไม่สามารถแปลงเป็นดิจิทัลมากกว่า 28 คำ

ตารางที่ 4.3 ตารางประเมินข้อมูลชุด training set ที่ 1

หนังสือ	หน้า	จำนวนคำทั้งหมด	จำนวนคำผิด ที่ตรวจพบ	เปอร์เซ็นต์ คำผิดที่ตรวจ พบ(%)	จำนวนคำเกิน	จำนวนคำที่ ไม่สามารถ แปลงเป็น ดิจิทัล	เปอร์เซ็นต์คำ ที่ไม่สามารถ แปลงเป็น ดิจิทัล(%)
กตเวทิตาปี 2542	15	4	2	50%	0	2	50%
	29	252	34	13.49%	12	4	1.59%
กตเวทิตาปี 2556	15	242	37	15.29%	0	49	20.25%
	29	257	47	18.29%	2	45	17.51%
รายงานประจำปี 2544	15	47	40	85.11%	0	4	8.51%
	29	585	78	13.33%	11	15	2.56%
รายงานประจำปี 2553	15	68	44	64.71%	0	0	0%
	29	596	76	12.75%	9	12	2.01%
รายงานประจำปี 2549	15	155	44	28.39%	15	1	0.65%
	29	304	53	17.43%	34	0	0%
	total	2510	455	18.13%	83	132	5.26%

ตารางที่ 4.4 ตารางประเมินข้อมูลชุด training set ที่ 2

หนังสือ	หน้า	จำนวนคำทั้งหมด	จำนวนคำผิด ที่ตรวจพบ	เปอร์เซ็นต์ คำผิดที่ตรวจ พบ(%)	จำนวนคำเกิน	จำนวนคำที่ ไม่สามารถ แปลงเป็น ดิจิทัล	เปอร์เซ็นต์คำ ที่ไม่สามารถ แปลงเป็น ดิจิทัล(%)
กตเวทิตาปี 2542	15	4	4	100%	0	0	0%
	29	252	40	15.87%	20	10	3.97%
กตเวทิตาปี 2556	15	242	46	19.01%	11	44	18.18%
	29	257	32	12.45%	2	62	24.12%
รายงานประจำปี 2544	15	47	26	55.32%	0	4	8.51%
	29	585	63	10.77%	7	28	4.79%
รายงานประจำปี 2553	15	68	36	52.94%	9	2	2.94%
	29	596	65	10.91%	60	2	0.34%
รายงานประจำปี 2549	15	155	43	27.74%	30	8	5.16%
	29	304	52	17.11%	34	0	0%
	total	2510	407	16.22%	173	160	6.37%

จากผลลัพธ์ตารางที่ 4.3 และ 4.4 ผู้จัดทำเลือกใช้ข้อมูลชุดที่ 1 เนื่องจากมีการแปลงข้อมูลเป็นดิจิทัลได้ครอบคลุมกว่า และมีจำนวนคำเกินน้อยกว่า เมื่อเทียบกับข้อมูลชุดที่ 2

4.1.4 ประสิทธิภาพการแก้ไขคำผิด

จากการเปรียบข้อมูลที่ไม่ถูกแก้ไขคำผิดในตารางที่ 4.5 กับข้อมูลที่ผ่านระบบการแก้ไขคำผิดในตารางที่ 4.3 พบว่าการใช้ระบบการแก้ไขคำผิดทำให้คำผิดที่เกิดขึ้นลดลงประมาณ 2% ทำให้เปอร์เซ็นต์ความถูกต้องหลังการแก้ไขคำผิดอยู่ที่ 76.61 % จาก 74.75 %

ตารางที่ 4.5 ตารางประเมินข้อมูลชุด training set ที่ 1 ที่ไม่ผ่านการแก้ไขคำผิด

หนังสือ	หน้า	จำนวนคำทั้งหมด	จำนวนคำผิดที่ต้องตรวจ	เปอร์เซ็นต์คำผิดที่ต้องตรวจ (%)	จำนวนคำเกิน	จำนวนคำที่ไม่สามารถแปลงเป็นดิจิทัล	เปอร์เซ็นต์คำที่ไม่สามารถแปลงเป็นดิจิทัล (%)
กตเวทิตาปี 2542	15	4	4	100%	0	0	0%
	29	252	30	11.9%	6	9	3.57%
กตเวทิตาปี 2556	15	242	42	17.36%	2	48	19.83%
	29	257	54	21.01%	2	62	24.12%
รายงานประจำปี 2544	15	47	27	57.45%	5	5	10.64%
	29	585	101	17.26%	23	0	0%
รายงานประจำปี 2553	15	68	30	44.12%	7	0	0%
	29	596	85	14.26%	30	0	0%
รายงานประจำปี 2549	15	155	57	36.77%	14	4	2.58%
	29	304	76	25%	7	0	0%
	total	2510	506	20.16%	96	128	5.1%

4.2 ผลลัพธ์จากการค้นหา

ในการประเมินการค้นหา ผู้จัดทำได้ทำการเปลี่ยนการประเมินการจากตารางประจำระบบการค้นหา เป็นการประเมินโดยการใช้ตาราง Confusion matrix แทนเนื่องจากจะให้ค่าสถิติที่ชัดเจนมากกว่าตารางแบบเดิม เพื่อหา Precision และ Recall ของการทำงาน การค้นหาในครั้งนี้มีคำค้นหาทั้งหมด 16 คำ หานักหนังสือ ห้องหมด 44 เล่ม โดยที่แต่ละคำบรรณารักษ์จะเป็นคนกำหนดว่าผลลัพธ์ที่ออกควรเป็นเล่มไหน

ตารางที่ 4.6 ตารางแสดงการทำ Confusion matrix

	TRUE	FALSE
POSITION	TP	FP
NEGATIVE	FN	TN

ตารางที่ 4.7 ตารางแสดงรายละเอียด Confusion matrix

TP =	จำนวนหนังสือที่ค้นหาได้ถูกต้อง	Recall	= การค้นหามีความครอบคลุมมากแค่ไหน
FP =	จำนวนหนังสือที่ไม่มีคำ แต่เจอในการค้นหา	Precision	= การค้นหาที่ออกมานั้นมีความถูกต้องมากเท่าไร
FN =	จำนวนหนังสือที่ถูกต้องแต่ค้นหาไม่เจอ	Accuracy	= การค้นหาสามารถค้นหาผลลัพธ์ได้แม่นยำมากเท่าไร
TN =	จำนวนหนังสือที่ไม่มีคำ และไม่ถูกค้นเจอ		

$$\text{Recall} = \frac{TP}{(TP+FN)} , \text{Precision} = \frac{TP}{(TP+FP)} , \text{Accuracy} = \frac{(TP+TN)}{(TP+FP+FN+TN)}$$

การทดสอบการค้นหาในครั้งนี้ไม่มีการซ่วยแก้คำจากเจ้าหน้าที่ บรรณาธิการแต่เป็นการทดสอบโดยใช้ระบบทั้งหมด ได้ผลลัพธ์ออกมายังดังตาราง 4.8

ตารางที่ 4.8 ตารางแสดงผลการค้นหาจากชุดข้อมูล 44 เล่มที่ไม่ผ่านการแก้ไขคำผิดจากมนุษย์

	TRUE	FALSE
POSITION	13	166
NEGATIVE	57	458

$$\text{Recall} = 18.57\% , \text{Precision} = 7.26\% , \text{Accuracy} = 67.87\%$$

หลังจากคำนวณค่าจากตารางได้ค่า Recall 18.57 % ค่า Precision อยู่ที่ 7.26 % และค่า Accuracy อยู่ที่ 67.87 % จากผลลัพธ์ดังกล่าว ทำให้ผู้จัดทำได้ลองทำการทดสอบรอบที่ 2 เปรียบเทียบข้อมูลที่ได้รับการแก้คำจากมนุษย์ โดยจะใช้ชุดข้อมูลทั้งหมด 6 เล่ม และใช้คำค้นหาเดิม เปรียบเทียบระหว่างข้อมูล 6 ชุดที่ได้รับการแก้ไขคำ ในโมเดล Word2Vec และข้อมูล 6 เล่มที่ไม่ผ่านการแก้ไขคำในระบบ Word2Vec ซึ่งได้ผลลัพธ์ดังตารางที่ 4.9 และ ตารางที่ 4.10

ตารางที่ 4.9 ตารางแสดงผลการค้นหาจากชุดข้อมูล 6 เล่มที่ไม่ผ่านการแก้ไขคำผิดจากมนุษย์

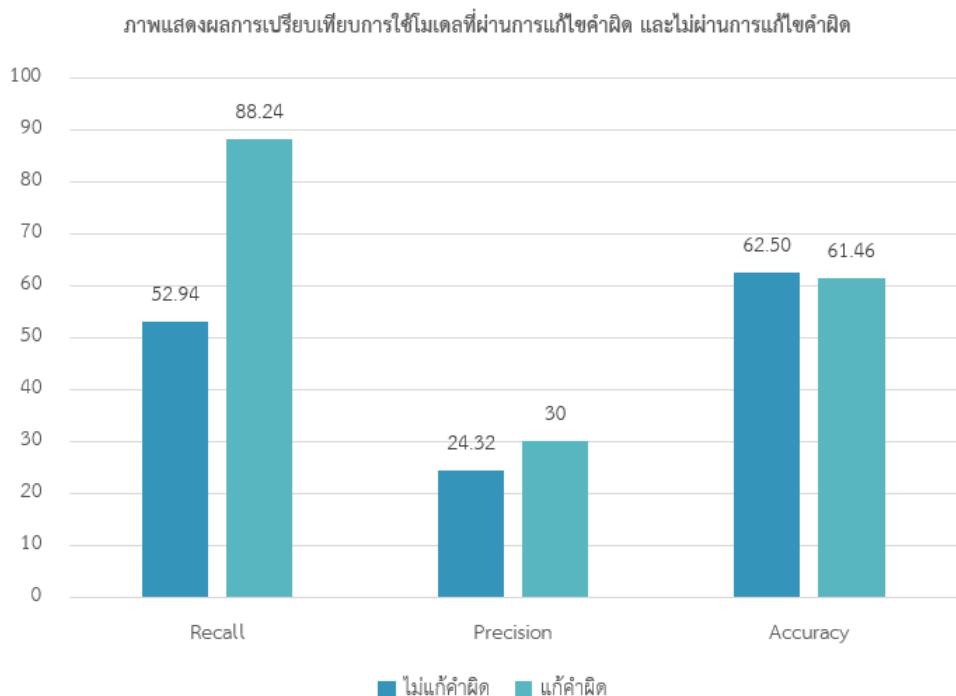
	TRUE	FALSE
POSITION	9	28
NEGATIVE	8	51

$$\text{Recall} = 52.94\% , \text{Precision} = 24.32\% , \text{Accuracy} = 62.5\%$$

ตารางที่ 4.10 ตารางแสดงผลการค้นหาจากชุดข้อมูล 6 เล่มที่ผ่านการแก้ไขคำผิดจากมนุษย์

	TRUE	FALSE
POSITION	15	35
NEGATIVE	2	44

$$\text{Recall} = 88.24\% , \text{Precision} = 30\% , \text{Accuracy} = 61.46\%$$



รูปที่ 4.2: ภาพแสดงผลการเปรียบเทียบการใช้โมเดลที่ผ่านการแก้ไขคำพิด และไม่ผ่านการแก้ไขคำพิด

จากผลลัพธ์ในตารางที่ 4.9 ได้ค่า Precision อยู่ที่ 24.32 % ค่า Recall 52.94 % และมี Accuracy 62.5 % หลังจากการแก้ไขคำและทำการทดลองมีค่า Precision อยู่ที่ 30 % ค่า Recall 88.24 % และ ค่า Accuracy อยู่ที่ 61.46 % จะเห็นได้ว่ามีค่า Precision และ Recall สูงขึ้น แต่เมื่อค่า Accuracy ต่ำลง เมื่อทางผู้จัดทำได้ตรวจสอบระบบการค้นหาพบว่าระบบการค้นหาค่าความสัมพันธ์ Word2Vec ยังไม่เดินกัน เนื่องจาก Word2Vec ของชุดข้อมูล 6 เล่นบันนมีจำนวน corpus หรือชุดข้อมูลที่นำไปใช้เรียนอย่างเกินไปทำให้ค่าความสัมพันธ์ที่ได้จาก Word2Vec มีค่าใกล้เคียงกัน ดังภาพที่ 4.3 ซึ่ง ทำให้ได้คำที่ไม่เกี่ยวข้องเข้ามาใช้ในการค้นหา อย่างเช่น 6 หรือ 2009 ที่ถูกดึงมาใช้หัวข้อที่ไม่มีความเกี่ยวข้อง ส่งผลให้มีหนังสือเล่มอื่นดิตามด้วย ซึ่งสำหรับคำค้นหาคำนี้ถ้า เทียบในระบบใหญ่แล้ว ถ้าคำนี้ไม่ถูกอ่านผิดระบบใหญ่จะสามารถแยกความสำคัญได้มากกว่าระบบเล็ก

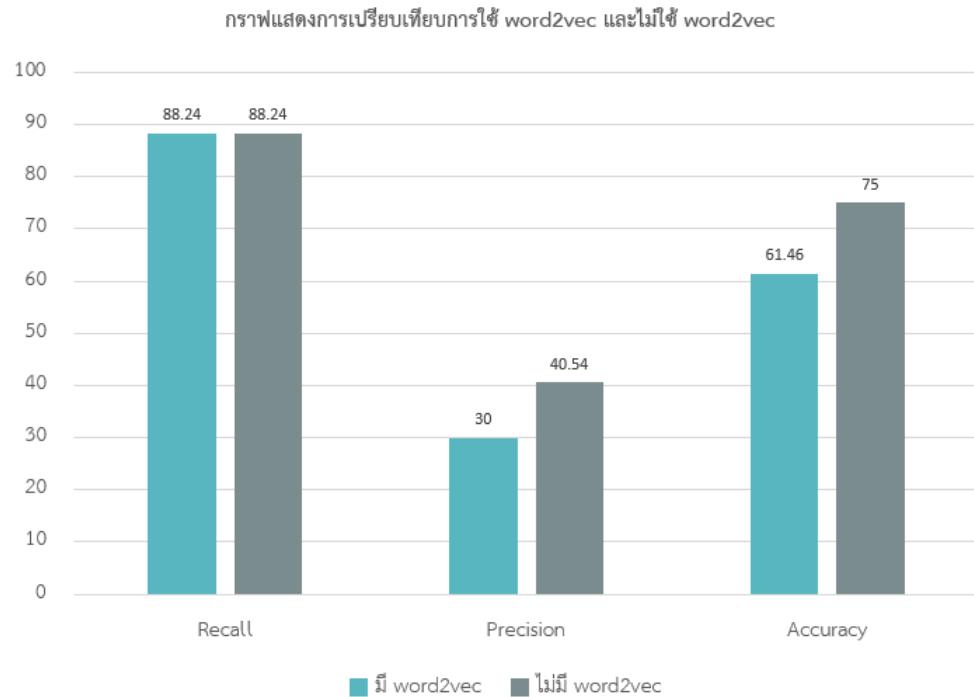
['น้ำค้าง', 'เอนกิมพด']

```
[{"key": "\u0e04\u0e17\u0e0e\u0e14", "value": [{"token": "\u0e04\u0e17\u0e0e\u0e14", "score": 0.9993730187416077}, {"token": "6", "score": 0.9993718266487122}, {"token": "\u0e04", "score": 0.9993717670440674}, {"token": "\u0e04\u0e17\u0e0e", "score": 0.9993667602539062}, {"token": "\u0e04\u0e14", "score": 0.999365508556366}, {"token": "IEEE", "score": 0.9993652701377869}, {"token": "\u0e04\u0e17\u0e0e\u0e14", "score": 0.9993649125099182}, {"token": "Japan", "score": 0.9993644952774048}, {"token": "\u0e04\u0e17\u0e0e", "score": 0.9993644952774048}, {"token": "\u0e04\u0e14", "score": 0.9993643760681152}], {"key": "\u0e04\u0e17\u0e0e\u0e14", "value": [{"token": "2009", "score": 0.9990187883377075}, {"token": "\u0e04", "score": 0.9990177750587463}, {"token": "\u0e04", "score": 0.9990175366401672}, {"token": "\u0e04\u0e14", "score": 0.999016284942627}, {"token": "The", "score": 0.9990156888961792}, {"token": "14", "score": 0.9990139603614807}, {"token": "\u0e04", "score": 0.9990135431289673}, {"token": "\u0e04\u0e14", "score": 0.9990133047103882}, {"token": "\u0e04\u0e14", "score": 0.9990126490592957}, {"token": "on", "score": 0.9990124106407166}]}]
```

รูปที่ 4.3: ภาพแสดงค่าแนวการค้นหาคำใหม่ๆ ออกจากโมเดล word2vec

ซึ่งถ้าเปรียบเทียบค่า Recall ในแต่ละโมเดลการแก้ไขคำผิด และโมเดลที่ไม่ได้แก้ไขคำผิด จะพบว่าผลลัพธ์การแก้ไขคำผิดจะช่วยให้ได้หนังสือที่คร่าวๆ คลื่นกับผลลัพธ์มากกว่าโมเดล Word2Vec ทำให้ได้หนังสือที่ไม่เกี่ยวข้องเพิ่มมากขึ้น

เช่นกัน ส่งผลให้มีค่าความแม่นยำ หรือค่า Accuracy ลดลง ดังนั้นทางผู้จัดทำได้ลองทำการเปรียบเทียบการค้นหาโดยใช้ Word2Vec และไม่ใช้ Word2Vec เข้ามาช่วย ได้ผลลัพธ์อุอกมาดังภาพที่ 4.4



รูปที่ 4.4: ภาพแสดงผลการเปรียบเทียบการใช้ word2vec และไม่ใช้ word2vec

การค้นหาที่ใช้ Word2Vec จะเป็นการค้นหาที่นำคำค้นหาเข้ามายadel Word2Vec หลังจากนั้นนำคำที่ได้ไปค้นหาคุณ TF-IDF เพื่อนำไปหาหางสือที่มีคีคะแนนความสัมพันธ์กับคำค้นหาโดยเรียงลำดับจากมากไปน้อย ส่วนการค้นหาที่ไม่ใช้ Word2Vec จะเป็นการค้นหาโดยการนำคำค้นหาไปหาคุณ TF-IDF โดยไม่ผ่านมายadel Word2Vec จากภาพที่ 4.4 จะเห็นได้ว่า การใช้ Word2Vec ทำให้ประสิทธิภาพในการค้นหาลดลง ถึงแม้ผลลัพธ์ความครอบคลุมของหางสือที่ถูกต้อง (recall) จะมีค่าเท่ากัน แต่ก็มีจำนวนหางสือที่ไม่เกี่ยวข้องเข้ามายอย่างกว่าเมื่อเทียบการค้นหาแบบบีช TF-IDF เพียงอย่างเดียว

นอกจากนี้ทางผู้จัดทำได้ทำการเปรียบเทียบประเภทของคำที่ใช้ในการค้นหากับการใช้ไม่ymodel Word2Vec พบว่าส่วนใหญ่จะเป็นข้อมูลที่เป็นคำเฉพาะที่มีการผิดพลาดเยอ ซึ่งเมื่อได้ลองนำข้อมูลค้นหาที่เป็นคำเฉพาะออกพบร่วมค่า Accuracy เพิ่มขึ้นเป็น 79.17 % ค่า recall 87.5 % และค่า Precision 43.75 % รวมถึงจำนวนจากตารางที่ 4.12 และเมื่อเปรียบเทียบกับตารางที่ 4.11 จะเห็นได้ว่าการแก้ไขคำผิดส่งผลกับการค้นหาเป็นอย่างมาก

ตารางที่ 4.11 ตารางแสดงผลการค้นหาจากชุดข้อมูล 6 เล่มที่ไม่ผ่านการแก้ไขคำผิดจากมนุษย์แบบไม่มีคำเฉพาะ

	TRUE	FALSE
POSITION	4	13
NEGATIVE	4	27

$$\text{Recall} = 50\% , \text{Precision} = 23.53\% , \text{Accuracy} = 64.6\%$$

ตารางที่ 4.12 ตารางแสดงผลการค้นหาจากชุดข้อมูล 6 เล่มที่ไม่ผ่านการแก้ไขคำพิจารณานุชร์แบบไม่มีคำเฉพาะ

	TRUE	FALSE
POSITION	7	9
NEGATIVE	1	31

$$\text{Recall} = 87.5\% , \text{Precision} = 43.75\% , \text{Accuracy} = 79.17\%$$

4.2.1 ผลการเปรียบเทียบประสิทธิภาพเวลาในการค้นหา

ทางผู้จัดทำได้ทำการทดสอบการค้นหาโดยกำหนด คำ 3 คำค้น ให้กับเจ้าหน้าที่บรรณารักษ์และบุคลากรรวม 2 คน โดยกำหนดขอบเขตในการค้นหาหนึ่งสื้อ 6 เล่ม ซึ่งบุคลากรนอกที่ใช้ระบบค้นหาของปูรเจคนี้ คันที่ 1 สามารถระบุหนังสือที่มีเนื้อหางบค้นหาได้ภายใน 1 นาที ในการค้นหา 3 คำค้นหา และเจอน้ำที่มีคำค้นหา 3 คำภายใน 11 นาที และคนที่ 2 สามารถระบุหนังสือที่มีเนื้อหางบค้นหาได้ภายใน 1 นาทีในการค้นหา 3 คำค้นหา และเจอน้ำที่มีคำค้นหา 3 คำภายใน 9 นาที เมื่อเปรียบเทียบกับเจ้าหน้าที่บรรณารักษ์ที่เป็น คันหา ด้วยวิธีการปกติ ทำให้ต้องลองสุ่มหนังสือทุกเล่มจนกว่าจะเจอน้ำที่มีคำค้นหาทั้ง 3 ชื่่อให้ใช้เวลาไปถึง 11 นาที ถ้าดูโดยภาพรวมแล้ว การใช้เว็บกับการค้นหาจากหนังสืออาจจะดูใช้เวลาไก้สักคียงกัน แต่หลังจากที่สอบถามเจ้าหน้าที่บรรณารักษ์พบ ว่า ส่วนใหญ่เสียเวลาให้กับ การเปิดหนังสือสุมเล่มประมาณ 6 นาที เนื่องจากไม่ว่าคราวจะเปิดจากเล่มไหน

4.3 ผลลัพธ์จากการดำเนินงานในส่วนของการทำเว็บไซต์

4.3.1 การประเมินการใช้งานของเว็บไซต์

ตารางที่ 4.13 ตารางแสดงผลการประเมินการทดสอบเว็บไซต์

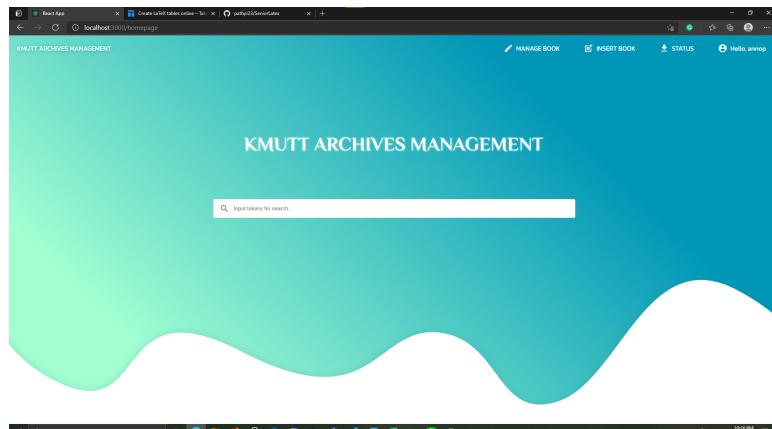
เกณฑ์การประเมิน	ผลลัพธ์		หมายเหตุ
	ผ่าน	ไม่ผ่าน	
1. สามารถเข้าสู่ระบบและออกจากระบบได้			
2. สามารถเพิ่มเอกสารเข้าสู่ระบบได้			
3. สามารถแก้ไขรายละเอียดเอกสารที่อยู่ในระบบได้			
4. สามารถตรวจสอบและแก้ไขคำที่เพิ่มเข้ามาในระบบในขั้นตอนเพิ่มเอกสารได้			ข้อสังเกตเกี่ยวกับการ Re-index ว่ามีผลต่อการค้นหาแบบ Real-time หรือไม่
5. สามารถลบเอกสารที่อยู่ในระบบได้			
6. สามารถค้นหาข้อมูลเอกสารภายในระบบได้			
7. สามารถเรียกดูเอกสารที่ต้องการได้			

4.3.2 การประเมินความพึงพอใจของบรรณาธิการต่อการออกแบบ UX/UI

ตารางที่ 4.14 ตารางประเมินความพึงพอใจจากการออกแบบ UX/UI

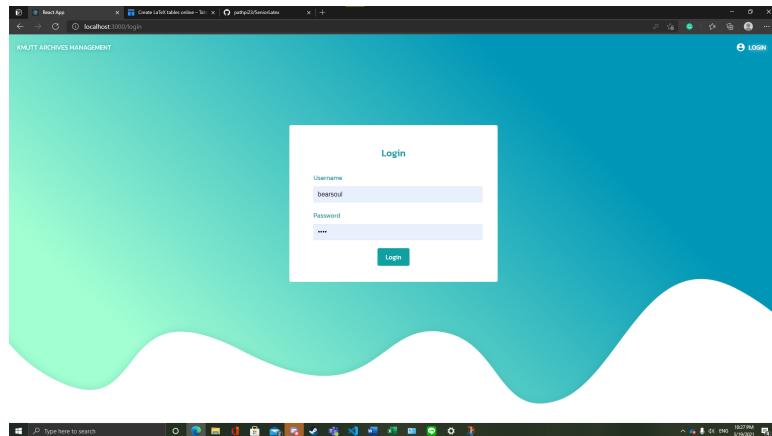
	4	3	2	1	คะแนนที่ได้
ความสมบูรณ์ของข้อมูล	ข้อมูลมีความสมบูรณ์ชัดเจนทำให้เข้าใจความหมายที่ต้องการจะสื่อได้เป็นอย่างดี	มีข้อมูลที่ชัดเจน และแม่นยำในบางครั้ง และสามารถแสดงความหมายที่ต้องการจะสื่อได้บ้าง	ข้อมูลมีความแม่นยำและชัดเจนบ้าง	มีข้อมูลที่ไม่ชัดเจน ไม่ครบ สื่อความหมายได้ไม่ดี	3
การออกแบบ	มีการออกแบบที่เน้นความสำคัญและจัดวางองค์ประกอบสี เสียง และการเคลื่อนไหว(animation) ได้อย่างเหมาะสม	มีการจัดหน้า และองค์ประกอบทำให้เห็นถึงความสำคัญของเนื้อหา มีการใช้การเคลื่อนไหว(animation) บ้าง	การวางหน้า และการจัดองค์ประกอบมีความไม่เหมาะสม มีการใช้การเคลื่อนไหว(animation) เข้ามาช่วยบ้าง	การวางหน้าและการจัดองค์ประกอบมีความไม่เหมาะสม และไม่มีการใช้การเคลื่อนไหว(animation) เข้ามาช่วยในการใช้งาน	4
การใช้งาน	ผู้ใช้สามารถใช้งานปุ่มหรือ้ายไปยังหน้าต่างๆได้อย่างง่ายดาย แต่มีลิงค์(Link) ที่พาไปผิดหน้าอย่างมากหนึ่งลิงค์(Link) หรือไม่มีเลย	ผู้ใช้สามารถใช้งานปุ่มหรือ้ายไปยังหน้าต่างๆได้อย่างง่ายดาย แต่มีลิงค์(Link) ที่พาไปผิดหน้าอย่างมากสองลิงค์(Link)	ผู้ใช้มีความสับสนในการใช้ปุ่ม หรือการ้ายไปยังหน้าต่างๆ บางครั้ง และมีลิงค์(Link) ที่พาไปผิดหน้าอย่างมากสามลิงค์(Link)	ผู้ใช้เกิดความสับสนในปุ่มหรือลิงค์(Link) ที่้ายไปหน้าต่างๆ	4
การใช้ภาษา	มีการใช้คำพิเศษหรือภาษาที่ไม่เหมาะสมอย่างมาก 1 จุด	มีการใช้คำพิเศษหรือภาษาที่ไม่เหมาะสมอย่างมาก 2 จุด	มีการใช้คำพิเศษหรือภาษาที่ไม่เหมาะสมอย่างมาก 3 จุด	มีการใช้คำพิเศษหรือภาษาที่ไม่เหมาะสมมากกว่า 4 จุด	4

4.3.3 หน้าหลัก



รูปที่ 4.5: ภาพแสดงหน้าเว็บหลัก

4.3.4 การเข้าสู่ระบบเว็บไซต์



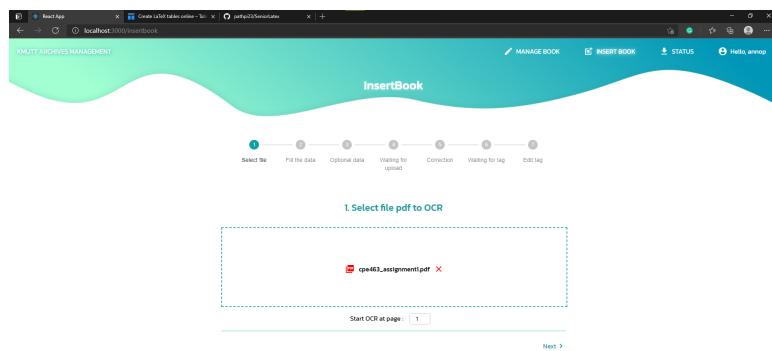
รูปที่ 4.6: ภาพแสดงหน้าเข้าสู่ระบบ

การเข้าสู่ระบบในเว็บไซต์เราได้ใช้ JSON Web Token (JWT) ในการดึงรหัสการเข้าใช้ระบบโดยที่เมื่อผู้ใช้งานเข้าสู่ระบบด้วยรหัสผู้ใช้งาน และรหัสผ่านที่ถูกต้องNode JS ก็จะคืน Token ที่ถูกหักล้างไปให้ทางเครื่องผู้ใช้งานเก็บใน local storage เพื่อที่จะเป็นการบ่งบอกว่าผู้ใช้งาน API ที่เหลือทั้งหมดไม่ว่าจะเป็นการค้นหาข้อมูล เพิ่มข้อมูลหนังสือ แก้ไขข้อมูลหนังสือ หรือลบข้อมูลหนังสือออกจากระบบ ถ้าผู้ใช้งานไม่ได้ส่ง Token มาด้วยหรือ Token นั้นมีการตัดแปลงแก้ไขระบบจะทำการลบ Token ภายใต้เครื่องทั้งหมดทำการออกจากระบบโดยทันที

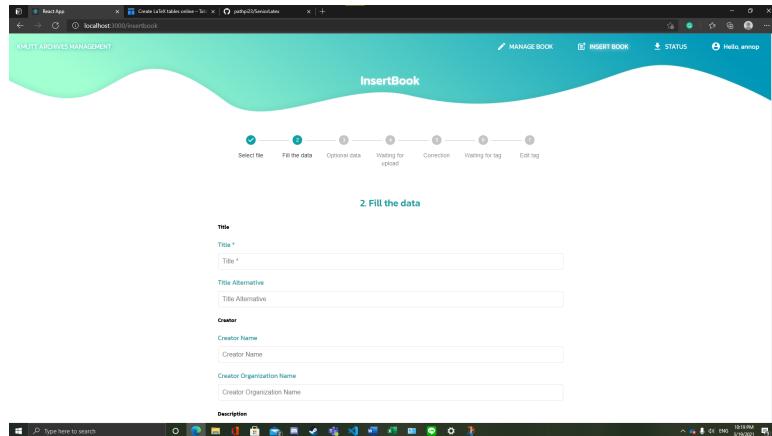
4.3.5 การเพิ่มหนังสือเข้าสู่ระบบฐานข้อมูล

เนื่องจากการเพิ่มหนังสือเข้าสู่ระบบมีขั้นตอนจำนวนมากและใช้เวลานานจึงแบ่งการอปะมวลผลเป็นส่วนของการเพิ่มข้อมูลของหนังสือ ส่วนของการแก้ไขและตรวจสอบคำกรองนำเข้าสู่ระบบ ส่วนของการตรวจสอบแก้ไขคำสำคัญ ซึ่งผู้ใช้งานไม่จำเป็นต้องรอภายในหน้าเพิ่มหนังสือสามารถไปทำงานพัฟฟ์ชั่นอื่นได้ตามปกติและเมื่อเสร็จกระบวนการเหล่านี้เสร็จจะสามารถกลับมาดำเนินการเพิ่มข้อมูลต่อได้โดยการกดที่หน้าแสดงสถานะ และกลับเข้าสู่กระบวนการเพิ่มข้อมูลหนังสือ

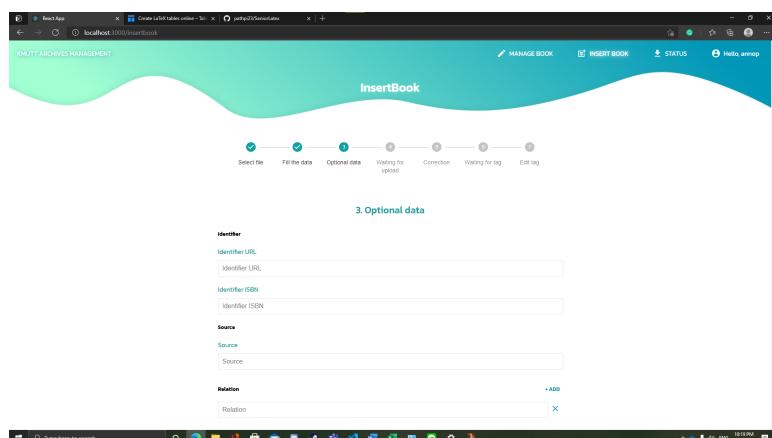
4.3.5.1 เพิ่มข้อมูลของหนังสือ



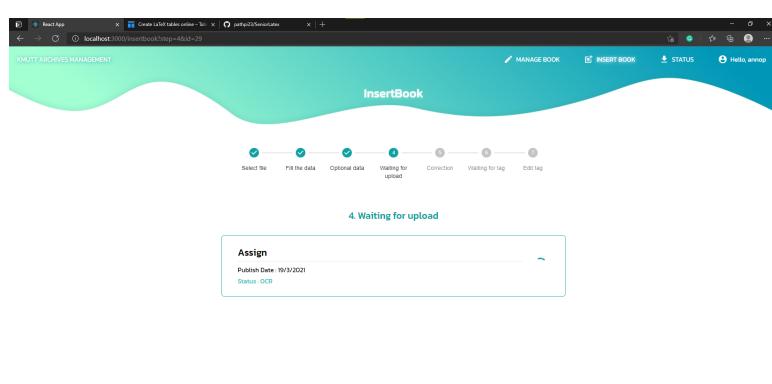
รูปที่ 4.7: ภาพแสดงขั้นตอนการเพิ่มหนังสือขั้นตอนการเพิ่มไฟล์



รูปที่ 4.8: ภาพแสดงขั้นตอนการเพิ่มหนังสือเข้าสู่ระบบขั้นกรอกข้อมูลขั้นที่ 1



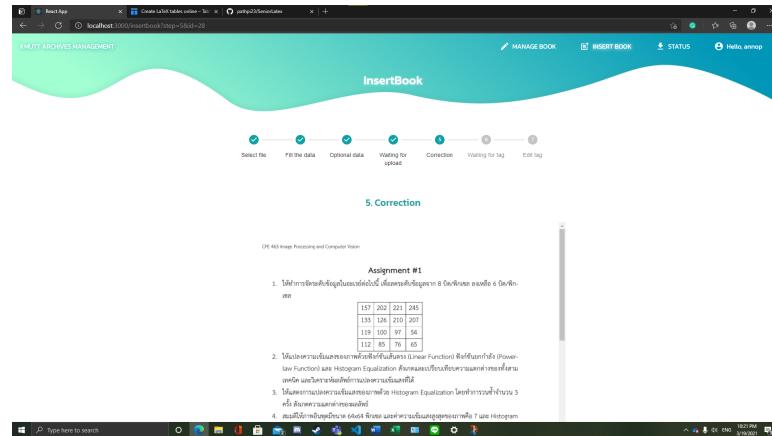
รูปที่ 4.9: ภาพแสดงขั้นตอนการเพิ่มหนังสือเข้าสู่ระบบขั้นกรอกข้อมูลขั้นที่ 2



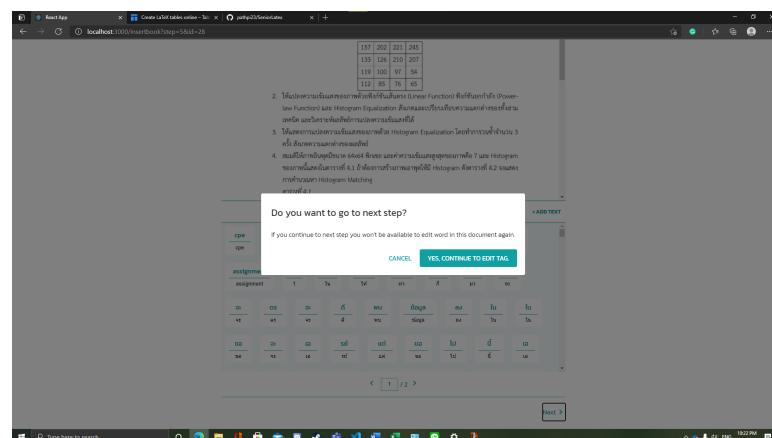
รูปที่ 4.10: ภาพแสดงขั้นตอนการเพิ่มหนังสือเข้าสู่ระบบขั้นการเตรียมข้อมูล

ในส่วนนี้จะเป็นการใช้ผู้ใช้งานทำการเลือกไฟล์ PDF และกรอกข้อมูลของหนังสือโดยที่เมื่อผู้ใช้งานบันทึกข้อมูลเรียบร้อยแล้วระบบก็จะทำการเพิ่มไฟล์ PDF เพื่อนำไปทำกระบวนการเปลี่ยน PDF เป็นรูปภาพและทำการ OCR และการเตรียมข้อมูลตัวอักษร เพื่อทำการแปลงข้อมูลออกมาให้ผู้ใช้งาน

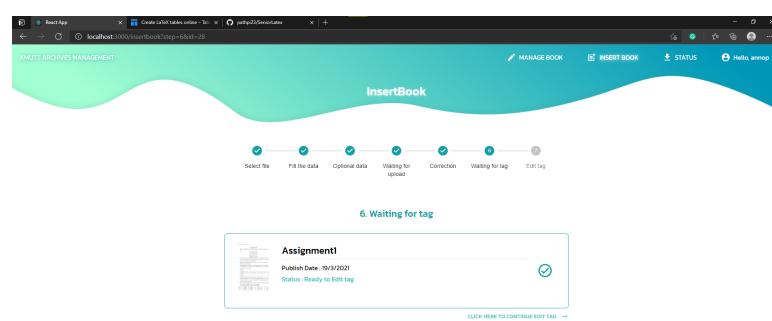
4.3.5.2 การแก้ไขและตรวจสอบคำก่อนนำเข้าสู่ระบบ



รูปที่ 4.11: ภาพแสดงขั้นตอนการเพิ่มหนังสือเข้าสู่ระบบขั้นการแก้ไขคำพิเศษ



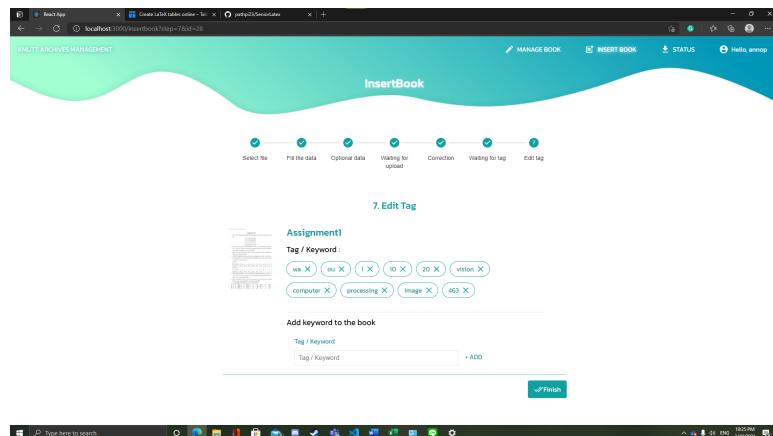
รูปที่ 4.12: ภาพแสดงหน้าต่างยืนยันการแก้คำ



รูปที่ 4.13: ภาพแสดงขั้นตอนการเพิ่มหนังสือเข้าสู่ระบบขั้นการสร้างคำสำคัญ

ในส่วนนี้จะเป็นผลลัพธ์ของการดำเนินการของ การเพิ่มข้อมูลหนังสือ จะมีคำขอต่อหน้าพร้อมรูปภาพประกอบเพื่อให้ผู้ใช้งานได้ตรวจสอบ คำเพิ่มและแก้ไขคำได้อย่างอิสระก่อนจะนำคำเหล่านี้เข้าสู่ระบบและในส่วนนี้ถ้ามีข้อผิดพลาดจะไม่สามารถมาแก้ไขคำในหนังสือ เล่นในระบบได้อีกโดยถ้ามีข้อผิดพลาดจะทำการเพิ่มคำเหล่านี้เข้าสู่ระบบและทำการคำนวนค่า TF-IDF ของคำเหล่านี้ก่อนจะสร้างคำสำคัญของหนังสือเล่มนี้ให้อัตโนมัติ

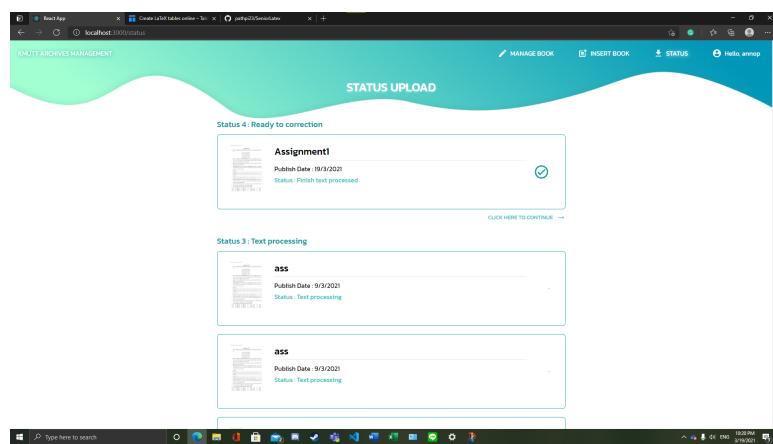
4.3.5.3 การตรวจสอบแก้ไขคำสำคัญ



รูปที่ 4.14: ภาพแสดงขั้นตอนการเพิ่มหนังสือเข้าสู่ระบบขั้นการแก้ไขคำสำคัญ

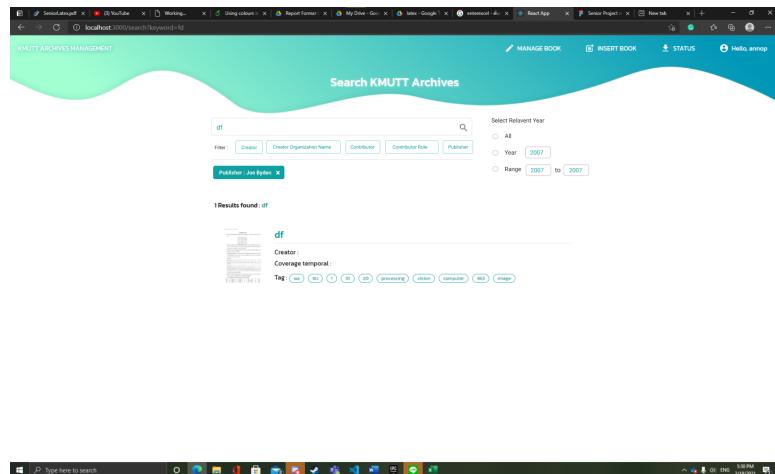
ในส่วนนี้จะเป็นผลลัพธ์ของการแก้ไขและตรวจสอบคำก่อนนำเข้าสู่ระบบโดยผู้ใช้งานได้คำสำคัญ ที่ทางระบบทำขึ้นอัตโนมัติเพื่อให้ผู้ใช้งานได้ตรวจสอบเพิ่มลดคำสำคัญ ก่อนจะยืนยันเพิ่มเข้าสู่ระบบ

4.3.6 การแสดงสถานะการเพิ่มหนังสือ



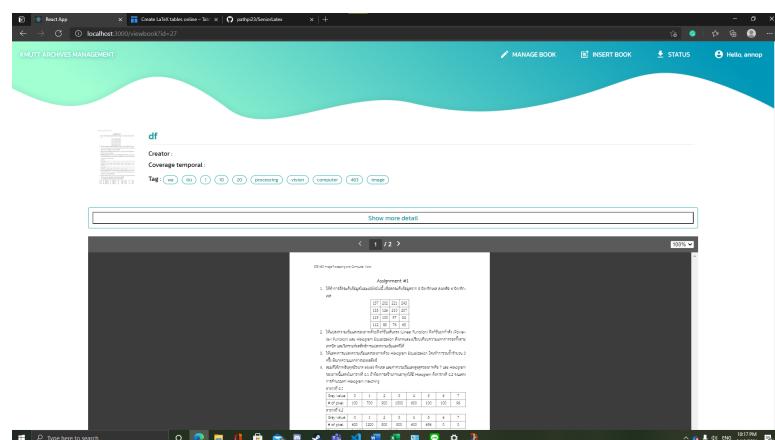
รูปที่ 4.15: ภาพแสดงสถานะของการเพิ่มข้อมูลเข้าสู่ระบบ

4.3.7 การแสดงการค้นหาหนังสือ



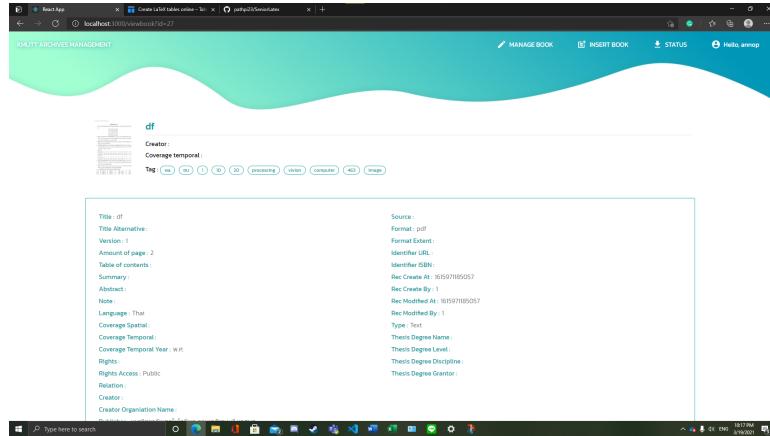
รูปที่ 4.16: ภาพแสดงหน้าการค้นหา

4.3.8 การแสดงข้อมูลหนังสือ



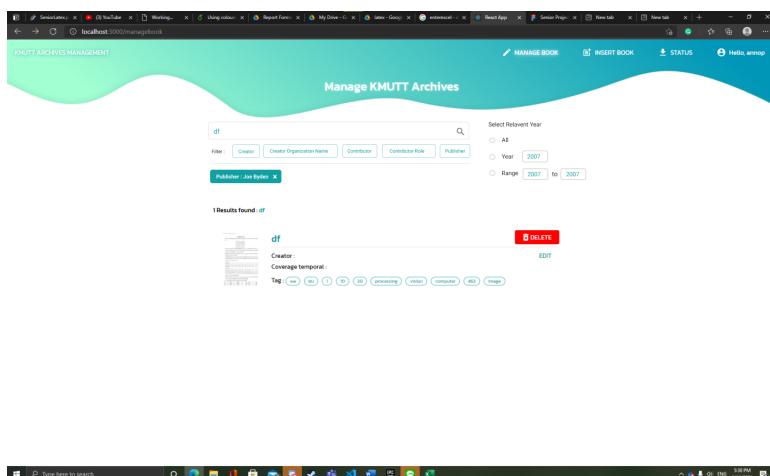
รูปที่ 4.17: ภาพแสดงหน้าแสดงหนังสือ

จะเป็นการแสดงข้อมูลของหนังสือที่อยู่ภายใต้ระบบที่ผู้ใช้งานกรอกเข้ามาในระบบพร้อมทั้งแสดง PDF ที่ถูกอัปโหลดขึ้นมา

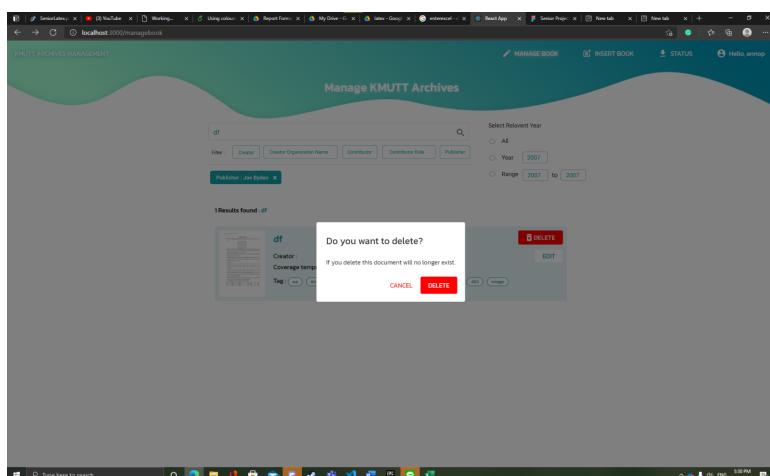


รูปที่ 4.18: ภาพแสดงข้อมูลของหนังสือ

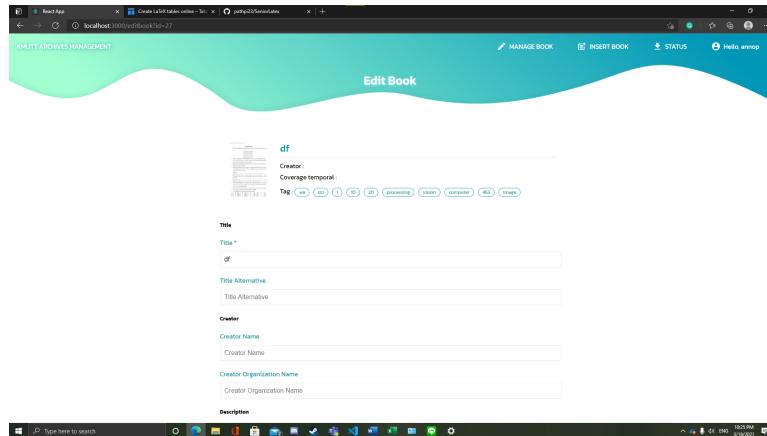
4.3.9 การแสดงการแก้ไขข้อมูลของหนังสือ



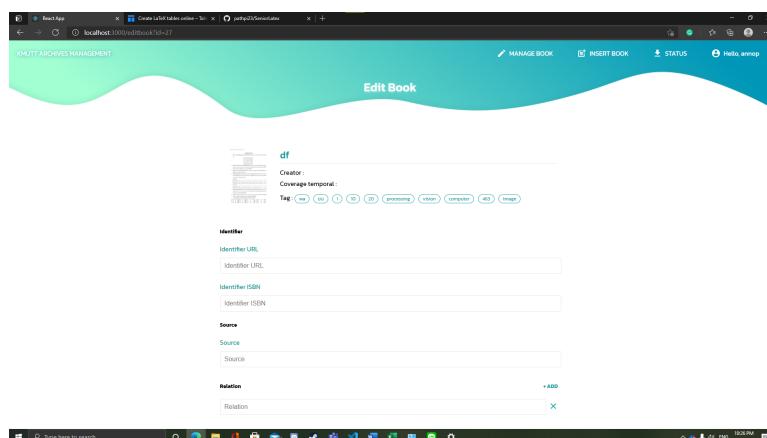
รูปที่ 4.19: ภาพแสดงหน้าการค้นหาในหน้าการจัดการหนังสือ



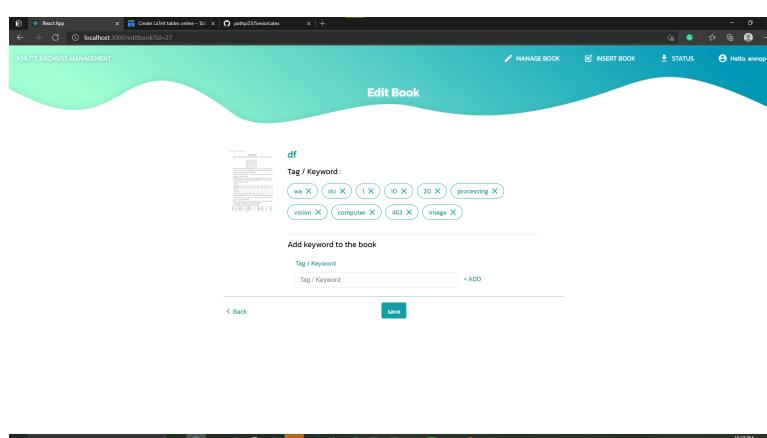
รูปที่ 4.20: ภาพแสดงหน้าการลบหนังสือ



รูปที่ 4.21: ภาพแสดงหน้าการแก้ไขข้อมูลขั้นที่ 1



รูปที่ 4.22: ภาพแสดงหน้าการแก้ไขข้อมูลขั้นที่ 2



รูปที่ 4.23: ภาพแสดงหน้าการแก้ไขคำสำคัญ

การแก้ไขข้อมูลจะแก้ได้ต่อเมื่อเพิ่มข้อมูลหนังสือเสร็จแล้วโดยที่จะสามารถแก้ไขข้อมูลในส่วนของข้อมูลหนังสือและคำสำคัญ ได้เหมือนกันกับการเพิ่มหนังสือโดยเมื่อแก้ไขเสร็จแล้วยังระบบจะทำการบันทึกข้อมูลใหม่ให้ทันที

บทที่ 5 สรุปผล

5.1 ผลการดำเนินงาน

ในภาคการเรียนที่ 1/2563 ทางคณะผู้จัดทำได้ทำการนำเสนอหัวข้อໂປຣເຈັກ ສຶກຂາຍແລະຮວບຮມໝ້ອມຸລືຕ່າງໆ ເກີບ Requirement ຈາກບຽນຮັກສູ່ ແລະໄດ້ອອກແບບເວັບໄປໜີ່ຕໍ່ User Interface, ໂຄງສ້າງຮຽນໝ້ອມຸລືແລະວິຊີ່ຕ່າງໆໃນການສ້າງເວັບໄປໜີ່ທີ່ຈະກຳທຳການແປ່ງຮູບປາພີໃຫ້ອູ້ໃນຮູບແບບດິຈິຫຼາດ ແລະກຳທຳການສຶກຂາຍ ເວັບໄປໜີ່ແລະອອກແບບວິຊີ່ການເຕີມໝ້ອມຸລືຮູບປາພີກອນທີ່ຈະນຳໄປການ OCR ສ້າງຮະບບເຕີມໝ້ອມຸລືຕ້ວອັກຊຣ ອຳນົດການຕັດຄຳ ແລະການແກ້ໄຂຄຳຜິດ ເພື່ອເຕີມໝ້ອມຸລືທີ່ໄດ້ຈາກການທຳ OCR ໃຫ້ອູ້ໃນຮູບແບບທີ່ເໝາະສົມສໍາຫຼວບຮະບບການຕັນຫາ ແລະສ້າງຮະບບຕັນຫາຄຳສັກຫຼຸດການຂອງ TF-IDF ເພື່ອໃຫ້ສ້າງຄຳສັກຫຼຸດຂອງຕ້ວອັກຊຣ

ณ ເວລາປັຈຸບັນໃນภาคເຮັດວຽກທີ່ 2/2563 ທາງຄະນະຜູ້ຈັດທຳໄດ້ວາງແຜນທີ່ຈະສ້າງເວັບໄປໜີ່ ຈັດທຳຮະບບການຕັນຫາ ແລະໂມໂຄດ Word2Vec ທຳການປະເມີນຮະບບການອອກແບບ User Interface ການເຕີມໝ້ອມຸລືຮູບປາພີ ຮະບບການແກ້ຄຳຜິດ ແລະຮະບບການຕັນຫາ

ตารางที่ 5.1 ตารางสรุปผลລັບຮັບການຕັດຄຳສັກຫຼຸດ

แผนการดำเนินการ	ยังไม่ดำเนินการ	กำลังดำเนินการ	เสร็จสิ้น
ສຶກຂາຍຕັ້ນຄວາຫາບໍ່ຢູ່ຫາຂອງໂຄຮກ			
เสนอຫัวข้อໂປຣເຈັກ			
ສຶກຂາຍແລະຫາໝ້ອມຸລືເກີບຕັ້ງກັບເທັນໂລຍືທີ່ໃໝ່ໃນໂປຣເຈັກ			
ປະເມີນຄວາມເປັນໄປໄດ້ແລະກຳທຳດ້ວຍເຫັນຫຼຸດຂອງໂປຣເຈັກ			
ຈັດເກີບ requirement ຈາກກສຸມຜູ້ໃໝ່ງານ			
ນຳເນັດອໂຄຮກງານຄັ້ງທີ່ 1			
ອອກແບບ UX/UI			
ການແປ່ງຮູບປາພີໃຫ້ອູ້ໃນຮູບແບບດິຈິຫຼາດ			
ນຳໝ້ອມຸລືທີ່ເກີບໄວ້ມາກຳທຳການຕັດແປ່ງຄຳກາຍາໄທແລະກຳທຳການສ້າງຄຳສັກຫຼຸດ ໂດຍໃໝ່ຫຼັກການຂອງ TF-IDF			
ຈັດທຳຮະບບການຕັນຫາ			
ຈັດທຳເວັບໄປໜີ່ແພລຕິໂອຣົມ			
ທດສອບຮະບບ			
ປັບປຸງແກ້ໄຂ			
ນຳເນັດອໂປຣເຈັກ			

5.2 สรุปผลการดำเนินงาน

การແປ່ງຮູບປາພີໃຫ້ອູ້ໃນຮູບແບບດິຈິຫຼາດ

จากผลลัพธ์ที่อยู่ในบทที่ 4 เรายังสามารถวัดผลลัพธ์การทำ Image processing ด้วยวิธีเช็คคำผ่านกระบวนการ OCR ว่ามีอะไรในการเตรียมรูปภาพแบบไหนให้ประสิทธิภาพของมาได้ดีกว่ากัน โดยที่เราเลือกใช้การเตรียมข้อมูลรูปภาพด้วยวิธีที่ 2 ซึ่งก็คือวิธีการลบพื้นหลัง เนื่องจากให้ผลลัพธ์โดยรวมที่มีประสิทธิภาพมากกว่าวิธีที่ 1 ที่ใช้วิธีการคัดเลือกข้อมูล การหมุน การลบรูปภาพ การลบเส้นและการรัดกลุ่ม แต่เนื่องจากการลบเส้นตรงทั้งแนวอนและแนวตั้งทำให้มีการลบตัวอักษรบางส่วนไปส่งผลให้เกิดการอ่านที่ยากขึ้น และการลบรูปภาพยังมีความผิดพลาดทำให้ลบเนื้อหาที่เป็นตัวอักษรบางส่วนไป ดังนั้น วิธีการลบพื้นหลังจึงมีประสิทธิภาพมากกว่าเนื่องจากไม่ได้ทำให้ข้อมูลเนื้อหาเกิดความเสียหายถึงแม้อาจจะทำให้คลบรูปภาพออกไปไม่หมดก็ตาม ขั้นตอนถัดมาจะนำรูปภาพที่ผ่านการทำ Image processing มาผ่านกระบวนการ OCR จาก Tesseract โดยเราได้ทำการ เปรียบเทียบข้อมูลที่นำมาให้โดยจะนำโมเดลภาษาไทย ในปี 2016 และ ในปี 2019 ของ Tesseract มาเปรียบเทียบประสิทธิภาพ และประเมินประสิทธิภาพการ แก้คำผิดภาษาไทยโดยใช้ library pythainlp และภาษาอังกฤษใช้ library pyspellchecker รวมถึงสร้างกลุ่มคำเฉพาะและ แก้ไขด้วยต้นเร่องด้วยวิธี minimum edit distance ว่าสามารถช่วยแก้คำผิดได้มากแค่ไหนและทำให้ผลลัพธ์หลังจากการแก้คำผิดนั้นผิดพลาดมากขึ้นหรือเปล่า

จากผลลัพธ์ที่อยู่ในบทที่ 4 จะเห็นว่าการแก้คำผิดด้วยวิธีต่างๆ สามารถแก้ไขคำผิดได้ประมาณ 2 % ทางผู้จัดทำคาดว่าเนื่องจากข้อมูลหนังสือที่เรานำมายังนั้น มีการใช้ภาษาที่แตกต่างกับหนังสือที่นำไปอีกทั้งยังมีการใช้คำเฉพาะอย่าง ชื่อคน สถานที่ วันเวลา งานวิจัย เป็นจำนวนมาก ซึ่งถึงแม้จะมีการแก้คำเฉพาะแต่ยังไม่ครอบคลุมข้อมูลทั้งหมด ทำให้การแก้ไขคำผิดมีประสิทธิภาพไม่สูงตามที่คาดหวังไว้ จะเห็นได้จากข้อมูลที่โมเดลในปี 2016 ที่ได้ผลลัพธ์ความถูกต้องก่อนแก้ไขคำผิดอยู่ที่ 74.75 % เพิ่มขึ้นเป็น 76.61 % หลังจากผ่านวิธีในการแก้ไขคำผิด และทำให้มีคำเกินลดลงเมื่อเทียบกับก่อนที่จะแก้คำผิด และจากการเปรียบเทียบโมเดลในปี 2016 และชุดโมเดลในปี 2019 พบว่าโมเดลชุด 2019 นั้นมีเปอร์เซ็นต์ความถูกต้องอยู่ที่ 77.41 % ซึ่งสูงกว่าชุดโมเดลปี 2016 แต่ว่า จำนวนคำเกินที่ได้จาก โมเดลปี 2019 นั้นมีมากกว่าปี 2016 กว่าเท่าตัว (ปี 2019 มีคำเกิน 173 คำ ปี 2016 มี 83 คำ) ทำให้เราเลือกใช้โมเดลในปี 2016 ในการทำ OCR ในโปรแกรมนี้

การค้นหาเอกสารภายในระบบ

จากผลลัพธ์ในบทที่ 4 จะเห็นได้ว่าการใช้งานเครื่อง TF-IDF ร่วมกับ Word2Vec กับข้อมูลภายในระบบได้ผลลัพธ์ที่ไม่ดีนักโดยที่ มีค่าความแม่นยำ (Accuracy) อยู่เพียงแค่ 61.46 % และค่าความครอบคลุม (Recall) อยู่ที่ 88.24 % ซึ่งเทียบกับการใช้ TF-IDF เพียงอย่างเดียว ได้ผลลัพธ์ที่ดีกว่าโดยที่มีค่าความแม่นยำ 75% ค่าความครอบคลุมอยู่ที่ 88.24 % คาดว่าเนื่องจากข้อมูลที่นำมาใช้งานภายในไม่มีผล Word2Vec มีจำนวนที่น้อยจนเกินไปประกอบกับจำนวนลักษณะการเขียนเนื้อหาภายในหนังสือมีการใช้คำเฉพาะทำให้หาความสัมพันธ์ของคำได้ยากยิ่งขึ้น รวมถึงการค้นหาคำความสัมพันธ์ของคำเฉพาะอย่างเช่นบุคคล นั้นยังทำได้ไม่ดีนัก จากการทดลองพบว่าเมื่อนำคำค้นที่เป็นชื่อเฉพาะออกที่ใหม่ค่าความแม่นยำเพิ่มขึ้นเป็น 79.17 % โดยที่ค่าความครอบคลุมเพิ่มเป็น 87.5 %

หลังจากได้ผลลัพธ์การเปรียบเทียบประสิทธิภาพเวลาในการค้นหานอกบทที่ 4 โดยการนำบุคคลธรรมชาติที่ใช้ระบบค้นหาของเราเปรียบเทียบกับเจ้าหน้าที่ บรรณารักษ์ที่เปิดค้นหาข้อมูลด้วยวิธีการปกติพบว่า ระยะเวลาในการเลือกหนังสือที่มีความเกี่ยวข้อง บุคคลธรรมชาติ ทั้ง 2 คน ที่ใช้ระบบสามารถค้นหาหนังสือที่มีคำที่เกี่ยวข้องเจือและถูกต้องได้ภายในเวลาเฉลี่ย 1 นาที ซึ่งเมื่อเปรียบเทียบกับเจ้าหน้าที่บรรณารักษ์ที่เปิดค้นหาข้อมูลด้วยวิธีปกติพบว่าจะเจอหนังสือที่เกี่ยวข้องก็ต้องเบิดอ่านข้อมูลข้างในก่อนทำให้ใช้เวลาไปถึง 11 นาทีถึงจะสามารถเลือกหนังสือที่ตรงกับคำที่ต้องการค้นหา แต่เนื่องจากระบบค้นหาภายในโปรแกรมยังไม่สามารถระบุหน้าที่มีคำค้นหาได้จึงทำให้ถ้าเปรียบเทียบให้บุคคลธรรมชาติเปิดหน้าที่มีคำค้นหาคนที่ 1 จะใช้เวลา 10 นาที และคนที่สองใช้เวลา 12 นาที และในส่วนของเจ้าหน้าที่บรรณารักษ์ที่ใช้วิธีการปกติจึงใช้เวลาไป 11 นาที เท่าเดิมเนื่องจากต้องเบิดเจอคำค้นหาก่อนถึงจะสามารถระบุหนังสือที่เกี่ยวข้องได้ และจากผลลัพธ์พบว่าการที่เจ้าหน้าที่บรรณารักษ์ใช้เวลาส่วนใหญ่ในการค้นหาหนังสือที่เกี่ยวข้องเนื่องจากการที่ส่วนใหญ่หนังสือแต่ไม่มีคำสำคัญที่ต้องการทำให้ต้องเสียเวลาในการค้นหาเพิ่มมากยิ่งขึ้น ซึ่งถ้าเจ้าหน้าที่บรรณารักษ์ใช้ระบบค้นหาภายในโปรแกรมนี้จะช่วยลดเวลาในการค้นหาไปถึง 5 นาที สำหรับการส่วนใหญ่หนังสือที่ไม่เกี่ยวข้องมาแล้วในส่วน ของผลลัพธ์ที่ทางเจ้าหน้าที่บรรณารักษ์คาดหวังสำหรับความแม่นยำอยู่ที่ 75 % ซึ่งระบบค้นหาของเราทำได้อย่างพอดีที่คาดหวังไว้

การออกแบบและการใช้งานเว็บไซต์แพลตฟอร์ม

จากผลลัพธ์การประเมินจากเจ้าหน้าที่บรรณารักษ์ผู้ใช้งานจริงได้ประเมินผลลัพธ์ค่อนข้างอ่อนโยนเป็นที่น่าพึงพอใจ แต่ยังมีบางส่วนที่ถูกใช้เพียงแค่โครงสร้างรูปภาพสื่อความหมายเจ้าหน้าที่บรรณารักษ์อาจจะยังไม่เข้าใจนักจึงต้องการเพิ่มเติมในส่วนของคำอธิบายเพิ่มประกอบ

ไปด้วยซึ่งทางเราได้ปรับปรุงเพิ่มคำอธิบายและสิ่งที่ผู้ใช้ไม่เข้าใจเรียบร้อยแล้ว และการออกแบบการทำงานของเว็บไซต์แพลฟอร์มโดยใช้ NodeJS React Django สามารถส่งข้อมูลส่งไปมาและรับส่งข้อมูลผ่าน Restful API และฐานข้อมูลทำได้อย่างไม่มีปัญหาใดๆ

5.3 ปัญหาที่พบและการแก้ไข

5.3.1 ปัญหาหน้าสีอ่านยาก

เนื่องจากหนังสือแต่ละเล่มมีลักษณะที่แตกต่างกันในเรื่องของสีของกระดาษและตัวอักษร ลักษณะการสแกนรูปภาพจึงทำให้การนำประโคนช้อความที่ถูกตัดออกมาทำ OCR แล้วเกิดความผิดพลาดเยอะ

การแก้ไข

ทำการแก้ไขกระบวนการกรารเตรียมข้อมูลรูปภาพ จากการพยายามข้ามหน้าสีเป็นพัฒนาการลบพื้นหลังในรูปแบบใหม่

5.3.2 ปัญหาการหมุนไม่ตรง

เนื่องจากภาษาไทยมีสรวรรณยุกต์ด้านบนต่อ กันสูงสุดต่อ กันถึง 2 ชั้นนอกจากนั้นยังมีสรวรรณยุกต์ด้านล่างทำให้บางที่ไม่สามารถแยกข้อความแต่ละบรรทัดออกมายได้อย่างสมบูรณ์จึงทำให้มีการหมุนที่ผิดพลาดเกิดขึ้น

การแก้ไข

ทำการแก้ไขกระบวนการกรารเตรียมข้อมูลรูปภาพ ปรับเปลี่ยนวิธีการหมุนเป็นการหา Arctan ที่จุดขอบด้านบนที่ทำให้การหมุนมีข้อผิดพลาดน้อยลง

5.3.3 ปัญหาการแก้ไขคำผิด

เนื่องจากการแก้ไขคำผิดยังไม่สามารถแก้คำผิดรูปแบบคำเฉพาะได้อย่าง เช่นชื่อคน หรือชื่อสถานที่ หรืออาจจะคิดว่าคำเฉพาะนั้นผิดพลาด และทำการแก้ไขให้อัตโนมัติทำให้คำที่ได้รับออกมาก็ข้อผิดพลาดขึ้น

การแก้ไข

ให้ผู้ใช้งานได้ตรวจสอบและแก้ไขได้เองก่อนเพิ่มข้อมูลเข้าสู่ระบบและเพิ่มข้อมูลคำเฉพาะบางส่วนลงไป

5.3.4 ปัญหาระยะเวลาในการเพิ่มข้อมูลหนังสือ

เนื่องจากการเพิ่มข้อมูลมีขั้นตอนจำนวนมากและใช้เวลานานผู้จัดทำจึงออกแบบเป็นระบบ Thread เพื่อที่จะให้เชิฟเวอร์ตอบกลับไปยังผู้ใช้งานและเปิด Thread ในการทำงานไม่ว่าจะเป็น OCR หรือการทำกรเตรียมข้อมูลตัวอักษร เพื่อเพิ่มความเร็วในการทำงานแต่ก็ไม่สามารถลดเวลาการทำงานลงได้

การแก้ไข

ทำการแก้ไขโดยการ Spawn Process แทนที่จะนำเนื้อหาจาก python นั้นเวลาเปิด Thread ยังคงใช้ core เดียวในการประมวลผลซึ่งต้องเปลี่ยนมาเป็นการ Spawn Process ที่แยกการใช้ core ของหน่วยประมวลผลทำให้สามารถลดเวลาในการเพิ่มข้อมูลหนังสือได้

5.3.5 ปัญหาของการหาคำเหมือนของโมเดล Word2Vec

จากการทดลองสร้างโมเดลโดยการนำหนังสือ กดเวลาทิتا และ รายงานประจำปีมาทำโมเดลพบว่าคำในประโยคส่วนใหญ่ของเหล่าหนังสือนี้ มีวิธีการเขียนที่แตกต่างจากหนังสือที่นำไปทั้ง ภาษาและคำที่นำมาใช้เขียนภายในหนังสือ ทำให้ความเมื่อนานมาทำโมเดล Word2Vec จึงทำให้ค่าความสัมพันธ์ของคำนั้นไม่ดี นอกจากนั้นยังมีเรื่องของจำนวนข้อมูลที่นำมาใช้ในแต่ละโมเดลซึ่ง แบ่งเป็นโมเดลที่มีข้อมูล 4 เล่ม และโมเดลที่มีข้อมูล 6 เล่ม ซึ่งจำนวนข้อมูลที่นำมาใช้งานยังไม่น่มากพอที่จะทำให้หาความสัมพันธ์ที่ดี นอกจากนั้นปัญหาระบบคำที่ผิดภายในข้อมูลก็ยังส่งผลทำให้การหาคำความสัมพันธ์ยังแยกด้วยถึงแม้ว่าจะมีแก้ไขคำผิดในข้อมูลหนังสือ 6 เล่มก็ตามแต่ด้วยจำนวนที่น้อย จึงทำให้ประสิทธิภาพการทำงานแย่

การแก้ไข หนังสือที่มีลักษณะการเขียนหนังสือแบบนี้เพิ่มเติมเข้ามาใช้ในการสร้างโมเดล นอกจากนั้นการแก้ไขคำผิดให้ถูกต้องทั้งหมด และการตัดคำภายในประโยคให้มีความถูกต้องมากยิ่งขึ้น

5.4 ข้อจำกัดและข้อเสนอแนะ

1. การแก้ไขคำเฉพาะยังไม่สามารถทำได้ถึงแม่ว่ามีการเพิ่มคำศัพท์เฉพาะลงไปเก็ตตามแต่ก็ยังต้องให้มุขย์เป็นผู้ตรวจสอบอีกรอบเพื่อความถูกต้อง
2. เนื่องจากลักษณะหนังสือแต่ละเล่มแตกต่างกันทำให้การเตรียมข้อมูลรูปภาพที่อาจจะไม่ได้ส่งผลลัพธ์ที่ดีที่สุดให้กับหนังสือทุกประเภท ทำให้ประสิทธิภาพในหนังสือบางเล่มน้อยกว่าหรือมากกว่าอีกเล่มได้
3. ลักษณะแสงของการสแกนหนังสือเนื่องจากไฟล์ที่ได้รับมาไม่มีการควบคุมในการสแกนหนังสือเข้ามาจึงทำให้ไฟล์มีความสว่างที่ไม่เท่ากันขนาดและการจัดวางที่ไม่เหมือนกันทำให้ประสิทธิภาพของแต่ละเล่มอาจจะไม่เท่าเดิม
4. การพัฒนาระบบการค้นหาด้วยการใช้ความสัมพันธ์ของลำดับรูปประโยค เนื่องจากรูปประโยคจะประกอบด้วยคำศัพท์มารวมกัน ซึ่งลำดับการเรียงคำทำให้เกิดความหมาย อย่างเช่น “วันนี้ กิน ข้าว อะไร ดี” กับ “ดี วันนี้ กิน อะไร ข้าว” จะเห็นได้ว่าประโยค อย่างหลังไม่สามารถตีความหมายได้อย่างชัดเจน หลังจากการวิเคราะห์ลักษณะกับรูปแบบระบบการค้นหาที่ยังขาดส่วนน้อย ทำให้การค้นหาของผู้ใช้งานสามารถค้นหาคีย์เวิร์ดได้ แต่ไม่สามารถค้นหาไปถึงความหมายที่จะสื่อถึงได้ในบางกรณีอย่าง “กรรมการ แข่งขัน” กับ “แข่งขัน กรรมการ จัดการ” โดยที่คำเหมือนกันแต่การสื่อความหมายต่างกัน ดังนั้นเราจึงได้นำทฤษฎีที่มีชื่อว่า Bi-gram, N-gram มาศึกษาโดยทฤษฎีนี้จะมาช่วยในการเข้มความสัมพันธ์ของลำดับรูปประโยคได้

หนังสืออ้างอิง

1. Ritambhara, "Minimum edit distance of two strings," <https://www.ritambhara.in/minimum-edit-distance-of-two-strings/>, [Online; accessed 12-October-2020].
2. Saixiii, 2017, "RESTful គីអេលូ REST គឺ ការសៀវភៅរាល់បែនិយ័យខ្មែរដោយ webservice," <https://saixiii.com/what-is-restful/#:~:text=Representational%20state%20transfer%20%E0%B8%AB%E0%B8%A3%E0%B8%B7%E0%B8%AD%20REST,XML%2C%20HTML%2C%20JSON%20%E0%B9%82%E0%B8%94%E0%B8%A2%20response/>.
3. ICHI.PRO, "ទុកប្រមាណផលភាពាហុទ្ធឌាតិ," <https://ichi.pro/th/chud-pramwl-phl-phas-a-thrrmchati-99851277240821>, [Online;accessed 17 May 2021].
4. Doxygen, 2020, "OpenCV," https://docs.opencv.org/3.4/d4/d73/tutorial_py_contours_begin.html, [Online; accessed 12-October-2020].
5. NECTEC, "AI For Thai," <https://aiforthai.in.th/index.php#home>, [Online; accessed 22-November-2020].
6. Google, 2020, "Tesseract OCR," <https://opensource.google/projects/tesseract>, [Online; accessed 22-November-2020].
7. Peter Norvig, 2016, "Image Morphology," <https://norvig.com/spell-correct.html>, [Online;accessed 17 May 2021].
8. Project Gutenberg, " Project Gutenberg," <https://www.gutenberg.org/>.
9. Wiktionary, 2021, "Wiktionary:Frequency lists," https://en.wiktionary.org/wiki/Wiktionary:Frequency_lists, [Online;accessed 17 May 2021].
10. Adam Kilgarriff, 1995, "BNC database and word frequency lists," <http://www.kilgarriff.co.uk/bnc-readme.html>, [Online;accessed 17 May 2021].
11. Pythainlp, "PyThaiNLP," https://pythainlp.github.io/docs/2.0/notes/getting_started.html.
12. PyPI, "pyspellchecker 0.6.2," <https://pypi.org/project/pyspellchecker/>.
13. Xin Rong, 2014, "word2vec Parameter Learning Explained," **CoRR**, vol. abs/1411.2738, 2014.
14. Y. Goldberg and O. Levy, 2014, "word2vec Explained: Deriving Mikolov et al.'s," <https://arxiv.org/pdf/1402.3722v1.pdf>.
15. Fasttext, 2018, "English word vectors," <https://fasttext.cc/docs/en/english-vectors.html>.
16. Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer, 2018, "Deep contextualized word representations," **CoRR**, vol. abs/1802.05365, 2018.
17. Ria Kulshrestha, 2019, "NLP 101: Word2Vec — Skip-gram and CBOW," [https://towardsdatascience.com/nlp-101-word2vec-skip-gram-and-cbow-93512ee24314#:~:text=Continuous%20Bag%20of%20Words%20Model%20\(CBOW\)%20and%20Skip%2Dgram&text=In%20the%20CBOW%20model%2C%20the,used%20to%20predict%20the%20context%20.](https://towardsdatascience.com/nlp-101-word2vec-skip-gram-and-cbow-93512ee24314#:~:text=Continuous%20Bag%20of%20Words%20Model%20(CBOW)%20and%20Skip%2Dgram&text=In%20the%20CBOW%20model%2C%20the,used%20to%20predict%20the%20context%20.), [Online; accessed 17 May 2021].
18. Keiron O'Shea and Ryan Nash, 2015, "An Introduction to Convolutional Neural Networks," **CoRR**, vol. abs/1511.08458, 2015.
19. techterms, 2018, "MVC," <https://techterms.com/definition/mvc>, [Online; accessed 10-October-2020].
20. Matt Zucker, 2016, "Page dewarping," <https://mzucker.github.io/2016/08/15/page-dewarping.html>.

ກາຄົນວກ A

DATABASE DICTIONARY

ตารางแสดงรายละเอียดข้อมูลในระบบฐานข้อมูล

ตารางที่ A.2 ตารางอธิบายความหมายตาราง term_word

term_word		
ชื่อคอลัมน์	ความหมาย	ประเภท
term_word_id	id สำหรับบ่งบอกคำศัพท์	INT (10) PK Auto_Increment
term	คำศัพท์	VARCHAR (191)
frequency	จำนวนความถี่ของหนังสือที่มีคำศัพทนี้อยู่	INT (191)
score_idf	ค่าคะแนน idf ของคำศัพทนี้	FLOAT (255,4)
rec_create_at	วันเวลาของการเพิ่มคำศัพทนี้เข้าสู่ระบบ	DATETIME (6) current_timestamp
rec_modified_at	วันเวลาที่อัปเดตข้อมูลของคำศัพท์	DATETIME (6) current_timestamp

ตารางที่ A.3 ตารางอธิบายความหมายตาราง user

user		
ชื่อคอลัมน์	ความหมาย	ประเภท
user_id	id สำหรับบ่งบอกผู้ใช้งาน	INT (10) PK Auto_Increment
name	ชื่อของผู้ใช้งาน	VARCHAR (50)
surname	นามสกุลของผู้ใช้งาน	VARCHAR (191)
role	ตำแหน่งของผู้ใช้งาน	VARCHAR (191)
username	ชื่อผู้ใช้งานสำหรับทำการ login	VARCHAR (191)
password	รหัสผ่านผู้ใช้งานสำหรับทำการ login	VARCHAR (191)
create_at	วันเวลาของผู้ใช้งานของการเพิ่มเข้าสู่ระบบ	DATETIME (6) current_timestamp
active	สถานะการลงทะเบียนผู้ใช้งาน	INT (11) Default 1

ตารางที่ A.4 ตารางอธิบายความหมายตาราง score

score		
ชื่อคอลัมน์	ความหมาย	ประเภท
score_id	id สำหรับบ่งบอกคะแนนของคำศัพท์	INT (10) PK Auto_Increment
score_tf	คะแนน tf ของคำศัพท์	FLOAT (255,4)
score_tf_idf	คะแนน tf-idf ของคำศัพท์	FLOAT (255,4)
index_term_word_id	id สำหรับบ่งบอกคำศัพท์	INT (10)
index_document_id	id สำหรับบ่งบอกหนังสือ	INT (10)
generate_by	คะแนนถูกคำนวณโดยใคร	VARCHAR (191) Default 'default'
rec_status	สถานะการใช้คะแนนนี้	INT (191) Default 1

ตารางที่ A.5 ตารางอธิบายความหมายตาราง pre_term_in_page

ชื่อคอลัมน์	ความหมาย	ประเภท
pre_term_in_page_id	id สำหรับบ่งบอกคำศัพท์ซึ่งคราวที่รอด้วยผู้ใช้งานตรวจสอบ	INT (10) PK Auto_Increment
pre_term	คำศัพท์ซึ่งคราวที่รอด้วยผู้ใช้ตรวจสอบ	VARCHAR (191)
index_page_in_document_id	id สำหรับบ่งบอกที่อยู่ของคำศัพท์ซึ่งคราวที่รอด้วยผู้ใช้งานตรวจสอบ	INT (10) FK

ตารางที่ A.6 ตารางอธิบายความหมายตาราง page_in_document

page_in_document		
ชื่อคอลัมน์	ความหมาย	ประเภท
page_in_document_id	id สำหรับบ่งบอกที่อยู่ของคำศัพท์ซึ่งคราวที่รอด้วยผู้ใช้งานตรวจสอบ	INT (10) PK Auto_Increment
page_index	หน้าของหนังสือ	INT (191)
name	ชื่อ File ของข้อมูล	VARCHAR (191)
rec_status_confirm	สถานะการยืนยันโดยผู้ใช้งาน	INT (2) PK Default 2
index_document_id	id สำหรับบ่งบอกหนังสือ	INT (10) FK

ตารางที่ A.7 ตารางอธิบายความหมายตาราง nodejs_log

nodejs_log		
ชื่อคอลัมน์	ความหมาย	ประเภท
nodejs_log_id	id สำหรับการจัดเก็บประวัติการทำงานผ่าน nodejs	INT (10) PK Auto_Increment
status_code	เก็บสถานะ HTTP หลังจากที่ส่งไปแล้วว่าได้สถานะใด	INT (191)
header_date	เก็บข้อมูล header ของ HTTP ที่ส่งไป	VARCHAR (191)
server	ชื่อรูปแบบของเซิฟเวอร์ที่ส่งไป	VARCHAR (191)
url	ตำแหน่งโดเมนหรือ IP ที่ส่งไป	INT (10) FK
content_type	รูปแบบเนื้อหาที่ส่งไป	VARCHAR (191)
rec_status	สถานะที่บอกร่วมกับการส่งเกิดข้อผิดพลาดระหว่างทาง	INT (191)
rec_create_date	วันเวลาที่ทำการส่ง ณ ตอนนั้น	DATETIME (6) current_timestamp

ตารางที่ A.8 ตารางอธิบายความหมายตาราง knex_migrations_lock

knex_migrations_lock		
ชื่อคอลัมน์	ความหมาย	ประเภท
index	บ่งบอกลำดับของไฟล์ migration ของ knex	INT (10) PK Auto_Increment
is_locked	สถานะของไฟล์ migration	INT (11)

ตารางที่ A.9 ตารางอธิบายความหมายตาราง knex_migrations

knex_migrations		
ชื่อคอลัมน์	ความหมาย	ประเภท
id	บ่งบอกลำดับการทำงานของไฟล์ migration ของ knex	INT (10) PK Auto_Increment
name	ชื่อไฟล์ migration ที่ถูกทำงานเรียบร้อย	VARCHAR (255)
batch	ลำดับที่	INT (11)
migration_time	เวลาที่ถูกสั่งให้ทำงาน	TIMESTAMP current_timestamp

ตารางที่ A.10 ตารางอธิบายความหมายตาราง indexing_publisher_document

indexing_publisher_document		
ชื่อคอลัมน์	ความหมาย	ประเภท
indexing_publisher_id	id สำหรับบ่งบอกสำนักพิมพ์	INT (10) PK Auto_Increment
publisher	ชื่อสำนักพิมพ์	VARCHAR (191)
frequency	จำนวนของสำนักพิมพ์ที่ถูกอ้างอิง	INT (191)

ตารางที่ A.11 ตารางอธิบายความหมายตาราง indexing_publisher_email_document

indexing_publisher_email_document		
ชื่อคอลัมน์	ความหมาย	ประเภท
indexing_publisher_email_id	id สำหรับบ่งบอกสำนักพิมพ์	INT (10) PK Auto_Increment
publisher_email	e-mail ของสำนักพิมพ์	VARCHAR (191)
frequency	จำนวนของสำนักพิมพ์ที่ถูกอ้างอิง	INT (191)

ตารางที่ A.12 ตารางอธิบายความหมายตาราง indexing_issued_date_document

indexing_issued_date_document		
ชื่อคอลัมน์	ความหมาย	ประเภท
indexing_issued_date_id	id สำหรับบ่งบอกปีที่เขียน	INT (10) PK Auto_Increment
issued_date	วันเวลาของปีที่เขียนหนังสือ	DATE
frequency	จำนวนของวันเวลาของปีที่เขียนที่ถูกอ้างอิง	INT (191)

ตารางที่ A.13 ตารางอธิบายความหมายตาราง indexing_creator_orgname_document

indexing_creator_orgname_document		
ชื่อคอลัมน์	ความหมาย	ประเภท
indexing_creator_orgname_id	id สำหรับบ่งบอกชื่อหน่วยงานรับผิดชอบสังกัด	INT (10) PK Auto_Increment
creator_orgname	ชื่อหน่วยงานรับผิดชอบสังกัด	VARCHAR (191)
frequency	จำนวนของหน่วยงานรับผิดชอบสังกัดที่ถูกอ้างอิง	INT (191)

ตารางที่ A.14 ตารางอธิบายความหมายตาราง indexing_creator_document

indexing_creator_document		
ชื่อคอลัมน์	ความหมาย	ประเภท
indexing_creator_id	id สำหรับบ่งบอกชื่อผู้เขียนหนังสือ	INT (10) PK Auto_Increment
creator	ชื่อของผู้เขียนหนังสือ	VARCHAR (191)
frequency	จำนวนของผู้เขียนหนังสือที่ถูกอ้างอิง	INT (191)

ตารางที่ A.15 ตารางอธิบายความหมายตาราง dc_contributors

dc_contributors		
ชื่อคอลัมน์	ความหมาย	ประเภท
dc_contributors_id	id สำหรับบ่งบอกข้อมูลความสัมพันธ์ของชื่อหน่วยข้อมูลผู้ร่วมงาน กับหนังสือ	INT (10) PK Auto_Increment
index_contributor_id	id สำหรับบ่งบอกชื่อผู้เขียนหนังสือ	INT(10) FK
index_document_id	id สำหรับบ่งบอกชื่อหน่วยข้อมูลผู้ร่วมงาน	INT(10) FK

ตารางที่ A.16 ตารางอธิบายความหมายตาราง indexing_contributor_document

indexing_contributor_document		
ชื่อคอลัมน์	ความหมาย	ประเภท
indexing_contributor_id	id สำหรับบ่งบอกชื่อหน่วยข้อมูลผู้ร่วมงาน	INT (10) PK Auto_Increment
contributor	ชื่อหน่วยข้อมูลผู้ร่วมงาน	VARCHAR (191)
frequency	จำนวนของหน่วยข้อมูลผู้ร่วมงานที่ถูกอ้างอิง	INT (191)

ตารางที่ A.17 ตารางอธิบายความหมายตาราง indexing_contributor_role_document

indexing_contributor_role_document		
ชื่อคอลัมน์	ความหมาย	ประเภท
indexing_contributor_role_id	id สำหรับบ่งบอกตำแหน่งของผู้ร่วมงาน	INT(10) Auto_Increment
contributor_role	ชื่อตำแหน่งของผู้ร่วมงาน	VARCHAR (191)
index_contributor	id สำหรับบ่งบอกชื่อหน่วยข้อมูลผู้ร่วมงาน	INT (191)

ตารางที่ A.18 ตารางอธิบายความหมายตาราง dc_type

dc_type		
ชื่อคอลัมน์	ความหมาย	ประเภท
DC_type_id	id สำหรับบ่งบอกประเภทของหนังสือ	INT (10) PK Auto_Increment
DC_type	ประเภทของหนังสือ	VARCHAR (191)
index_document_id	id สำหรับบ่งบอกหนังสือ	INT (10)

ตารางที่ A.19 ตารางอธิบายความหมายตาราง dc_relation

dc_relation		
ชื่อคอลัมน์	ความหมาย	ประเภท
DC_relation_id	id สำหรับบ่งบอกหนังสือที่เกี่ยวข้อง	INT (10) PK Auto_Increment
DC_relation	ชื่อหนังสือที่เกี่ยวข้อง	VARCHAR (191)
index_document_id	id สำหรับบ่งบอกหนังสือ	INT (10)

ตารางที่ A.20 ตารางอธิบายความหมายตาราง dc_keyword

dc_keyword		
ชื่อคอลัมน์	ความหมาย	ประเภท
DC_keyword_id	id สำหรับบ่งบอกคำสำคัญ	INT (10) PK Auto_Increment
DC_keyword	คำศัพท์	VARCHAR (191)
index_document_id	id สำหรับบ่งบอกหนังสือ	INT (10)

ตารางที่ A.21 ตารางอธิบายความหมายตาราง document

document		
ชื่อคอลัมน์	ความหมาย	ประเภท
document_id	id สำหรับบ่งบอกหนังสือ	INT(10) Auto_Increment
status_process_document	สถานะการทำงานของหนังสือ	INT (2)
name	ชื่อไฟล์ PDF หนังสือ	VARCHAR (191)
version	ครั้งที่ตีพิมพ์	INT (191)

page_start	หน้าหนังสือที่กำหนดเป็นหน้าเริ่ม	INT (191)
amount_page	จำนวนหน้าทั้งหมดของหนังสือ	INT (191)
path	ตำแหน่งไฟล์ PDF ที่ผู้ใช้งานอัปโหลดเข้าสู่ระบบ	TEXT
path_image	ตำแหน่งไฟล์รูปภาพของหนังสือ	TEXT
name	ชื่อไฟล์ PDF หนังสือ	VARCHAR (191)
DC_title	ชื่อหนังสือ	VARCHAR (191)
DC_title_alternative	ชื่อรองของหนังสือ	VARCHAR (191)
DC_description_table_of_contents	สาระสำคัญที่มาจากการบัญญัติ	TEXT
DC_description_note	รายละเอียดทั่วไปของหนังสือ	TEXT
DC_description_summary	สาระสำคัญของข้อมูลสารสนเทศที่ผ่านการค้นหา รวมรวม วิเคราะห์	TEXT
DC_description_abstract	ข้อมูลสรุปจากบทคัดย่อ วิทยานิพนธ์ และเนื้อหา	TEXT
DC_format	รูปแบบข้อมูลที่ถูกจัดเก็บในระบบ	VARCHAR (191)
DC_format_extent	ขนาดของไฟล์หนังสือ	VARCHAR (191)
DC_identifier_URL	แหล่งที่มาของหนังสือ	VARCHAR (191)
DC_identifier_ISBN	เลขมาตรฐานสากลของหนังสือ	VARCHAR (191)
DC_source	หน่วยข้อมูลต้นฉบับ	VARCHAR (191)
DC_language	ภาษาของหนังสือ	VARCHAR (191)
DC_coverage_spatial	สถานที่ของหนังสือที่เป็นเจ้าของ	VARCHAR (191)
DC_coverage_temporal	ช่วงเวลาในหน่วยปีของหนังสือ	VARCHAR (191)
DC_rights	ระดับการเข้าถึงของข้อมูล	VARCHAR (191)
DC_rights_access	ตำแหน่งที่มีสิทธิ์ในการเข้าถึงข้อมูล	VARCHAR (191)
thesis_degree_name	ชื่อเต็มของปริญญา	VARCHAR (191)
thesis_degree_level	ระดับของปริญญา	VARCHAR (191)
thesis_degree_discipline	สาขาวิชา	VARCHAR (191)
thesis_degree_grantor	มหาวิทยาลัย	VARCHAR (191)
index_creator	id สำหรับบุคคลซึ่งเป็นผู้เขียนหนังสือ	INT (10) FK
index_creator_orgname	id สำหรับบุคคลซึ่งเป็นหน่วยงานรับผิดชอบสังกัด	INT (10) FK
index_publisher	id สำหรับบุคคลซึ่งเป็นสำนักพิมพ์	INT (10) FK
index_publisher_email	id สำหรับบุคคลซึ่งเป็นสำนักพิมพ์	INT (10) FK
index_issued_date	id สำหรับบุคคลซึ่งเป็นผู้เชี่ยวชาญ	INT (10) FK
rec_create_at	วันเวลาของหนังสือที่ถูกนำเข้าสู่ระบบ	DATETIME (6) current_timestamp
rec_create_by	id สำหรับบุคคลซึ่งเป็นผู้ใช้งานที่นำหนังสือเข้าสู่ระบบ	INT (10) FK

rec_modified_at	วันเวลาของหนังสือที่ถูกแก้ไขข้อมูล	DATETIME (6) current_timestamp
rec_modified_by	id สำหรับบุบคอกผู้ใช้งานที่แก้ไขหนังสือในระบบ	INT (10) FK
rec_status	ค่าสถานะของหนังสือสำหรับการใช้งาน	INT (2)