โครงการระบบจัดหมวดหมู่องค์ความรู้แบบอัตโนมัติเพื่อพัฒนาการเรียนการสอน

2559:03

นายศุภณัฐ ทัตตินาพานิช, ณัฐ, 56070501053, zarkzaki@hotmail.com

นายอินทัช แสงกระจ่าง, อาร์ต, 56070501068, artkrub7@gmail.com

ที่ปรึกษาโครงงาน รศ.ดร. ธีรณี อจลากุล

วันที่ – 18 พฤศจิกายน 2559

_____

ข้าพเจ้าได้อ่านรายงานและตรวจเนื้อหาของรายงานเรียบร้อยแล้ว

Abstract

The education is an important part of improving Thai citizens' quality. It's a main tool for creating creativity to children for a sustainable country improvement. In Thailand, Thai government gives funds for researching in improving an education in school and the research data are stored in digital document files such as PDF files. The contents are about statistic, teaching techniques and so on. By the way, these files are not publicly published and cannot be searched due to their file type. So, if we created a knowledge sharing platform that can be used widely would be so helpful on making low quality schools performing better. This project provides a technology for collecting, analyzing, categorizing and extracting important parts of documents to make them searchable and be useful for educational society.

But, documents that stored in PDF format or text files are chunks of data that have no structure (schemaless). Therefore, these files cannot be analyzed by an ordinary software. So we decide to make an automatic document categorize system by using machine learning and text mining techniques to put each part of documents into groups. By using machines to categorize documents, we will be able to handle and categorize lots of documents from schools in Thailand.

In the first step of development, we need experts from Sodsri – Saritwong Foundation to read and tag cores of some documents into categories. This step will make a training data for system and the system will learn from this training data and create a machine learning model that learned about keywords for each category. Then the system will automatically tags each part of contents of documents that has been ingested into them. This machine learning system will reduce workload of humans a lot. Then we will create a web application that can search contents that are about a user's searching keyword. A user can read an important part of each document and can download a document from a web for more details.

บทคัดย่อ

การศึกษา ถือเป็นรากฐานที่สำคัญในการพัฒนาทรัพยากรมนุษย์ และช่วยในการพัฒนาประเทศชาติอย่างยั่งยืน ซึ่งในประเทศไทยนั้น รัฐบาลมีการให้ทุนสนับสนุนเพื่อการศึกษาค้นคว้า ทดลองหาวิธีการต่างๆ ที่จะช่วยเพิ่มประสิทธิภาพของระบบการศึกษาให้ดีขึ้น โดยผลที่ได้จากการวิจัยหรือทดลองนี้ จะถูกเขียนออกมาเป็นรูปเล่มรายงานและเก็บในรูปแบบของไฟล์ PDF ที่ประกอบไปด้วย รายงานสำหรับผู้บริหารที่แสดงถึงตัวเลขสถิติต่างๆ และรายงานสำหรับการเรียนการสอน ที่อธิบายรายละเอียดแนวทางการเรียนการสอน เป็นต้น อย่างไรก็ตาม เอกสารเหล่านี้ไม่ได้มีการเผยแพร่ในวงกว้าง และถูกเก็บในรูปแบบไฟล์ PDF ที่ไม่สามารถสืบค้นได้ จึงไม่สามารถถูกนำมาใช้ประโยชน์ได้อย่างเต็มที่ การสร้าง knowledge sharing platform สำหรับคุณครู ผู้ปกครอง และผู้บริหาร ที่สามารถสืบค้นหาข้อมูลได้อย่างสะดวกจึงเป็นเรื่องจำเป็น เพื่อให้การศึกษากระจายตัวได้อย่างทั่วถึง ข้อเสนอโครงการฉบับนี้จึงเสนอแนวคิดในการใช้เทคโนโลยีเพื่อรวบรวม คัดกรอง จัดหมวดหมู่ รวมถึงสกัดเนื้อหาส่วนที่สำคัญจากเอกสาร เพื่อให้เอกสารการทดลองด้านนวัตกรรมการเรียนการสอนสามารถถูกนำไปใช้ประโยชน์ และสร้างสังคมของการแลกเปลี่ยนเรียนรู้ของคุณครูและผู้ปกครองได้

ทั้งนี้ ข้อมูลที่อยู่ในไฟล์ PDF หรือไฟล์ text นั้นเป็นข้อมูลที่ไม่มีโครงสร้าง (Schemaless) ไม่สามารถนำมาประมวลผลเพื่อวิเคราะห์ด้วยซอฟท์แวร์ทั่วๆไปได้ จึงจำเป็นต้องมีการพัฒนาระบบจัดหมวดหมู่องค์ความรู้แบบอัตโนมัติขึ้น โดยอาศัยเทคโนโลยีทางด้าน Text Mining และเทคโนโลยีทางด้าน Machine Learning เข้ามาช่วยเพื่อให้สามารถใช้งานกับรายงานจำนวนมากจากทั่วประเทศได้อย่างมีประสิทธิผล

ในการพัฒนาระบบในระยะเริ่มต้นนั้น จะมีคุณครูอาสาสมัครจากทางมูลนิธิสดศรี-สฤษดิ์วงศ์เข้ามาช่วยอ่านและทำการระบุข้อความส่วนที่เป็นเนื้อหาใจความสำคัญของรายงานนั้นๆ และสร้าง Tag เพื่อบ่งบอกหัวเรื่องของเนื้อหา เพื่อใช้สำหรับการจัดหมวดหมู่ โดยคุณครูจะช่วยวิเคราะห์รายงานเพียงส่วนน้อยเท่านั้น หลังจากนั้นเนื้อหาและ Tag ที่คุณครูสร้างขึ้นจะถูกนำมาพัฒนา Machine Learning Model โดย Model จะถูกสอนให้เรียนรู้คำต่างๆที่เกี่ยวข้องกับ tag ที่อยู่ในรายงาน เมื่อมีรายงานเล่มใหม่เข้ามาในระบบ ระบบจะทำการวิเคราะห์เนื้อหาและสามารถแสดงส่วนที่เป็นใจความสำคัญ รวมถึงจัดประเภทหมวดหมู่ของรายงานได้โดยอัตโนมัติ ซึ่งช่วยลดการใช้เวลาและทรัพยากรบุคคล เมื่อผู้ใช้งานเข้ามาใช้ระบบนี้ จะสามารถสืบหาข้อมูลที่เกี่ยวข้องกับการพัฒนาการเรียนการสอน ด้วยการใส่ Keyword จากนั้น web application จะให้ผลลัพธ์ออกมาเป็นข้อความที่เกี่ยวข้องพร้อมทั้งแนบลิงค์สำหรับดาวน์โหลดเอกสาร ผู้ใช้งานสามารถอ่านสรุปใจความสำคัญที่ระบบแสดงก่อน และหากตรงกับความสนใจสามารถดาวน์โหลดรายงานทั้งเล่มไปเพื่อศึกษารายละเอียดต่อไป

กิตติกรรมประกาศ

การที่โครงการระบบจัดหมวดหมู่องค์ความรู้แบบอัตโนมัติเพื่อพัฒนาการเรียนการสอนนี้ สามารถดำเนินงานจนสำเร็จลุล่วงมาถึงขั้นนี้ได้นั้น เป็นเพราะความกรุณาของทางมูลนิธิสดศรี-สฤษดิ์วงศ์ โดยการประสานของของพี่หญิง ผู้ซึ่งให้ความช่วยเหลือทั้งทางด้านเอกสารที่นำมาใช้ คุณครูที่มาช่วยทั้งในด้านการจัดกลุ่มเอกสารและด้านอื่นๆ และยังให้ความช่วยเหลือด้านเงินทุนในการดำเนินโครงการมาด้วย และขอขอบคุณทางสถาบันอาศรมศิลป์ โดยการประสานงานของคุณอภิษฎา ทองสอาด หรือพี่ปุ้ม ที่ช่วยให้คำแนะนำในด้านเอกสาร และให้เอกสารเพิ่มเติมจากทางมูลนิธิสดศรี-สฤษดิ์วงศ์ อีกด้วย ทางกลุ่มจึงขอขอบคุณองค์กรทั้งสองมาไว้ ณ ที่นี้

สารบัญ

สารบัญรูปภาพ

บทที่ 1 บทนำ

บทที่ 2 ที่มา ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

บทที่ 3 การออกแบบและระเบียบวิธีวิจัย

บทที่ 4 ผลการวิจัยและอภิปรายผล

บทที่ 1 คำนำ

1.1 ที่มาของปัญหาและแนวทางการปัญหา

จากสถิติการศึกษาขั้นพื้นฐานของประเทศไทยในรายงานประจำปีของ World Economic Forum ปี 2014-2015 [1] พบว่าประเทศไทยอยู่ในลำดับที่ 90 จาก 144 ประเทศทั่วโลกที่ได้รับการจัดอันดับ ซึ่งถือได้ว่าอยู่ในลำดับค่อนข้างต่ำ ในขณะเดียวกันผลการวิเคราะห์ในรายงานของ International Institute of Management Development และ Pearson-The Economist Intelligence Unit พบว่าการศึกษาของไทยถูกจัดให้อยู่ในกลุ่มต่ำสุดเช่นเดียวกัน ซึ่งรายงานเหล่านี้ล้วนเป็นตัวบ่งชี้ให้เห็นว่าการศึกษาของไทยยังมีจุดบกพร่องอีกมาก ควรที่จะต้องได้รับการพัฒนาอย่างเร่งด่วน

หนึ่งในปัจจัยสำคัญที่มีผลต่อการพัฒนาการศึกษา คือการสอนของคุณครู เพราะคุณครูเปรียบเสมือนผู้ที่ถ่ายทอดความรู้ต่างๆและพัฒนาเด็กให้เติบโตไปเป็นทรัพยากรมนุษย์ที่มีคุณภาพ ดังนั้น รัฐบาลไทยจึงมีการให้ทุนสนับสนุนกับครูในการศึกษาค้นคว้า ทดลองหาวิธี ในการพัฒนาการเรียนการสอนให้มีประสิทธิภาพและเข้าถึงเด็กนักเรียนได้มากขึ้น โดยเฉพาะการศึกษาขั้นพื้นฐานในระดับประถามและมัธยมศึกษา ซึ่งในปัจจุบัน ได้มีเอกสารที่ถูกเขียนออกมาเพื่อรายงานผลการทดลอง และวิธีในการพัฒนาการเรียนการสอนที่ดี (best practice) ซึ่งรายงานเหล่านี้มักจะหนาและอยู่ในรูปแบบของไฟล์ PDF ทำให้ครูสืบค้นข้อมูลได้ยาก และต้องเสียเวลาในการอ่านหนังสือหลายร้อยหน้าจำนวนหลายเล่มเพราะเอกสารไม่มีการรวบรวมและจัดเป็นหมวดหมู่ ทำให้ประสิทธิภาพในการสืบค้นข้อมูลนั้นไม่ดี อีกทั้งยังอาจได้ข้อมูลที่ไม่ครบถ้วน

ทางผู้จัดทำจึงจะทำการรวบรวมเอกสารรายงานเหล่านี้ เพื่อทำให้เกิดเป็น knowledge sharing platform ที่คุณครูสามารถเข้ามาสืบค้นหามูล และศึกษาค้นคว้าได้อย่างง่าย ระบบดังกล่าวจะช่วยรวบรวมและทำการจัดหมวดหมู่เอกสาร รวมทั้ง ทำการวิเคราะห์ คัดแยกเนื้อหาส่วนต่างๆ ในไฟล์เอกสารและทำการ tag ข้อความสำคัญให้โดยอัตโนมัติ ว่าเนื้อหาในแต่ละส่วนมีความเกี่ยวข้องกับเรื่องอะไรบ้าง และทำการจัดเก็บข้อมูลเหล่าลงไปยังระบบฐานข้อมูล ความรู้จากเอกสารเหล่านี้จะได้ถูกนำไปพัฒนาการเรียนการสอน และพัฒนาให้การศึกษาของไทยก้าวไปสู่ในระดับต้นๆของโลกได้ในอนาคต

1.2 วัตถุประสงค์

- เพื่อสร้างฐานข้อมูลที่ก่อให้เกิดเป็น knowledge sharing platform ของประเทศไทย
- เพื่อให้คุณครู ผู้บริหาร และผู้ปกครองสามารถสืบค้นข้อมูลได้อย่างมีประสิทธิภาพและรวดเร็ว และสามารถนำองค์ความรู้ไปพัฒนาการเรียนการสอนได้

- เพื่อศึกษาการทำ Document และ Text Clustering สำหรับการคัดแยกเนื้อหาและจัดหมวดหมู่เนื้อหาภายในเอกสาร
- เพื่อสร้าง Machine Learning Model สำหรับเรียนรู้เอกสารภาษาไทยทางด้านการศึกษา
- เพื่อสร้าง Web Application สำหรับการจัดการองค์ความรู้

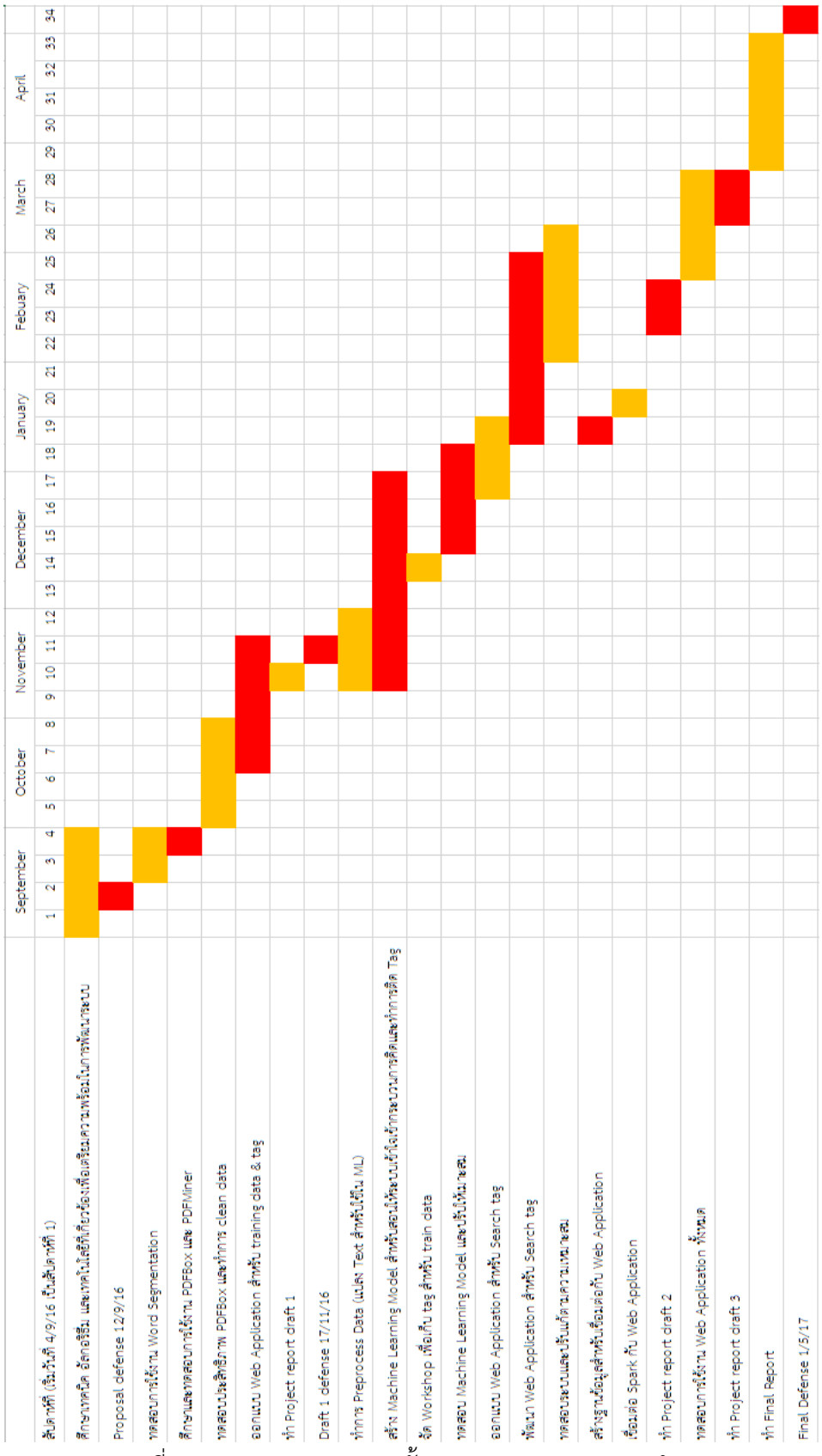
1.3 ขอบเขตของงานวิจัย

เป้าหมายของโครงการโครงการสร้างระบบจัดหมวดหมู่องค์ความรู้แบบอัตโนมัติด้วยเทคโนโลยี Text Mining และ Machine Learning คือการสร้าง knowledge sharing platform ที่จะช่วยรวบรวมข้อมูลรายงานตัวอย่างการเรียนการสอนที่ดี (best practice) มาทำการคัดแยก จัดหมวดหมู่ เพื่อให้คุณครูสามารถสืบค้นข้อมูลได้อย่างมีประสิทธิภาพและรวดเร็ว สามารถนำความรู้ไปพัฒนาและปรับใช้ตามเพื่อให้เกิดการเรียนการสอนที่ดีที่จะช่วยพัฒนาศักยภาพของเด็กนักเรียนได้ โดย platform มีขอบเขต ดังนี้

- สร้างเครื่องมือสำหรับรับไฟล์ PDF/text และดึงข้อความภาษาไทยในแต่ละย่อหน้าออกมา โดยข้อความจะถูกประมวลผล Text และใส่เข้าใน Machine Learning Model เพื่อติด Tag ที่เกี่ยวข้องกับเนื้อหาต่อไป
- สร้าง Machine learning model ที่สามารถรับข้อมูลจากไฟล์ text ภาษาไทยที่ได้จากขั้นตอนที่ 4.1 และทำการติด tag สำหรับแต่ละย่อหน้า โดยวิธีการ supervised classification ซึ่งจะแบ่งเป็น 2 ขั้นตอนคือ
    - เครื่องมือสำหรับการ train model โดยจะรับ text และ tag ของย่อหน้าต่างจากผู้เชี่ยวชาญ และนำมาปรับจนได้โมเดลทางคณิตศาสตร์ที่เหมาะสม
    - เครื่องมือสำหรับการทำนายหรือการติด tag อัตโนมัติด้วยโมเดลทางคณิตศาสตร์ที่พัฒนาขึ้นจะรับข้อความภาษาไทยเป็นอินพุธและทำการติด tag ให้ย่อหน้าต่างๆที่อธิบายประเด็นสำคัญของหนังสือ/รายงานแต่ละเล่ม ผลลัพธ์จะถูกเก็บลงในฐานข้อมูลเพื่อการสืบค้นต่อไป
- สร้าง Web application โดยมีคุณสมบัติหลักคือ
    - สามารถรับไฟล์รายงานในรูปแบบไฟล์ text หรือ pdf เพิ่มเติมจากผู้ใช้ และนำไปประมวลผลโดยใช้ Machine Learning Model ซึ่งจะทำการติด tag โดยอัตโนมัติและเก็บลงฐานข้อมูล

- สามารถสืบค้นข้อมูลตัวอย่างการเรียนการสอนที่ดี โดยผ่านการพิมพ์ข้อความหรือคีย์เวิร์ดลงในช่อง Search หรือผ่านการคลิกเลือกจากเมนู Advance Filter หรือจากการคลิกบนหน้าจอแผนที่ได้
- สามารถแสดงผลการสืบค้นในหลายระดับ คือประเภทเอกสาร หมวดหมู่ของ tag และตัวอย่างข้อความที่สำคัญ นอกจากนี้ผู้ใช้ระบบจะสามารถดาวน์โหลดไฟล์ต้นฉบับออกจากระบบได้หากต้องการ
- ผู้ใช้ระบบสามารถให้ review และ กด Like ข้อความและแชร์บทความที่น่าสนใจบน Facebook ได้ ซึ่งระบบจะนำข้อมูล Review/Like/Share ไปใช้ในการจัดอันดับข้อความและรายงานต่อไป

1.4 ขั้นตอนการทำงานและระยะเวลาการดำเนินงาน



ภาพที่ 1.1 Gantt Chart แสดงขั้นตอนและระยะเวลาการทำงาน

บทที่ 2 ที่มา ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 การทบทวนวรรณกรรม

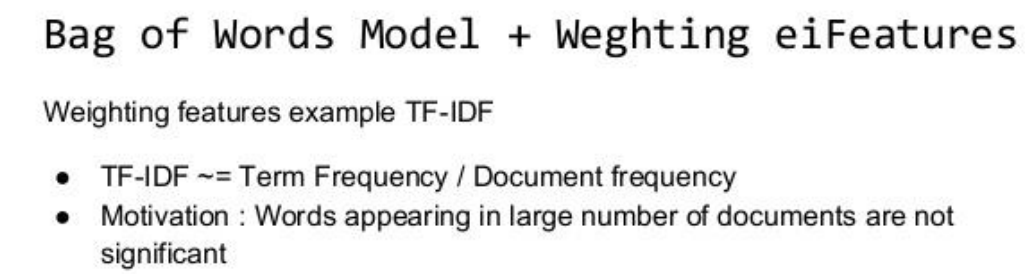
ในปัจจุบันนี้ มีโปรแกรมสำหรับการแปลง unstructured information(เอกสาร,รูปภาพ) ให้เป็น structured information(SQL tables) โดยใช้ machine learning ในการแปลงข้อมูลคือ Deepdive Stanford University จะเป็นโปรแกรมที่สามารถอ่านข้อมูลในหลากหลายรูปแบบ เช่น ข้อความในรูปแบบ text file หรือข้อมูลที่อยู่ในฐานข้อมูล แล้วสามารถนำข้อมูลต่างๆ เหล่านั้นมาเชื่อมโยงกันโดยใช้ machine learning และนำมาทำการวิเคราะห์ข้อมูลต่างๆได้ โดยใช้หลักการทำ document clustering และการทำ Topic Discovery ต่างๆ เช่น การนำบทความที่เขียนไว้และฐานข้อมูลมาสรุปผลร่วมกัน ซึ่งนอกจาก Deepdive [2] แล้ว จะมีโปรแกรมสำหรับดึงข้อมูลจาก unstructured information ได้แก่ AlchemyLangage API [3] ซึ่งใช้ IBM Watson ในการทำ Machine Learning โดยจะสามารถอ่านข้อมูลที่เป็น text file ต่างๆ โดยใช้ข้อมูลเหล่านั้น เทียบกับ public model หรือ Custom model โดยผลลัพธ์ที่ได้ออกมาจากการใช้ Alchemy API ได้แก่ Sentiment ของคำ, Name Entity Recognition และ Keywords ต่างๆ เป็นต้น หรือ Aylien [4] ที่เป็นโปรแกรมที่รับ text file และทำการตรวจสอบคำสำคัญ, สรุปของบทความ หรือการสร้าง hashtag จาก model ของทางระบบที่สร้างไว้ ซึ่งโดยส่วนใหญ่ของโปรแกรมเหล่านี้ จะรองรับสำหรับภาษาในภาษาอังกฤษหรือภาษาที่รากศัพท์มาจากภาษาละติน เนื่องจากมี Library ในการจัดการทางภาษาศาสตร์จาก NLP Stanford

สำหรับเครื่องมือและเทคโนโลยีต่างๆที่ทางกลุ่มได้เลือกใช้ในขั้นตอนต่างๆ จะแบ่งเป็นการทำ Word Segmentation ซึ่งในปัจจุบันมีโปรแกรมสำหรับตัดคำภาษาไทยต่างๆได้แก่ LexTo เป็นโปรแกรมในการจัดคำที่จะใช้วิธี Dictionary base ในการที่จะเลือกแบ่งคำจากประโยค และ TLex เป็นโปรแกรมในการตัดคำภาษาไทยโดยใช้ machine learning ชื่อว่า Condition Random Fields โดยทางกลุ่มได้เลือกใช้ Lexto สำหรับการตัดคำเนื่องจากมีการเพิ่มคำใหม่ต่างๆ การใช้ Dictionary base จะง่ายกว่าในการเรียนรู้, การทำ Topic Discovery นั้น จะมีวิธีการทำ Clustering สำหรับ Text Analysis 2 ตัว ได้แก่ Latent Dirichlet Allocation [5] โดยจะเป็นการทำ topic discovery จากข้อมูลต่างๆ โดยจะตรวจสอบเนื้อหาภายในนำมาเปรียบเทียบกับเอกสารอื่นๆใน topic ที่เกี่ยวข้อง และ Latent semantic analysis [6] จะทำการสร้าง matrix สำหรับเก็บจำนวน frequency ของคำและใช้วิธีการทางคณิตศาสตร์เรียกว่า Singular value decomposition (SVD) ในการลดจำนวนมิติของ array ใน matrix เพื่อหาค่าที่มีความเกี่ยวข้องมากที่สุดจากในเอกสารนั้นๆ ซึ่งทางกลุ่มได้เลือกใช้ LDA เนื่องจากสามารถตรวจสอบหาความเกี่ยวเนื่องระหว่างเอกสารได้

ดีกว่า [7] และสุดท้ายการทำ Classification จะมีเทคนิคต่างๆ เช่น One-vs-Rest สามารถให้ผลลัพธ์การจัดกลุ่มได้หลายผลลัพธ์ (multiclass classification) โดยจะใช้วิธีการจัดกลุ่มข้อมูลนั้นๆ เข้าในแต่ละหมวดหมู่แล้วทำการเปรียบเทียบผลกับหมวดหมู่อื่นๆ แล้วเลือกผลลัพธ์ที่ทำให้การจัดกลุ่มมีความแม่นยำสูงที่สุด โดยการจัดกลุ่มอันนี้จะทำให้ข้อมูล 1 ย่อหน้าสามารถมี tag ได้หลายอย่าง, Neural network เป็น classification algorithm ที่เลียนแบบการทำงานของระบบประสาทของมนุษย์ และสุดท้าย Decision Tree เป็น rule-based classification คือการสร้างต้นไม้ของกฎต่างๆ เพื่อที่จะจัดกลุ่มข้อมูล โดยทางกลุ่มได้เลือกใช้ Neural Network [8] เนื่องจากเป็น machine learning algorithm ที่มีความยืดหยุ่นสูง และมีการปรับปรุงประสิทธิภาพของ model ได้เรื่อยๆ ระหว่างที่กำลังทำงานอยู่ ซึ่งต่างจาก rule-based algorithm ที่จะตายตัวเมื่อการสร้าง model เสร็จสิ้น ซึ่งสามารถสรุปเทคนิคและเทคโนโลยีที่ใช้ได้ดังต่อไปนี้

2.1.1 **Word Segmentation** [9] การแบ่งคำในภาษาต่างๆ เช่นภาษาอังกฤษหรือภาษาที่ใช้ Latin Alphabet จะมีการแบ่งคำโดยการใช้ space เป็น word delimiter และภาษาญี่ปุ่นหรือจีนที่คำที่มีความหมายในตัวเองและมีการทำ sentence delimiter แต่สำหรับภาษาไทยนั้น Sentense และ Word จะไม่มี delimiter ใดๆเลย ทำให้จำเป็นต้องใช้โปรแกรมสำหรับการทำ Word Segmentation เพื่อที่จะระบุคำว่าในรูปประโยคนี้จะมีคำใดบ้าง และวิธีการทางการทำ Word Segmentation นั้นจะมีการทำ Forward Step และ Backward Step สำหรับการหาค่าความน่าจะเป็นของรูปประโยคเพื่อให้ได้ผลลัพธ์ที่ดีที่สุดในการแบ่งคำbag-of-words model เป็นโมเดลในการทำ mapping ของคำต่างๆให้กลายเป็นตัวเลข เพื่อที่จะสามารถนำไปใช้ประโยชน์ในเชิงคณิตศาสตร์และการทำสถิติต่างๆต่อไป

2.1.2 **Bag-of-words model** [10] เป็น Model ที่ใช้วิธีการเก็บตัวอักษรในเอกสารและทำการ mapping กับตัวเลข เพื่อใช้เป็นตัวแปรสำหรับนำไปใช้ประโยชน์สำหรับการทำ Mathmethic Model และการทำ Machine Learning และการทำ Bag-of-words Model นั้น จะเป็นส่วนหนึ่งของขั้นตอนแรกการ TF-IDF สำหรับการนับความถี่ของคำและสร้าง model N-gram ได้อีกด้วย



	Mary	Loves	Movies	Cinema	Art	John	Went	to	the	Delicatessen	Robert	Football	Game	and	for
d1	0.3779	0.3779	0.3779	0.3779	0.3779									0.0001	
d2							0.4402	0.001	0.02			0.4558	0.458		
d3			0.001				0.01		0.01		0.458				0.0001

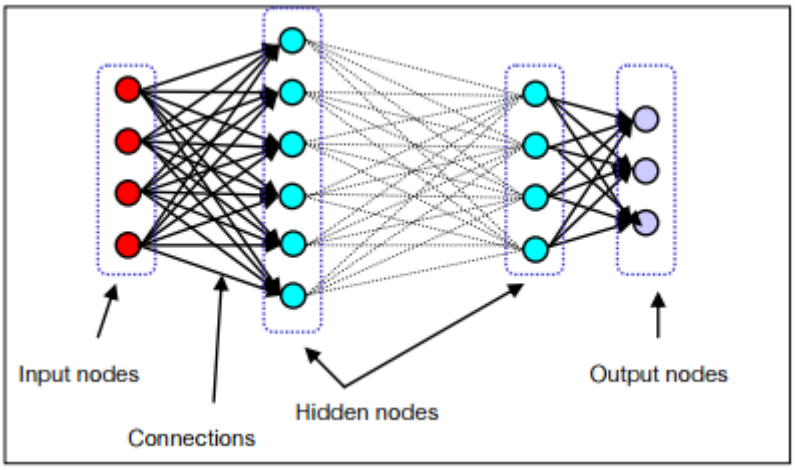
ภาพที่ 2.1 – ตัวอย่างการทำ Bag of Words Model [10]

2.1.3 **Term frequency – Inverse document frequency (TF-IDF)** [11] เป็นวิธีทางสถิติที่จะทำการตรวจสอบคำต่างๆในบทความเพื่อนำไปเปรียบเทียบกับบทความทั้งหมด เพื่อหาอัตราส่วนว่าคำๆนี้มีความสำคัญต่อบทความโดยรวมมากน้อยแค่ไหน โดย TF-IDF จะแบ่งขั้นตอนเป็น 2 ส่วนคือ Term frequency โดยในขั้นตอนนี้จะทำการนับจำนวนครั้งที่คำต่างๆปรากฏในบทความหนึ่งๆ และการทำ Inverse document frequency โดยในขั้นตอนนี้จะเป็นนำคำต่างๆในบทความมาเปรียบเทียบกับบทความทั้งหมดและคำนวณหาค่าน้ำหนักความสำคัญนั้นๆจากบทความทั้งหมด โดยการทำ TF-IDF สามารถใช้ประโยชน์ในการหาคำสำคัญในบทความต่างๆ ซึ่งสามารถนำไปประยุกต์ใช้ได้อย่างหลากหลายเช่น การทำ Search engine หรือการทำ Text Summarization

2.1.4 **Latent Dirichlet Allocation** เป็น clustering algorithm ที่ใช้สำหรับการทำ topic discovery จากข้อมูลต่างๆ ที่ใส่เข้าไป ซึ่งจะมีการเรียกใช้ vector ของคำที่ได้จากการทำ bag-of-word model มาทำการหาความถี่ของคำเทียบกับเอกสารต่างๆ และทำการแปลงสร้าง model ความเกี่ยวข้องของคำต่างๆ เทียบกับเอกสารอื่นๆที่ได้ทำการเรียนรู้ เพื่อค้นหา Keyword ที่สำคัญสำหรับนำไปใช้งานต่อ ซึ่ง LDA นั้นจะมองเอกสารเป็นการรวมกันของ topics ต่างๆที่ซ่อนอยู่ โดยแต่ละ topic จะมีค่าคำต่อความน่าจะเป็น ซึ่งจะบ่งบอกคำนี้มีความเกี่ยวข้องกับ topic ดังกล่าวมากน้อยเพียงใด โดยจะใช้สำหรับการดึง tag ที่เกี่ยวข้องต่างๆจาก paragraph เพื่อนำไปใช้สำหรับการ train model ในขั้นตอนการทำ classification

2.1.5 **Neural network** [12] เป็น machine learning algorithm ที่มีหลักการทำงานที่เลียนแบบการทำงานของโครงสร้างในระบบประสาทของมนุษย์ โดยมีการส่งข้อมูลที่ทำการเรียนรู้อยู่ในระบบเข้าสู่ node ต่างๆ และหาค่าน้ำหนักในแต่ละ node แล้วทำการส่งข้อมูลไปยัง node ย่อยๆ ต่างๆ ไป

เรื่อยๆ จนได้ผลลัพธ์การจัดกลุ่มที่ดีที่สุด โดยการทำ neural network จะช่วยทำให้การระบุว่า tag เรื่องหนึ่งๆ มีความเกี่ยวข้องกับ paragraph ที่เรียนรู้หรือไม่ มีความแม่นยำในระดับที่น่าพึงพอใจ



ภาพที่ 2.2 – ลักษณะโครงสร้างของ Neural Network

2.2 ภาษา, เครื่องมือ และซอฟต์แวร์ที่ใช้ในการพัฒนา

2.2.1 **Hadoop Distributed File System (HDFS)** [13] เป็นระบบการจัดเก็บข้อมูลที่ออกแบบมาสำหรับการจัดการข้อมูลขนาดใหญ่ (Big data) โดย HDFS ถูกออกแบบมาสำหรับระบบที่มีคอมพิวเตอร์หลายๆ ตัวช่วยกันประมวลผล และ HDFS จะเหมาะกับการทำงานในลักษณะ “Write once, Read many” หรือข้อมูลที่เน้นการอ่านข้อมูลมากกว่าการเขียน,แก้ไข โดยลักษณะการทำงานของ HDFS ที่กล่าวไปข้างต้นนั้น มีความเหมาะสมกับรูปแบบการใช้งานของโครงการนี้เป็นอย่างมาก เนื่องจากข้อมูลที่เข้ามาในระบบนั้น จะถูกเขียนลงไปเพียงครั้งเดียว ไม่มีการแก้ไข และมีการอ่านข้อมูลขึ้นมาหลายๆ ครั้งในระหว่างการทำ Machine learning ซึ่งเข้ากันได้ดีกับรูปแบบการใช้งาน HDFS

2.2.2 **โปรแกรม Spark ML** [14] เป็น library ที่มีอยู่ในโปรแกรม Apache Spark ซึ่ง Spark ML เป็น libraryที่ใช้ทำ Machine Learning โดยที่สามารถทำงานแบบขนาน (Parallel programming) ได้ ซึ่ง Apache Spark เป็น engine สำหรับการทำการประมวลผลข้อมูลขนาดใหญ่ (Big data processing) ที่สามารถทำงานได้อย่างรวดเร็ว เนื่องจากใช้การประมวลผลในหน่วยความจำหลัก (In-memory processing) ทำให้การเข้าถึงข้อมูลทำให้รวดเร็วมากขึ้น ซึ่ง Spark ML นี้เป็น Machine learning library ที่ถูกใช้งานร่วมกับ big data platform อย่าง Hadoop กันอย่างแพร่หลาย และมีประสิทธิภาพในการทำงานสูง ทำให้ทางกลุ่มเลือกใช้โปรแกรมนี้

2.2.3 **โปรแกรม Apache Impala** [15] เป็นโปรแกรมจัดการฐานข้อมูลแบบ SQL ที่เป็น Open source ที่ถูกออกแบบมาให้ใช้งานร่วมกับ Hadoop ecosystem โดย Impala จะเหมาะกับการเก็บข้อมูลที่ต้องการนำมาวิเคราะห์แบบรวดเร็ว เนื่องจากตัวโปรแกรมมี latency ต่ำและมี throughput ที่สูง

และยังมีความสามารถในการเพิ่มประสิทธิภาพของระบบได้ง่าย (Scalable) ซึ่งฐานข้อมูลของโครงงานนี้มีปริมาณมาก และต้องการความรวดเร็วในการใช้งานเวลาผู้ใช้ค้นหาข้อมูล ทำให้ Impala มีความเหมาะสมกับงานมากที่สุด ทั้งด้านความสามารถในการจัดการฐานข้อมูลขนาดใหญ่ และประสิทธิภาพในการเรียกใช้งานข้อมูล

2.2.4 **โปรแกรม Apache HBase** [16] เป็นโปรแกรมจัดการฐานข้อมูลแบบ NoSQL ที่เป็น Open source ที่ถูกใช้งานร่วมกัน Hadoop ecosystem โดยฐานข้อมูลแบบ NoSQL จะมีความยืดหยุ่นด้านโครงสร้างมากกว่าฐานข้อมูลแบบ SQL ดังนั้น ทางกลุ่มจึงนำ HBase มาใช้งานร่วมกันกับ Hive เพื่อเก็บข้อมูลที่เหมาะสมลงในฐานข้อมูลแต่ละโปรแกรม โดย HBase จะเก็บข้อมูลจำพวกเนื้อหาของแต่ละเอกสารที่ถูกแบ่งย่อหน้าแล้ว ซึ่งจำนวนย่อหน้าของแต่ละเอกสารจะมีไม่เท่ากัน ดังนั้นฐานข้อมูลแบบ NoSQL จึงเหมาะสมกับการเก็บข้อมูลลักษณะนี้ ส่วน Hive ที่เป็นฐานข้อมูลแบบ SQL จะจัดเก็บข้อมูลเรื่อง tag ของแต่ละย่อหน้าไว้ เพื่อให้สามารถทำการ Query ผ่านหน้าเว็บไซต์ได้อย่างรวดเร็ว

2.2.5 **โปรแกรม PDFBox** [17] เป็นโปรแกรมสำหรับการแปลงไฟล์ในรูปแบบ PDF ให้เป็น text file ซึ่ง PDFBox เป็นโปรแกรมภาษา Java ที่ใช้สำหรับการดึงข้อมูลต่างๆออกมาจาก PDF Document เช่น ตัวอักษรในภาษาต่างๆ เช่น ไทย อังกฤษ จีน และอื่นๆ หรือสามารถดึงภาพออกจาก PDF ได้ โดยสำหรับโปรเจคนี้จะเน้นที่การดึงข้อความออกจาก PDF Document เพื่อสำหรับนำไป preprocess ต่อ ซึ่งใช้ Extract text function สำหรับการดึง text line ออกจาก PDF

2.2.6 **โปรแกรม LexTo** เป็นโปรแกรมที่ถูกพัฒนาด้วยภาษา Java โดยโปรแกรมนี้สามารถใช้ในการแบ่งคำต่างๆในภาษาไทยจากประโยคให้กลายเป็นคำซึ่งแบ่งด้วย delimiter ซึ่งคำต่างๆที่ใช้ในการแบ่งนั้นจะมี Dictionary ที่จะทำการเก็บคำทั้งหมดเอาไว้ แล้วโปรแกรมจะนำมาเปรียบเทียบเพื่อแบ่งคำตามที่ Dictionary ได้กำหนดไว้ ซึ่ง LexTo เป็นโปรแกรมตัดคำภาษาไทยแบบ Open Source ที่ทางกลุ่มสามารถนำมาใช้งานได้ และมีความแม่นยำในระดับที่พอรับได้ ทำให้ทางกลุ่มเลือกใช้โปรแกรม LexTo

2.2.7 **ภาษา Java** เป็นภาษาโปรแกรมในลักษณะ Object-Oriented Programming ที่ได้รับความนิยมสูงสุดในปัจจุบัน ถูกเลือกนำมาใช้ในการเขียนโปรแกรมสำหรับทำ Text preprocessing เนื่องจากมีความยืดหยุ่นในการทำงานสูง

2.2.8 **ภาษา Python** เป็นภาษาโปรแกมที่ทำงานในลักษณะ Scripting language โดยนำมาใช้ร่วมกับโปรแกรม SparkML ที่ใช้ในการทำ machine learning

2.2.9 **ภาษา PHP** เป็นภาษาโปรแกรมในที่ทำงานในลักษณะ Server Scripting language ซึ่งใช้ในการทำ Web Application เพื่อใช้ในการติดต่อกับทาง Database ผ่าน ODBC และ JDBC

บทที่ 3 การออกแบบและระเบียบวิธีวิจัย

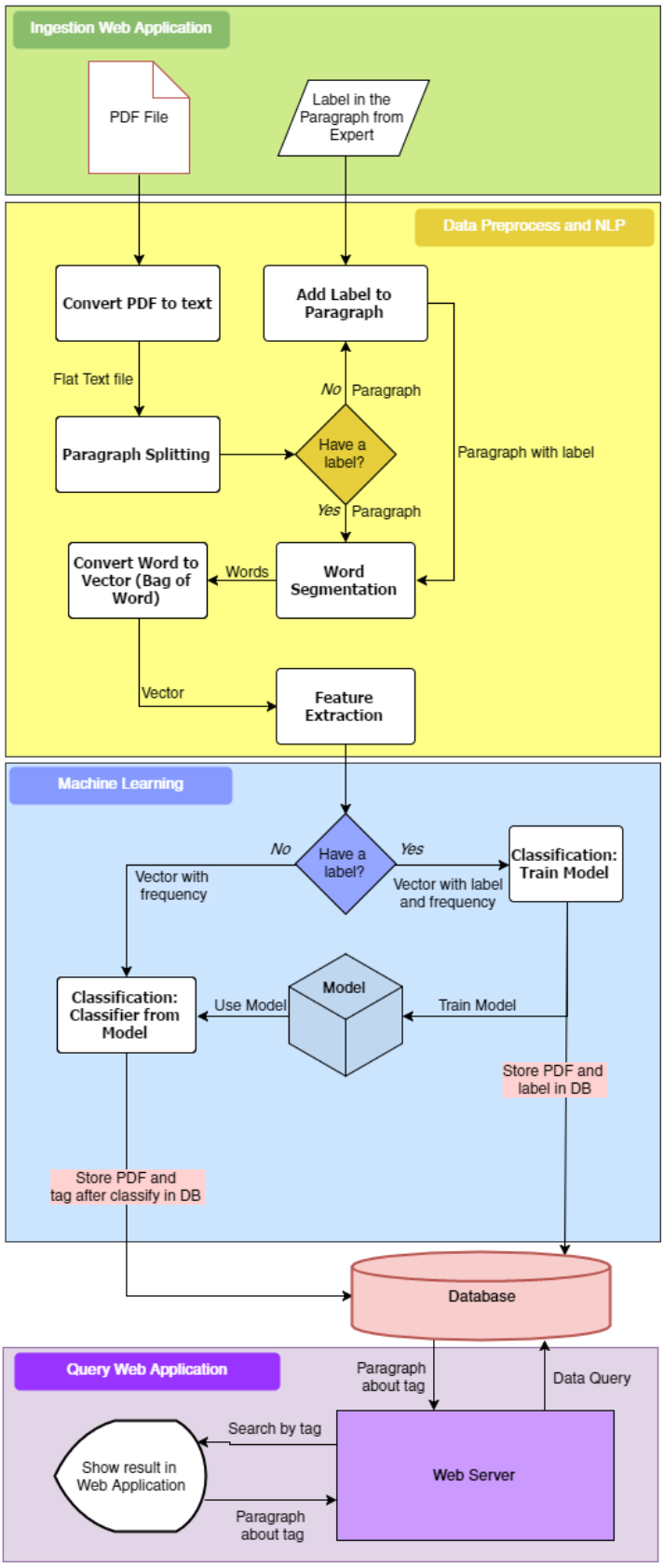
3.1 ขั้นตอนการทำงาน

เมื่อต้องการที่จะค้นหาเอกสารที่เกี่ยวข้องกับเรื่องใดเรื่องหนึ่งขึ้นมาใช้งาน ผู้ใช้จะสามารถค้นหาข้อมูลได้อย่างรวดเร็วด้วย platform ที่ทางกลุ่มพัฒนาขึ้น โดย platform ที่พัฒนาขึ้นจะแบ่งออกได้เป็น 2 ส่วนหลักๆ

- ส่วนของผู้พัฒนาและผู้ดูแลระบบ โดยผู้พัฒนาจะทำการเตรียมเอกสารที่เป็นไฟล์ PDF ตัวอย่าง ซึ่งเอกสารเหล่านี้จะมีผู้เชี่ยวชาญเฉพาะมาช่วยในการระบุคำสำคัญต่างๆเพื่อทำการเตรียม machine learning model โดยหลังจากที่ทำการสร้าง model เสร็จแล้ว เอกสารที่เหลือจะทำการ tag เอกสารได้โดยอัตโนมัติ โดยใช้ machine learning model ข้างต้น เช่น ถ้าต้องการให้ตัว model สามารถทำการจำแนกเนื้อหาที่เกี่ยวข้องกับเรื่อง “การดึงความสนใจนักเรียน” ผู้พัฒนา/ผู้ดูแลจะต้องเตรียมเอกสารที่มีเนื้อหาที่เกี่ยวข้องกับเรื่องการดึงความสนใจของนักเรียน และให้ผู้เชี่ยวชาญช่วยระบุว่า มีคำใดบ้างที่สามารถระบุได้ว่า ข้อความนี้มีความเกี่ยวข้องกับ "การดึงความสนใจของนักเรียน" และนำไปทำการเตรียม model โดยหลังจากสร้าง model เสร็จแล้ว ผู้พัฒนาสามารถนำเอกสารที่เกี่ยวข้องกับ "การดึงความสนใจนักเรียน" มาทำการ tag เอกสารโดยอัตโนมัติได้

- ส่วนของผู้ใช้งาน ผู้ใช้สามารถเข้ามาใช้งานผ่าน web application ที่ทางกลุ่มพัฒนาขึ้นมา แล้วทำการค้นหาเนื้อหาที่เกี่ยวข้องกับสิ่งที่ผู้ใช้ต้องการ แล้วเนื้อหาส่วนนั้นก็จะปรากฏขึ้นมา และมีไฟล์เอกสารนั้นให้ผู้ใช้สามารถ download ไปอ่านได้ ยกตัวอย่างเช่น ครูสมศรีต้องการที่จะหาข้อมูลเรื่อง “การดึงความสนใจนักเรียน” เพื่อนำไปเตรียมการเรียนการสอนสำหรับชั้นเรียน สิ่งที่คุณครูต้องทำก็คือ ค้นหาด้วยคำว่า “ดึงความสนใจนักเรียน” ในหน้าเว็บ แล้วเว็บก็จะทำการแสดงผลย่อหน้าที่เกี่ยวข้องกับการดึงความสนใจนักเรียนจากเอกสารต่างๆในระบบ รวมถึงแสดง tag ที่เกี่ยวข้องกับย่อหน้านั้นๆ โดยแต่ละย่อหน้าก็จะมี tag ที่เกี่ยวข้องเป็นของตัวเอง และมีลิงค์สำหรับดาวน์โหลดเอกสารที่มีข้อความนั้นอยู่ให้คลิกเพื่อดาวน์โหลดได้

Flowchart

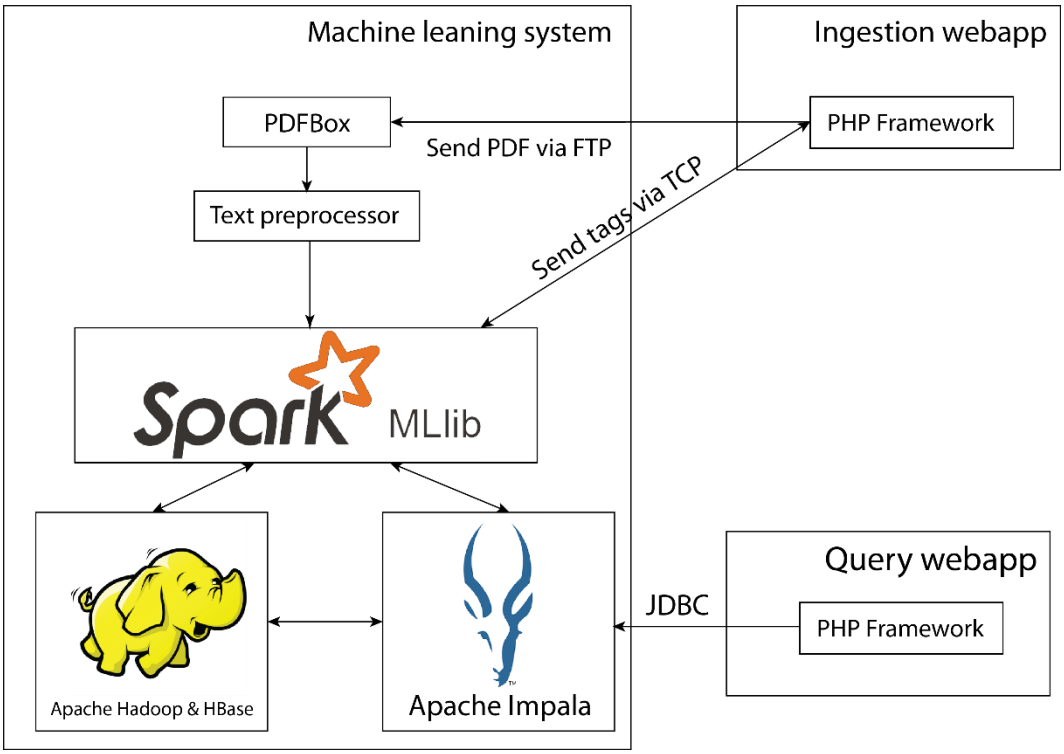


ภาพที่ 3.1 ลำดับการทำงานของซอฟต์แวร์

ส่วนประกอบในการทำงาน จะแบ่งขั้นตอนต่างๆออกเป็น 5 ส่วน ได้แก่

1. **Ingestion Web Application** โดยจะแบ่งวิธีการรับข้อมูลเป็น 2 แบบคือ 1.ทำการรับ PDF กับ Label สำหรับใช้ใน Train Model และ 2. รับ PDF อย่างเดียวสำหรับ Predict Tag จาก PDF นั้น
2. **Preprocessing Data** โดยในขั้นตอนนี้จะทำการเปลี่ยน PDF ที่รับมาให้กลายเป็น Flat Text และทำการรับ Label หากเป็นขั้นตอนการทำ Train Model ซึ่งหลังจากได้ Text มาแล้ว จะทำการเปลี่ยน Text เหล่านั้นมาสร้าง Vector ของคำ ซึ่งสำหรับภาษาไทยนั้น จำเป็นต้องมีการตัดแบ่งคำ (Word Segmentation) สำหรับประโยคออกเพื่อทำ NER (Name Entity Recognition) และทำการตัดคำต่างๆที่ไม่มีความหมายต่างๆทิ้งไปได้แก่ คำเชื่อม เช่น และ, หรือ, กับ เป็นต้น และคำขยายความต่างๆ เช่น การ ความ เป็นต้น และสุดท้ายจะทำ bag-of-word เพื่อสร้าง vector ของคำและนำไปใช้ในขั้นตอน Topic Discovery
3. **Topic Discovery** จะเป็นการดึงคำสำคัญหรือความเกี่ยวข้องต่างๆที่อยู่ใน paragraph ออกมา เช่น การทำ TF-IDF เพื่อหาความถี่ของคำ และการลดมิติของคำให้เหลือเพียงคำสำคัญต่างๆโดยการใช้ Machine Learning: Clustering คือ Latent Dirichlet Allocation
4. **Machine Learning: Classification** เป็นการสร้าง Machine สำหรับการจำแนกผลลัพธ์จากข้อมูลที่เข้ามา โดยจะแบ่งขั้นตอนการใช้งานเป็น 2 ขั้นตอนได้แก่ 1.การทำ Training และ Testing Model โดยในขั้นตอนนี้จะนำคำต่างๆที่ได้จากขั้นตอนข้างต้น รวมกับ label ที่ผู้เชี่ยวชาญได้ระบุไว้มาสร้าง Model สำหรับการ Classification ออกมา 2.การ Prediction จากเอกสารต่างๆ เพื่อให้ได้ผลลัพธ์ออกมาเป็น tag เพื่อนำไปใช้ในการสืบค้นใน Database ต่อไป โดยเทคนิค Classification คือ Neural Network
5. **Query Web Application** จะเป็นการสร้าง Web Application เพื่อติดต่อกับ Database โดยตัว Web Application นั้น จะทำการ Query Tag ที่ต้องการสืบค้นจาก Database แล้วนำมาแสดงผล

3.2 สถาปัตยกรรม



ภาพที่ 3.2 สถาปัตยกรรมของซอฟต์แวร์

จากรูปข้างต้น จะเห็นได้ว่าโครงสร้างของตัวโปรแกรมจะถูกแบ่งออกเป็น 3 ส่วนหลักๆ ได้แก่ ส่วนของระบบในการทำ Machine learning, ส่วนของหน้าเว็บที่ใช้ในการรับ PDF และส่วนของหน้าเว็บที่ใช้ในการค้นหาข้อมูลจาก Tag โดยส่วนของระบบ Machine learning จะประกอบไปด้วย

- โปรแกรม PDFBox ที่ใช้ในการแปลงไฟล์ PDF ให้เป็นไฟล์ข้อความ
- ส่วนของโปรแกรมที่ใช้ในการจัดเตรียมข้อความเพื่อที่จะนำไปใช้ในการทำ Machine learning ได้แก่ โปรแกรมสำหรับจัดเรียงข้อมูลที่ไม่เรียบร้อย (data cleaning), โปรแกรมแบ่ง paragraph และโปรแกรม LexTo
- โปรแกรม Spark MLlib ซึ่งเป็นโปรแกรมที่ใช้ในการทำ machine learning สำหรับ hadoop ecosystem
- โปรแกรม Apache Hadoop และ HBase ที่ใช้ในการจัดเก็บข้อมูลเกี่ยวกับไฟล์ PDF และเนื้อหาภายในไฟล์นั้น
- โปรแกรม Apache Impala เป็นโปรแกรมจัดการฐานข้อมูลที่ใช้ในระบบ hadoop ecosystem

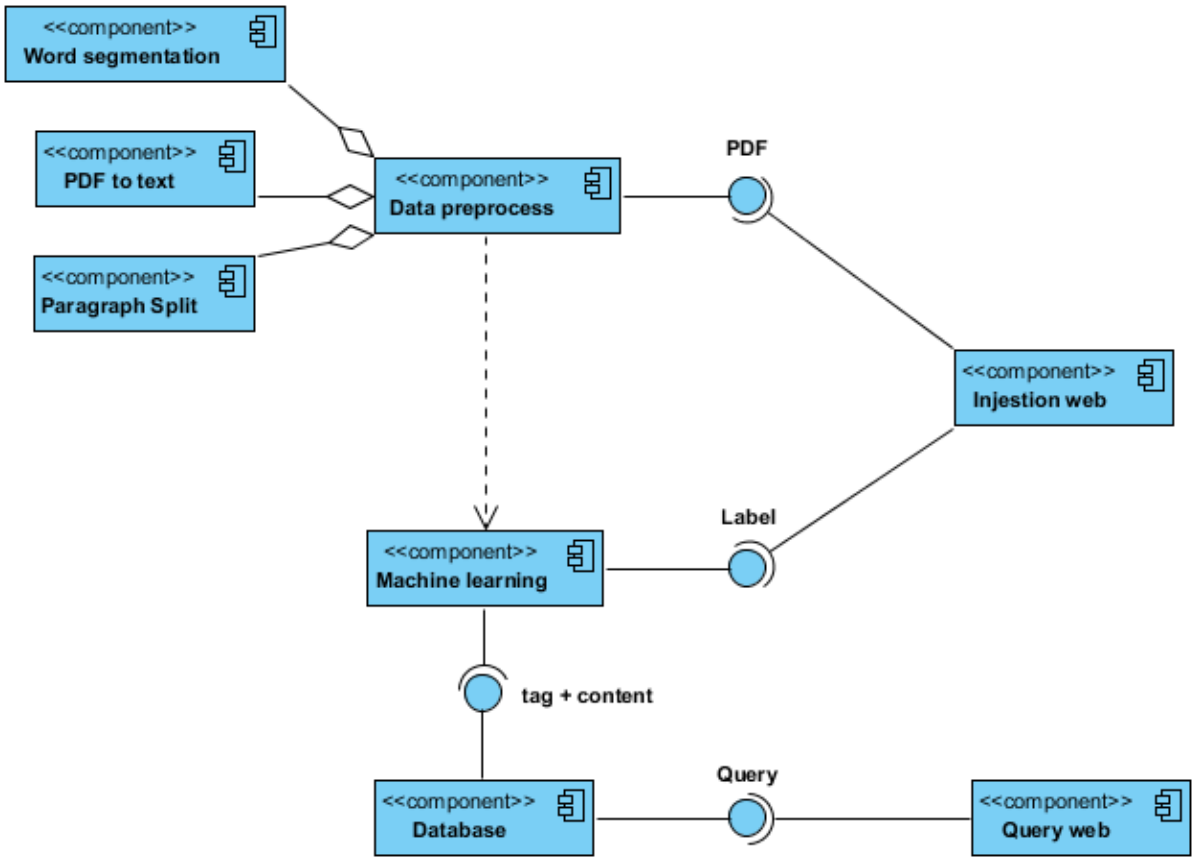
ส่วนต่อมาเป็นส่วนของหน้าเว็บที่ใช้ในการรับ PDF ซึ่งจะพัฒนาขึ้นด้วยภาษา PHP โดยมีหน้าที่รับไฟล์ PDF ที่อัพโหลดขึ้นมาจากผู้ใช้งาน และ tag ของเนื้อหาในไฟล์ PDF นั้น (ในกรณีที่เป็นไฟล์ PDF ที่ใช้ใน

การเรียนรู้ระบบ) แล้วทำการส่งมาที่เครื่องที่ทำการทำ machine learning ด้วย FTP protocol และส่งข้อความ tag ด้วย TCP protocol

ส่วนสุดท้ายเป็นส่วนของหน้าเว็บที่จะใช้ผู้ใช้ค้นหาเนื้อหาจาก tag ซึ่งพัฒนาขึ้นด้วยภาษา PHP เช่นเดียวกัน โดยหน้าเว็บจะรับ tag ที่ผู้ใช้ต้องการค้นหามาแล้วไปทำการ Query ใน Impala ออกมาแสดงผลให้ผู้ใช้งาน โดยติดต่อผ่าน OBDC

3.3 ลักษณะของการออกแบบซอฟต์แวร์

Component Diagram



ภาพที่ 3.3 Component Diagram ของซอฟต์แวร์

จาก component diagram ข้างต้น จะเห็นได้ว่า โปรแกรมมีการแบ่งข้อมูลเป็น 3 ส่วนหลักๆ ได้แก่ ส่วนรับไฟล์ PDF เข้ามาในระบบ, ส่วนการทำ machine learning และส่วนของการ query ผลลัพธ์ออกมาแสดงผล โดยแต่ละส่วนจะมีการรับ-ส่งข้อมูลดังต่อไปนี้

Ingestion Web Application

- Input - PDF file อย่างเดียว หรือ PDF file ที่มีการระบุ tag ในแต่ละ paragraph แล้ว
- Output - Notification ว่ามีการรับ Input File สำเร็จแล้ว

Machine Learning: Train Model

- Input - PDF File ที่มีการระบุ tag ในแต่ละ paragraph โดยผู้เชี่ยวชาญเฉพาะทาง
- Output - Model ที่สามารถคาดเดา tag จาก paragraph และ Tag ที่ได้จาก pdf เก็บลงใน Database

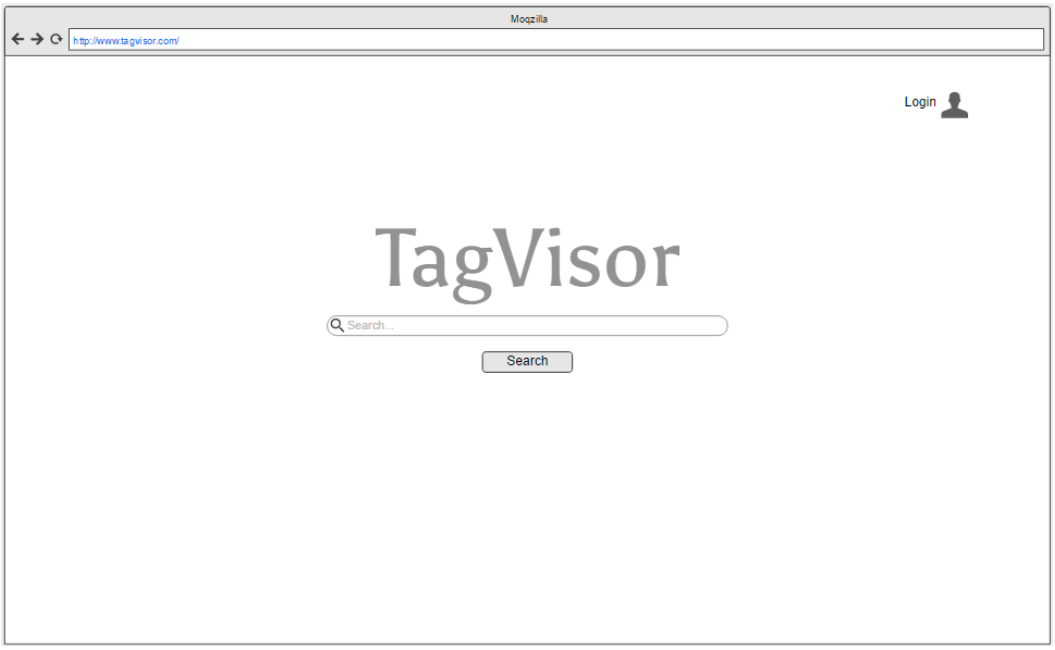
Machine Learning: Prediction

- Input - PDF File
- Output - ข้อมูล Tag ที่ได้จากการ Prediction จาก PDF ที่เป็น Input โดยใช้ model ที่สร้างขึ้น และเก็บข้อมูล PDF และ Tag ใหม่ที่ได้จากการ Prediction ลงใน Database

Query Web Application

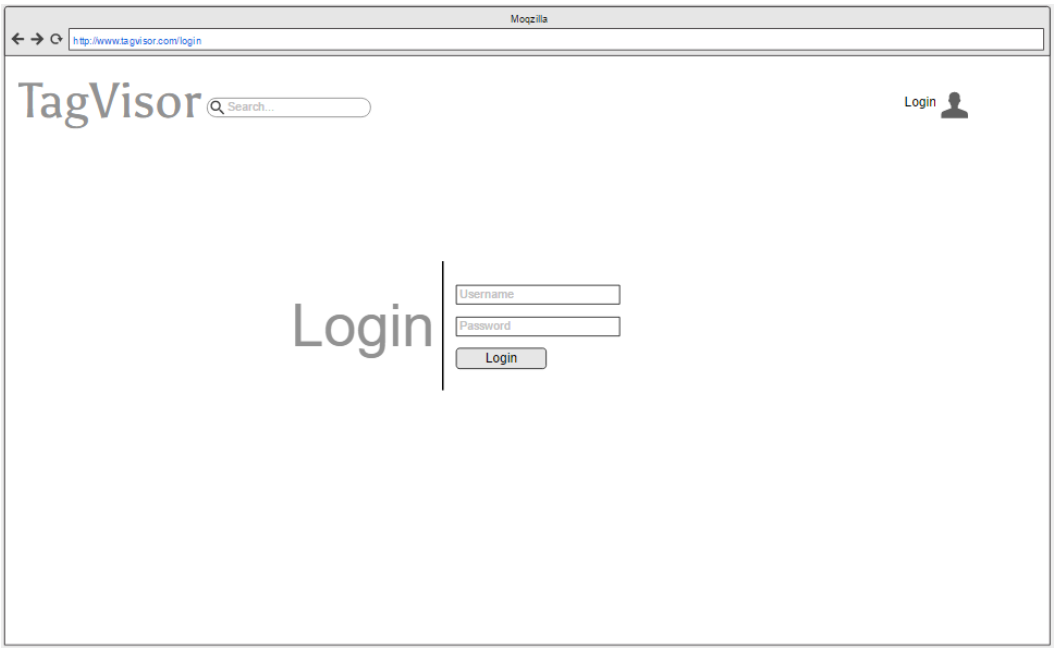
- Input - tag ที่ต้องการจะสืบค้น
- Output - ย่อหน้าที่มีความเกี่ยวข้องกับ tag นั้นๆ และข้อมูลเกี่ยวกับย่อหน้านั้น ได้แก่ tag ของย่อหน้านั้นทั้งหมด, เอกสารที่เขียนข้อความนั้น และ link download เอกสารนั้นในรูปแบบไฟล์ PDF

ตัวอย่าง Web Application ที่ได้ออกแบบไว้



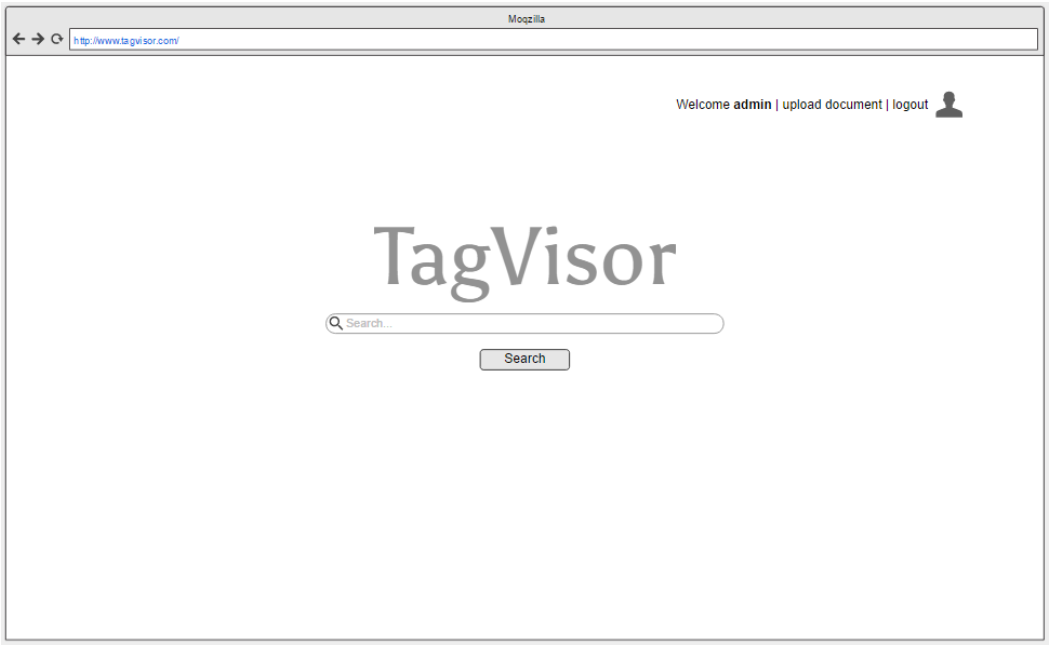
ภาพที่ 3.4 main page

หน้าจอหลักของ web application เมื่อยังไม่ทำการล็อกอิน จะมีส่วนให้ค้นหาข้อมูลจาก tag ต่างๆ และปุ่มสำหรับล็อกอิน



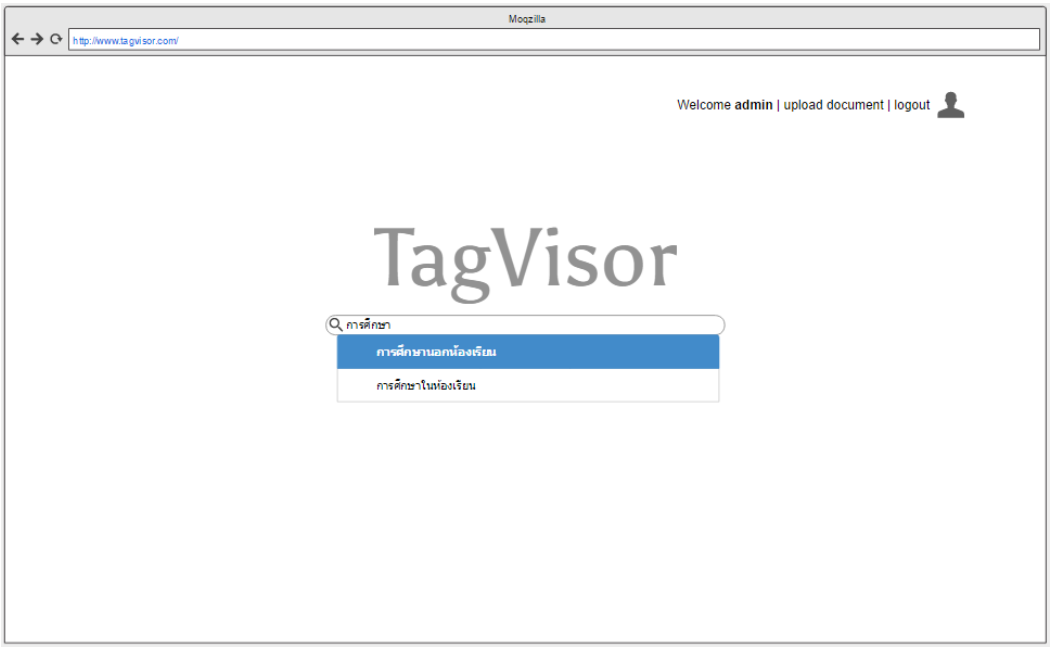
ภาพที่ 3.5 Log In

หน้าจอสำหรับการล็อกอินเข้าใช้งานระบบ เพื่อให้ User กรอก Username และ Password สำหรับใช้งาน



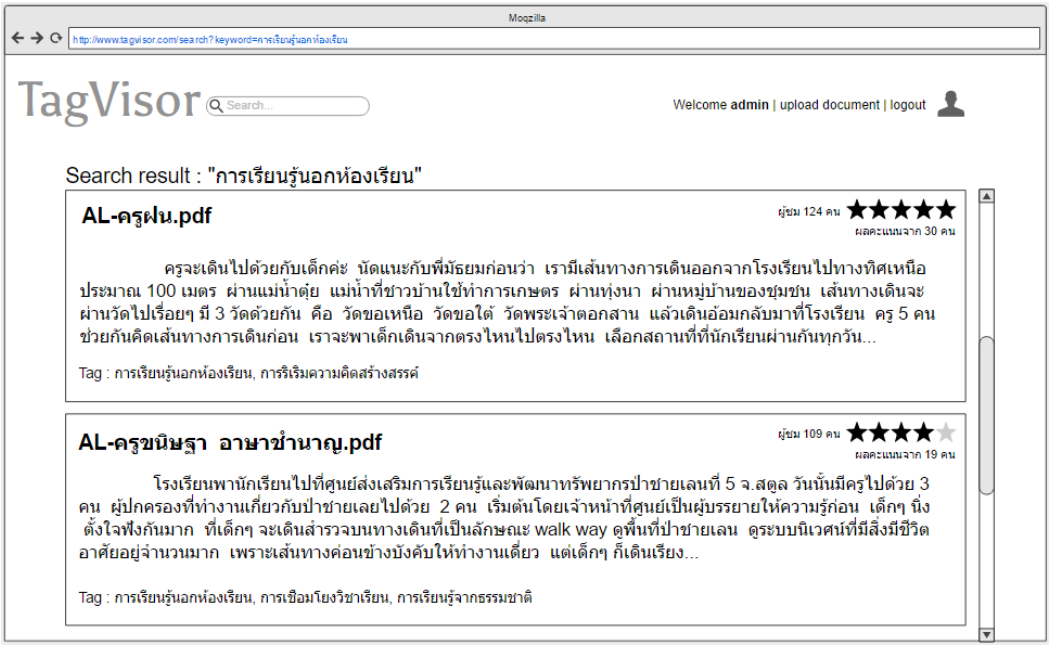
ภาพที่ 3.6 หน้าจอหลักที่มีการ Log In

หน้าจอหลักของ web application เมื่อทำการล็อกอินแล้ว จะมีปุ่มสำหรับให้ upload เอกสารเพิ่มเข้ามา



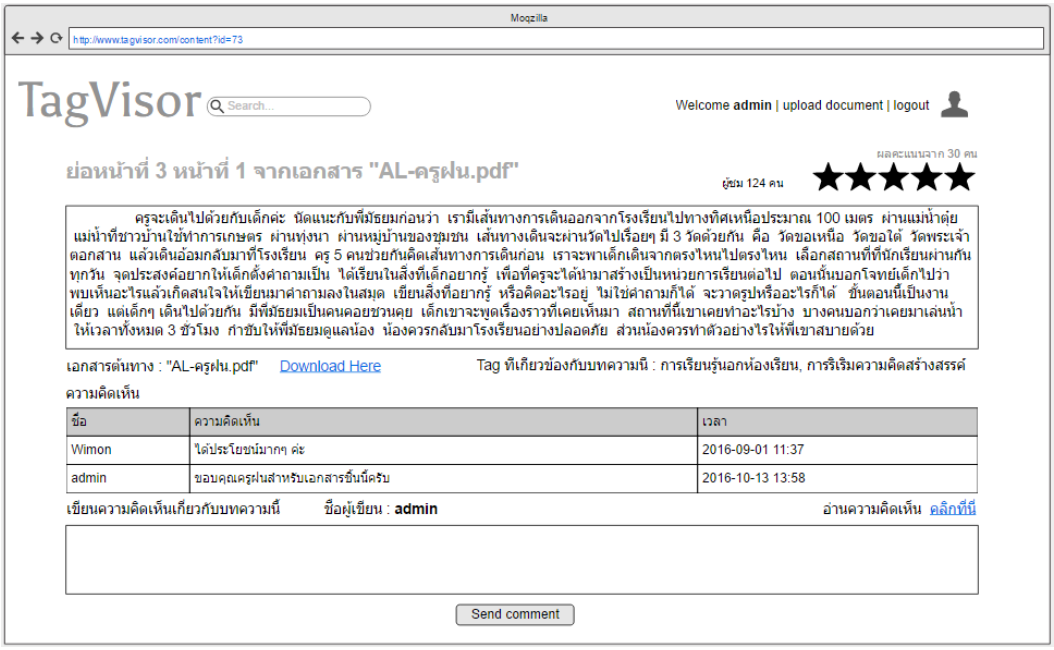
ภาพที่ 3.7 หน้าจอระหว่างการค้นหา

เมื่อผู้ใช้งานทำการพิมพ์สิ่งที่ต้องการค้นหามาบางส่วน จะมีขึ้น Suggestion ให้ผู้ใช้งานเลือกค้นหา



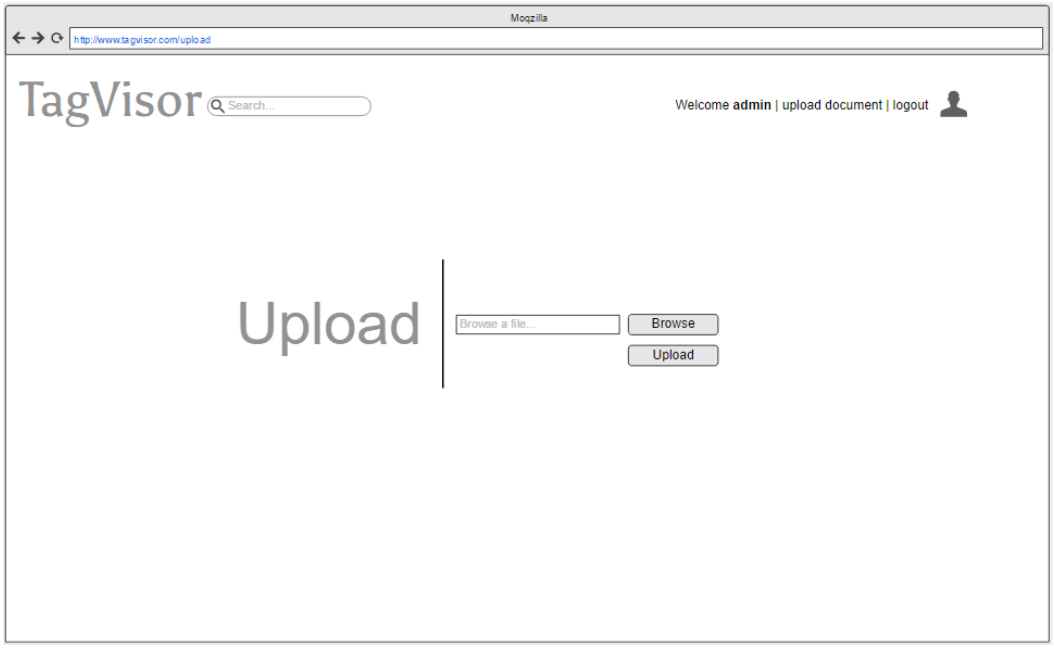
ภาพที่ 3.8 หน้าจอผลการค้นหา

หน้าผลการค้นหา จะแสดงผลเนื้อหาบางส่วนที่เกี่ยวข้องกับหัวข้อที่ค้นหา ชื่อไฟล์เอกสารที่มีข้อความนั้น tag ทั้งหมดของเนื้อหาส่วนนั้น ผู้เข้าชมเนื้อหานั้น และคะแนนที่ผู้ใช้แต่ละคนมอบให้เอกสารนั้นๆ

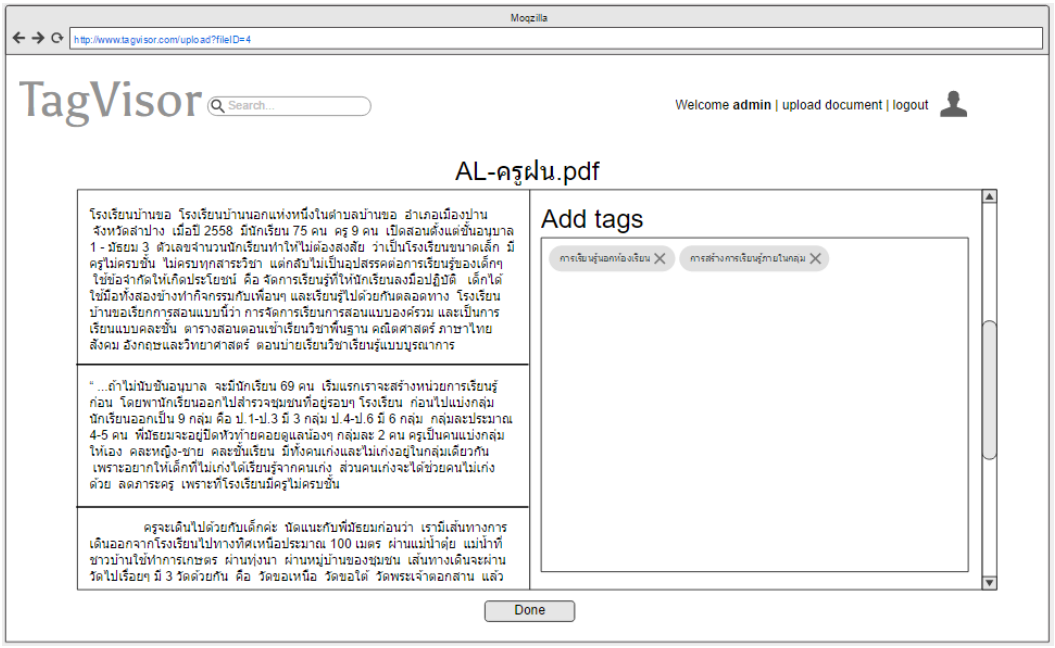


ภาพที่ 3.9 หน้าจอแสดงรายละเอียดของ Paragraph

เมื่อคลิกเข้าไปดูรายละเอียดเพิ่มเติมของเนื้อหา จะแสดงผลเนื้อหาส่วนนั้นๆ แบบเต็ม ความคิดเห็นของผู้ใช้ต่อบทความนั้น และมีลิงค์ให้ดาวน์โหลดเอกสารนั้นๆ



ภาพที่ 3.10 หน้าจอส่วนของการอัพโหลดเอกสารขึ้นระบบ



ภาพที่ 3.11 หน้าจอสำหรับใส่ tag

ถ้าเป็นส่วนของเอกสารที่อัพโหลดเพื่อให้ machine learning นำไปเรียนรู้ จะมีส่วนของหน้าจอที่ให้ผู้ใช้ระบุว่า ข้อความแต่ละส่วนที่ตัดมานั้น มีเนื้อหาเกี่ยวข้องกับเรื่องอะไรบ้าง โดยผู้ใช้จะคลิกเลือกข้อความแต่ละชุด และใส่ tag ที่คิดว่าเกี่ยวข้องทั้งหมดลงไป

3.4 ข้อจำกัดของซอฟต์แวร์

- เนื่องจากภาษาไทยเป็นภาษาที่มีความซับซ้อนสูง ทั้งทางด้านตัวอักษร ที่มีสระบน-ล่าง และทางด้านรูปประโยคที่ไม่มีความแน่นอน ทำให้การเขียนโปรแกรมที่สามารถประมวลผลภาษาไทยได้อย่างสมบูรณ์แบบจึงเป็นเรื่องยาก ดังนั้นความแม่นยำในการ tag อาจจะต่ำกว่าการใช้งานกับภาษาอังกฤษที่มีรูปประโยคและการตัดคำที่แน่นอนกว่า
- ข้อมูลที่จะนำไปเข้าระบบ Machine Learning เพื่อให้ระบบทำการเรียนรู้ด้วยตนเองนั้น จะต้องใช้มนุษย์เป็นตัวช่วยในการกำหนดข้อมูลก่อนในเบื้องต้น (Supervised learning) เพราะฉะนั้น ถ้าเราต้องการให้ระบบเรียนรู้เนื้อหาเรื่องใหม่ๆ จะต้องมีการใช้ผู้เชี่ยวชาญที่เกี่ยวข้องกับเรื่องที่จะให้ระบบเรียนรู้มาช่วยทำการ Tag ย่อหน้าก่อนที่จะนำข้อมูลเข้าไปในระบบ ความเข้าใจและการทำงานของผู้เชี่ยวชาญในการ Train Machine จึงมีความสำคัญเป็นอย่างยิ่ง
- การระบุย่อหน้าจาก PDF นั้นสามารถทำได้ยาก เนื่องจากการระบุย่อหน้าจาก PDF จำเป็นต้องใช้ค่าตำแหน่งของตัวอักษรต่างๆ เพื่อระบุว่าย่อหน้าควรจะอยู่ตำแหน่งไหน ซึ่ง PDF ที่ได้รับมานั้น มีรูปแบบการจัดหน้าและ font ที่แตกต่างกันรวมถึงรูปแบบคำภาษาไทยและภาษาอังกฤษในเอกสาร จะทำให้ตำแหน่งของคำเกิดการคลาดเคลื่อนซึ่งจะส่งผลให้ย่อหน้าที่ได้ออกมาอาจเกิดความผิดพลาดได้

3.3 ลักษณะเด่นของซอฟต์แวร์

- ระบบสามารถรองรับเอกสารที่เป็นภาษาไทยได้ ซึ่งในปัจจุบันนั้น ตามที่เราได้หาข้อมูลมายังไม่มีq ซอฟแวร์ใดๆ ที่สามารถทำการแยกเอกสารภาษาไทยและนำไปเก็บข้อมูลลงในระบบฐานข้อมูล อีกทั้งระบบสามารถรับเอกสารที่จะนำเข้าไปในระบบได้ทั้ง Text File เช่น Microsoft Word และเอกสารที่มีการเก็บในรูปแบบอื่นที่ไม่ใช่ Text File เช่น PDF File
- ระบบมีการใช้ Machine Learning: Classification ในการทำ Prediction เพื่อหา Tag ทำให้การจำแนก Tag จะมีมาตรฐานตามที่ได้กำหนดไว้ และสามารถทำให้ระบบเรียนรูปเอกสารชนิดต่างๆ มากยิ่งขึ้นเพื่อเพิ่มความถูกต้องในการจำแนกเอกสารได้
- ระบบใช้ Hadoop Ecosystem ในการเก็บข้อมูลเอกสารต่างๆ โดยใช้ Impala และ Hbase และในการประมวลผลข้อมูล (การทำ Machine Learning) โดยใช้ Spark ทำให้สามารถรองรับข้อมูลจำนวนมากได้

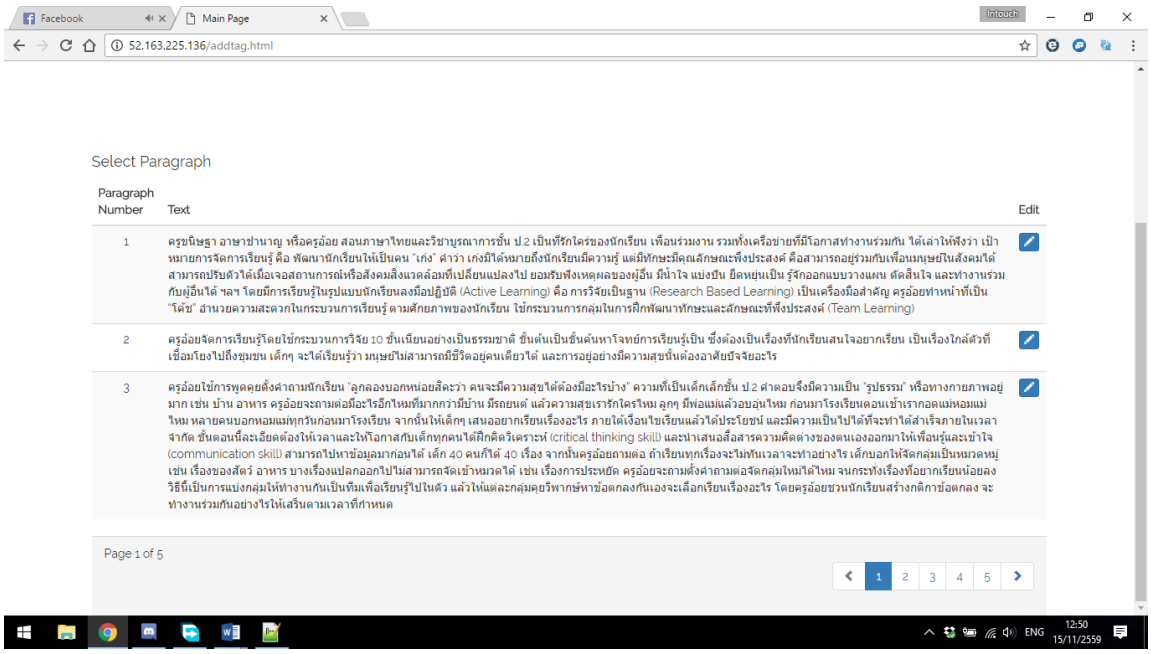
บทที่ 4 ผลการวิจัยและอภิปรายผล

4.1. ตัวอย่างภาพหน้าจอของโปรแกรม

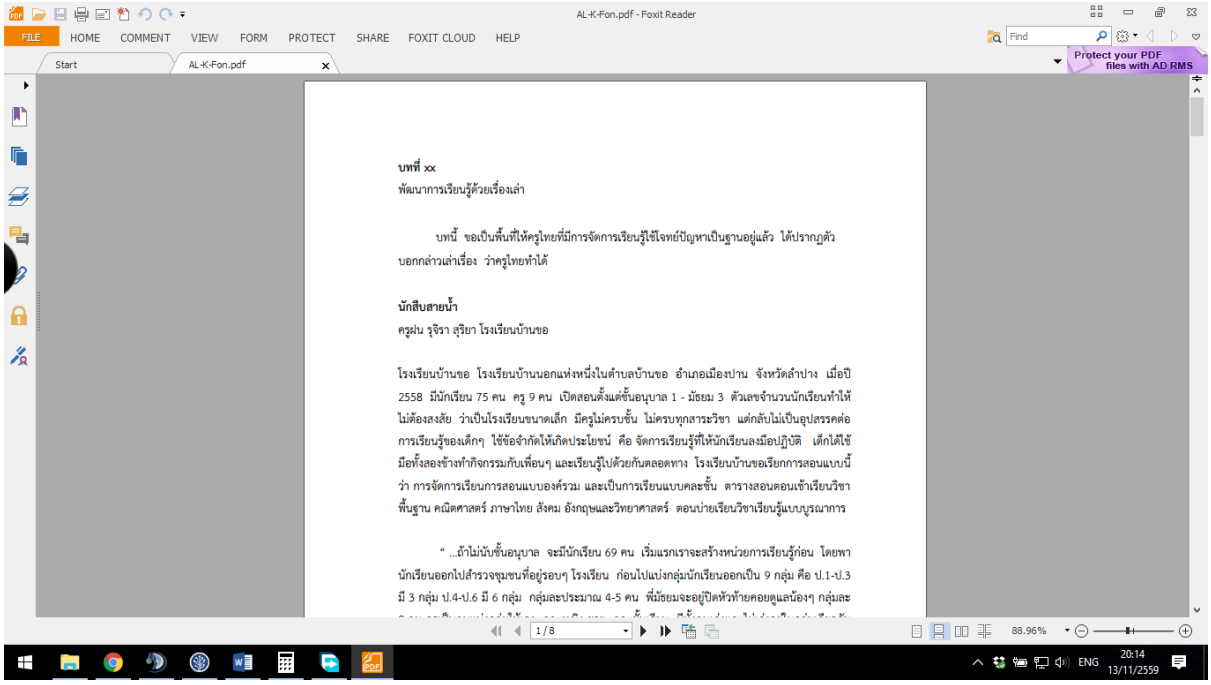
ภาพที่ 4.1 ภาพตัวอย่างของหน้าเว็บสำหรับการ Search

ภาพที่ 4.2 ภาพตัวอย่างของหน้าเว็บสำหรับการ Upload เอกสารเข้าไปในระบบ

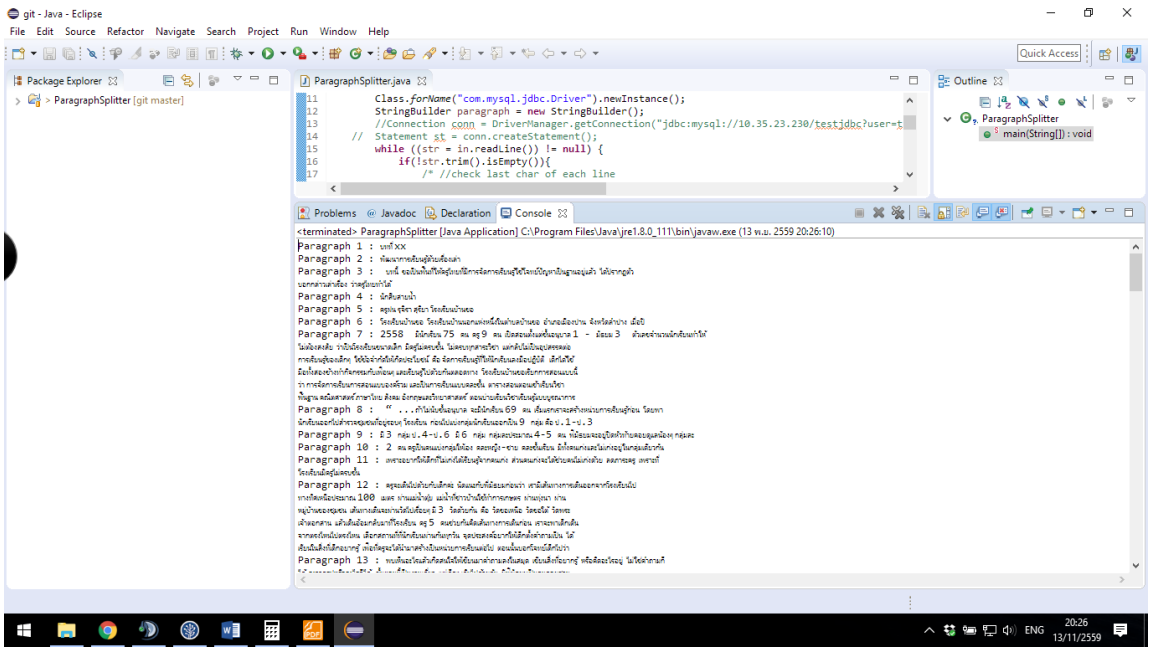
ภาพตัวอย่างของหน้าเว็บสำหรับการ Upload เอกสารเข้าไปในระบบ โดยหลังจากทำการ Upload จะมีการแบ่ง paragraph ในระบบและแสดงผลออกมาให้ใส่ tag ในหน้าสำหรับใส่ tag



ภาพที่ 4.3 ภาพตัวอย่างของหน้าเว็บในการ tag ข้อความแต่ละ paragraph ในไฟล์ PDF



ภาพที่ 4.4 ภาพตัวอย่างของไฟล์ PDF ที่จะทำการแบ่ง Paragraph



ภาพที่ 4.5 ภาพตัวอย่างของการแบ่ง paragraph จากเนื้อหาในไฟล์ PDF

4.2. อภิปรายผล

จากการที่เราได้ศึกษาการติดตั้งระบบ Hadoop Ecosystem ได้แก้ Spark ML, Impala และ Hbase สามารถติดตั้งได้อย่างไม่มีปัญหา และได้ทำการติดตั้ง Web Application เป็น Apache Server ซึ่งทำการเชื่อมต่อ VPN เข้ากับ Cluster ภายในมหาวิทยาลัย โดยตัว Web Application จะเขียนด้วยภาษา HTML, PHP และ Javascript

ในส่วนของ Web Application นั้น ได้มีการสร้างหน้าหลักสำหรับการให้ Expert สามารถ Upload เอกสารที่เป็น PDF และใส่ Tag โดยในขณะนี้ ได้อยู่ระหว่างขั้นตอนการเชื่อมต่อระหว่าง Web Application กับ Ubuntu Server เพื่อให้สามารถ execute java command สำหรับแยก text และนำไปแสดงผลที่ Web Application และจะมีการทำการเก็บข้อมูลลงไปใน Impala และ Hbase

ในส่วนของการแปลง PDF to Text นั้น ทางกลุ่มได้ทดสอบการใช้ PDFMiner และ PDFBox โดย PDFBox สามารถจำแนกบรรทัดและแบ่ง Line ได้ถูกต้องมากกว่า PDFMiner โดย PDFMiner นั้น สำหรับภาษาไทย จำเป็นต้องมีการแยก TTag ของ PDF เพื่อระบุลำดับตัวอักษรแล้วจึงนำมาแปลงทีหลัง จึงจะได้ความถูกต้องที่ดียิ่งขึ้น แต่ PDFBox นั้นสามารถดึง Text ธรรมดาออกมาได้มีความถูกต้องใกล้เคียงกว่า

ในส่วนของการทำ Paragraph Splitter นั้นจะมีปัญหาที่พบระหว่างการดำเนินงานของโครงการนี้ มีหลากหลายประการ เช่น การที่โครงสร้างของไฟล์ PDF ไม่มีการเก็บข้อมูลของการกด enter เพื่อเว้นบรรทัดแยกเอาไว้ ทำให้เวลาที่แปลงไฟล์เอกสารที่เป็น PDF มาเป็นไฟล์ข้อความนั้น จะพบปัญหาว่า ไฟล์ข้อความที่ได้มานั้น จะมองการกด enter เพื่อขึ้นย่อหน้าใหม่เป็นเพียงแค่ spacebar อันหนึ่งเท่านั้น ไม่ใช่อักขระพิเศษที่

เป็นตัว enter หรือ new line ทำให้เวลาที่ต้องการจะแบ่ง paragraph จะต้องใช้วิธีการตรวจหา spacebar ที่ท้ายบรรทัดแทน ซึ่งบางทีจะพบปัญหาว่า ในบรรทัดที่มีการพิมพ์แล้วกด space bar แล้วโปรแกรม word processing ทำการขึ้นบรรทัดใหม่ให้พอดี ก็จะมี space bar เป็นอักขระสุดท้ายของบรรทัดเช่นกัน ดังนั้น การแบ่ง paragraph จังยังมีการแบ่งแบบผิดๆ ถูกๆ อยู่บ้าง และสาเหตุที่ทางกลุ่มไม่สามารถใช้ตัว tab ที่ข้างหน้าบรรทัดเพื่อแบ่ง paragraph ได้นั้น เนื่องจากมีเอกสารที่ได้รับมาเป็นตัวอย่างอยู่หลายฉบับที่ไม่มีการกด tab เพื่อขึ้นย่อหน้าใหม่ มีเพียงแค่การกด enter ลงมาเพื่อการขึ้นบรรทัดใหม่เท่านั้น ทำให้ทางกลุ่มตัดสินใจที่จะไม่ใช้วิธีการค้นหาตัว tab ที่ตัวอักษรตัวแรกของบรรทัด

ปัญหาอีกประการที่พบก็คือ ปัญหาเรื่องการวางตำแหน่งอักขระของภาษาไทย ซึ่งจะมีอักขระอยู่ตัวหนึ่งที่มีตัวอักขระเขียนอยู่หลายตำแหน่งในตัวเดียว คือ “สระอำ” ทำให้โปรแกรมแปลงไฟล์ PDF เป็นไฟล์ข้อความนั้นอ่านสระอำออกมาตามปกติไม่ได้ และส่งผลให้สระอำถูกแปลงออกมาในรูปแบบของ Space bar ตามด้วยสระอาแทน ดังนั้น ทางกลุ่มจึงเขียนโปรแกรมภาษาจาวาง่ายๆ เพื่อตรวจหารูปแบบตัวอักษรในแบบที่กล่าวไว้ข้างต้น และแปลงเป็นสระอำเสียก่อนที่จะทำการทำกระบวนการอื่นๆ ต่อไป

4.3 สถานะการดำเนินงาน

การทำ PDF to Text	√
ติดตั้ง Impala, Spark, Web Server	√
สร้าง Web Application สำหรับ รับ document และ tag	√
การทำ paragraph splitter	√
การทำ Preprocess Data ข้อมูลภาษาไทย	
หา document และ tag สำหรับ train model	
สร้าง model โดยใช้ Classification	
สร้าง Web Application สำหรับ Search Tag	

บรรณานุกรม

[1] World Economic Forum. **Thailand Report** [Online]**.** Available: http://www3.weforum .org/docs/GCR2014-15/THA.pdf [2016, October 18]

[2] Ce Zhang.Ph.D. Dissertation, University of Wisconsin-Madison. **DeepDive: A Data Management System for Automatic Knowledge Base Construction** [Online]. Available: http://cs.stanford.edu/people/czhang/zhang.thesis.pdf [2016, October 18]

[3] **AlchemyLanguageAPI** [Online]. Available: https://alchemy-language-demo.mybluemix.net/. [2016, October 18]

[4] **AYLIEN** [Online]. Available: http://aylien.com/. [2016, October 18]

[5] Blei, D. M., Ng, A. Y. and Jordan, M. I. In: Journal of Machine Learning Research 3, pp. 993-1022. **Latent Dirichlet allocation** [Online]. Available: http://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf [2016, October 18]

[6] **Latent Semantic Analysis of Wikipedia with Spark** [Online]. Available: http://www.slideshare.net/SandyRyza/lsa-47411625. [2016, October 18]

[7] Costin Chiru, Traian Rebedea and Silvia Ciotec. **Comparison between LSA-LDA-Lexical Chains** [Online]. Available: http://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf [2016, October 18]

[8] Rich Caruana, Alexandru Niculescu-Mizil. **An Empirical Comparison of Supervised Learning Algorithms** [Online]. Avaliable: https://www.scribd.com/document/113006633/2006-An-Empirical-Comparison-of-Supervised-Learning-Algorithms [2016, October 18]

[9] Graham Neubig, Nara Institute of Science and Technology (NAIST). **Word Segmentation** [Online]. Avaliable: http://www.phontron.com/slides/nlp-programming-en-03-ws.pdf [2016, October 18]

[10] Hady Elsahar. **Word Embedding: Why the hype** [Online]. Avaliable: http://www.slideshare.net/hadyelsahar/word-embedings-why-the-hype-55769273 [2016, November 15]

[11] Cambridge University Press. **Tf-idf weighting** [Online]. Avaliable: http://nlp.stanford.edu/IR-book/html/htmledition/tf-idf-weighting-1.html [2016, November 15]

[12] ประหยัด สุพะกำ. **Artificial Neural Network** [Online]. Avaliable: http://alaska.reru.ac.th/text/ann.pdf [2016, November 15]

[13] The Apache Software Foundation. **Apache Hadoop** [Online]. Avaliable: http://hadoop.apache.org/ [2016, October 18]

[14] The Apache Software Foundation. **MLlib | Apache Spark** [Online]. Avaliable: http://spark.apache.org/mllib/ [2016, October 18]

[15] The Apache Software Foundation. **Apache Impala** [Online]. Avaliable: https://impala.apache.org/ [2016, October 18]

[16] The Apache Software Foundation. **Apache HBase** [Online]. Avaliable: http://hbase.apache.org/ [2016, October 18]

[17] The Apache Software Foundation. **PDFBox** [Online]. Avaliable: https://pdfbox.apache.org [2016, November 14]