2.1 สาระสำคัญ

เนื่องจากในปัจจุบัน เอกสารความรู้ต่างๆ มากมาย ยังไม่ถูกนำมาจัดเก็บและจัดหมวดหมู่ให้เป็นระบบ ทำให้องค์ความรู้เหล่านั้นไม่สามารถนำมาใช้งานได้อย่างมีประสิทธิภาพ โดยในปัจจุบันนั้น การที่จะนำความรู้ที่มีอยู่เหล่านั้นมาจัดหมวดหมู่เพื่อใช้ในการสืบค้น จะต้องใช้มนุษย์เป็นผู้จัดการ ซึ่งจำเป็นต้องใช้แรงงานคนในการมาจำแนกหมวดหมู่เหล่านั้น โดยการพัฒนาการจัดการองค์ความรู้หรือ Knowledge management นี้ จะช่วยให้การนำความรู้ที่มีอยู่มาใช้ได้อย่างมีประสิทธิภาพมากขึ้น และช่วยส่งเสริมการพัฒนาประเทศชาติให้มุ่งไปสู่ความเป็นสังคมอุดมปัญญาได้ โดยที่ระบบ Knowledge management ที่มีอยู่ในปัจจุบันนั้น ยังไม่มีระบบที่สามารถนำเอาเอกสารไปทำการวิเคราะห์และ tag หมวดหมู่ของเนื้อหาได้โดยอัตโนมัติ โดยเฉพาะอย่างยิ่งในส่วนของภาษาไทย ซึ่งยังไม่มีนักพัฒนารายใดพัฒนาเทคโนโลยีในลักษณะนี้ออกมา ดังนั้น เราจึงคิดที่จะทำระบบที่สามารถนำเอาเอกสารหรือบทความต่างๆที่ถูกจัดเก็บไว้ในรูปของ PDF นำไปวิเคราะห์ข้อความและทำการสรุปว่า ข้อความนี้มีเนื้อหาที่เกี่ยวข้องกับเรื่องใดบ้าง และนำไปจัดเก็บลงไปยังฐานข้อมูล เพื่อให้สามารถทำการสืบค้นได้ โดยสิ่งที่ท้าทายสำหรับการทำโครงงานชิ้นนี้ก็คือ การที่ภาษาไทยไม่มีการแบ่งคำที่ที่ชัดเจนเหมือนภาษาอังกฤษที่มีการใช้ space คั่น ทำให้การวิเคราะห์รูปประโยคมีความยาก, การแบ่ง paragraph ต่างๆในรูปแบบ PDF file และการทำ machine learning ให้ได้ความแม่นยำในระดับที่สามารถนำไปใช้ได้จริงนั้น จะต้องใช้การเลือกใช้ algorithm และการปรับแต่งที่เหมาะสมกับข้อมูลที่นำมาใช้ จึงทำให้โครงการนี้มีความท้าทายในการดำเนินการ และเป้าหมายในการทำโครงการนี้ จะเป็นการพัฒนา model ที่สามารถนำข้อความจากเอกสารมา tag และทำการจัดหมวดหมู่ได้ และพัฒนา web application ที่สามารถสืบค้นข้อมูลที่ได้จาก model ข้างต้น โดย web application ที่จะสร้างขึ้นนั้น จะเป็น web application สำหรับสืบหาข้อมูลที่เกี่ยวข้องกับ การพัฒนาการเรียนการสอน ซึ่งจะสามารถทำการสืบค้นได้ด้วยการใส่ tag ที่ต้องการ และ web application จะให้ผลลัพธ์ออกมาเป็น ย่อหน้าที่เกี่ยวข้องกับ tag ที่เราทำการค้นหาพร้อมทั้งแนบลิ้งสำหรับ download เอกสาร

2.2 คำสำคัญ

Machine Learning, Clustering, Latent Semantic Analysis, Classification, One vs Rest, Big data, Spark, Impala, PDF to Text, Keyword extraction, Text Analysis, Tag Box Extraction from PDF

3. หลักการและเหตุผล

ในยุคปัจจุบัน ที่มีการทำเอกสารในเรื่องต่างๆ ออกมาเป็นจำนวนมาก การทำ Knowledge management หรือการนำเอกสารข้อมูลเหล่านั้นมาจัดการให้เป็นระบบ นับเป็นเรื่องที่สำคัญมาก โดยเฉพาะในองค์กรหลายๆแห่ง การมีระบบ knowledge management จะช่วยทำให้องค์กรนั้นๆ สามารถบริหารจัดการงานองค์ความรู้ที่มีอยู่ได้อย่างมีประสิทธิภาพสูงสุด แต่ในปัจจุบัน เอกสารความรู้ต่างๆ ที่ถูกนำมาเผยแพร่อยู่นั้น มักจะอยู่ในรูปแบบของเอกสารในหน้ากระดาษ หรือเอกสารที่เป็นไฟล์ PDF ซึ่งยังไม่สามารถนำมาใช้งานได้อย่างมีประสิทธิภาพเท่าที่ควร เพราะว่า เอกสารเหล่านั้นมักจะมีข้อความอยู่มากมาย ที่เกี่ยวข้องกับเนื้อหาที่แตกต่างกัน แต่ว่าเมื่อผู้ที่ต้องการใช้งานความรู้เหล่านั้น ต้องการทำการหาเนื้อหาที่เฉพาะเจาะจงกับที่เขาสนใจในเอกสารนั้นๆ เขาก็ต้องทำการค้นหาด้วยตนเองโดยวิธีต่างๆ ไม่ว่าจะเป็นการไล่อ่านเนื้อหาทั้งหมดด้วยตนเอง ซึ่งใช้เวลามากในการอ่านและหาใจความสำคัญที่เขาต้องการ หรือใช้การค้นหา keyword ที่เขาต้องการด้วยวิธีต่างๆ เช่นการเปิดหาสารบัญ ซึ่งเอกสารบางฉบับก็ไม่มีสารบัญให้ หรือใช้การ search หา keyword ที่ต้องการ ซึ่งอาจจะเกิดการข้ามเนื้อหาในส่วนที่เกี่ยวข้องกับเรื่องที่ผู้ที่ค้นหาต้องการ แต่ไม่มี keyword ที่เขาใช้ค้นหาไปได้ ซึ่งสิ่งที่ได้กล่าวไปข้างต้นนั้น นับว่าเป็นปัญหาใหญ่ในการค้นคว้าหาข้อมูลเพื่อทำการศึกษาเป็นอย่างมาก เนื่องจากการที่ไม่มีระบบ knowledge management สำหรับเอกสารทั่วๆ ไปนั้น ทำให้แหล่งความรู้ที่สามารถนำมาสืบค้นได้นั้นลดลงเป็นอย่างมาก และทำให้ความรู้จำนวนมากถูกทิ้งร้างไว้ไม่ได้ถูกนำมาใช้ให้เกิดประโยชน์ ดังนั้น ทางกลุ่มของเราจึงสนใจที่จะพัฒนา machine learning model ที่สามารถคัดแยกเนื้อหาในส่วนต่างๆ ในไฟล์เอกสาร และทำการ tag ข้อความเหล่านั้นได้โดยอัตโนมัติว่า เนื้อหาในส่วนนั้นๆ มีความเกี่ยวข้องกับเรื่องอะไรบ้าง และทำการจัดเก็บข้อมูลเหล่านั้นลงไปยังระบบฐานข้อมูลเพื่อให้สามารถทำการสืบค้นได้ง่ายและรวดเร็ว และทำให้การจัดการแหล่งความรู้ หรือ Knowledge management นั้น สามารถใช้งานกับเอกสารที่เป็นไฟล์ PDF ได้ ซึ่งส่งผลให้ความรู้ถูกนำไปใช้งานต่อ และเกิดการพัฒนาประเทศชาติในองค์รวมมากยิ่งขึ้น

4. วัตถุประสงค์

1. ศึกษาการทำ Text Processing ของภาษาไทย ได้แก่การตัดคำและการระบุประเภทของคำ

2. สร้าง Machine Learning Model สำหรับเรียนรู้เอกสารภาษาไทยและ tag ที่กำหนดเพื่อใช้สำหรับการประมวลผลออกมาเป็น Paragraph ที่สำคัญและ Tag ที่เกี่ยวข้องจากเอกสารใดๆในขอบเขตที่เกี่ยวข้อง

3. สร้าง Web Application สำหรับค้นหา Tag ที่สนใจ และแสดงผลลัพธ์ออกมาเป็น Paragraph และ Tag ที่เกี่ยวข้องพร้อมเอกสารฉบับสมบูรณ์ในรูปแบบ PDF

5. ปัญหาหรือประโยชน์ที่เป็นเหตุผลให้ควรพัฒนาโปรแกรม

ในปัจจุบันนั้น มีโปรแกรมที่ถูกพัฒนาโดยมหาวิทยาลัยสแตนฟอร์ด ที่ชื่อว่า Stanford Deepdive ที่ทำการนำข้อความ, ตารางหรือรูป (unstructured information) มาทำการวิเคราะห์เนื้อหาเหล่านั้นได้โดยการใช้ Machine Learning เพื่อให้ได้ผลลัพธ์ออกสร้างเป็นฐานข้อมูล SQL tables (structured information) เช่น GeoDeepDive ที่สามารถค้นหาข้อมูลทางธรณีวิทยาจากบทความทางวิชาการได้ แต่ว่าโปรแกรม Stanford Deepdive ที่กล่าวมานั้น ถูกพัฒนาขึ้นสำหรับการวิเคราะห์เอกสารในภาษาอังกฤษเป็นหลัก ซึ่งภาษาไทยที่มีรูปแบบของประโยค การจัดเรียงคำ การวางตำแหน่งคำ,ตัวอักษร และอื่นๆ ที่แตกต่างจากภาษาอังกฤษเป็นอย่างมาก ทำให้การทำระบบ tag เอกสารอัตโนมัติสำหรับภาษาไทยนั้น ไม่สามารถใช้ Stanford Deepdive ได้ และการ tag หมวดหมู่ให้กับข้อความจำนวนมากโดยใช้มนุษย์ในการจัดการนั้น จะเป็นการเสียเวลาไปเป็นจำนวนมาก ทำให้ทางกลุ่มของเราสนใจที่จะพัฒนาโปรแกรมในลักษณะคล้ายกันกับ Stanford Deepdive ที่สามารถนำมาใช้กับภาษาไทยได้ เพื่อให้เอกสารต่างๆ ที่เป็นภาษาไทยนั้น ถูกนำมาใช้ประโยชน์ และนำมาศึกษาต่อได้อย่างมีประสิทธิภาพ เช่น การสร้างฐานข้อมูลที่สามารถสืบค้นได้จากเอกสารที่เกี่ยวข้องกับการพัฒนาการเรียนการสอนของโรงเรียนต่างๆ เพื่อให้ครูสามารถค้นหาเอกสารที่มีความเกี่ยวข้องหรือแนวทางที่ตนสนใจเพื่อจะนำไปใช้ในการพัฒนาการเรียนการสอนของตนเองต่อไป

6. เป้าหมายและขอบเขตของโครงการ

1. สร้างโปรแกรมสำหรับรับ PDF ภาษาไทย เพื่อทำการระบุและแบ่ง paragraph ต่างๆจาก PDF ที่รับมาและจะทำการดึง text ภาษาไทยใน PDF ออกมาจาก paragraph เหล่านั้นออกมาเพื่อทำการ Text processing ต่อ เพื่อให้ได้ผลลัพธ์สุดท้ายเป็น word ที่สำคัญต่างๆ สำหรับนำไปใช้ในการทำ Machine Learning

2. ทำการศึกษาและพัฒนา machine learning model ที่สามารถรับข้อมูล text file ภาษาไทยที่ได้จากขั้นตอนที่ 1 โดยวิธีการ supervised classification ซึ่งจะแบ่งเป็น 2 ขั้นตอนคือ

2.1 การทำ training model จะรับ text และ tag ของ paragraph

2.2 การทำ prediction จาก model โดยจะรับ text เป็น paragraph และแสดง tag เป็นผลลัพธ์และทำการเก็บลงใน Database

3. ทำการพัฒนา web application เพื่อที่จะใช้ในการสืบค้นข้อมูลที่ทำการ tag มาแล้วจากขั้นตอน prediction จากใน Database

รายละเอียดของการพัฒนา

7.1 Story board

เมื่อต้องการที่จะค้นหาเอกสารที่เกี่ยวข้องกับเรื่องใดเรื่องหนึ่งขึ้นมาใช้งาน ผู้ใช้จะสามารถค้นหาข้อมูลได้อย่างรวดเร็วด้วย platform ที่ทางกลุ่มพัฒนาขึ้น โดย platform ที่พัฒนาขึ้นจะแบ่งออกได้เป็น 2 ส่วนหลักๆ

- ส่วนของผู้พัฒนาและผู้ดูแลระบบ โดยผู้พัฒนาจะทำการเตรียมเอกสารที่เป็นไฟล์ PDF ตัวอย่างซึ่งเอกสารเหล่านี้จะมีผู้เชี่ยวชาญเฉพาะมาช่วยในการระบุคำสำคัญต่างๆเพื่อทำการเตรียม machine learning model โดยหลังจากที่ทำการสร้าง model เสร็จแล้ว เอกสารที่เหลือจะทำการ tag เอกสารได้โดยอัตโนมัติ โดยใช้ machine learning model ข้างต้น เช่น ถ้าต้องการให้ตัว model สามารถทำการจำแนกเนื้อหาที่เกี่ยวข้องกับเรื่อง “การดึงความสนใจนักเรียน” ผู้พัฒนา/ผู้ดูแลจะต้องเตรียมเอกสารที่มีเนื้อหาที่เกี่ยวข้องกับเรื่องการดึงความสนใจของนักเรียนไว้ และให้ผู้เชี่ยวชาญช่วยระบุว่า มีคำใดบ้างที่สามารถระบุได้ว่า ข้อความนี้มีความเกี่ยวข้องกับ "การดึงความสนใจของนักเรียน" และนำไปทำการเตรียม model โดยหลังจากสร้าง model เสร็จ

แล้ว ผู้พัฒนาสามารถนำเอกสารที่เกี่ยวข้องกับ "การดึงความสนใจนักเรียน" มาทำการ tag เอกสารโดยอัตโนมัติได้

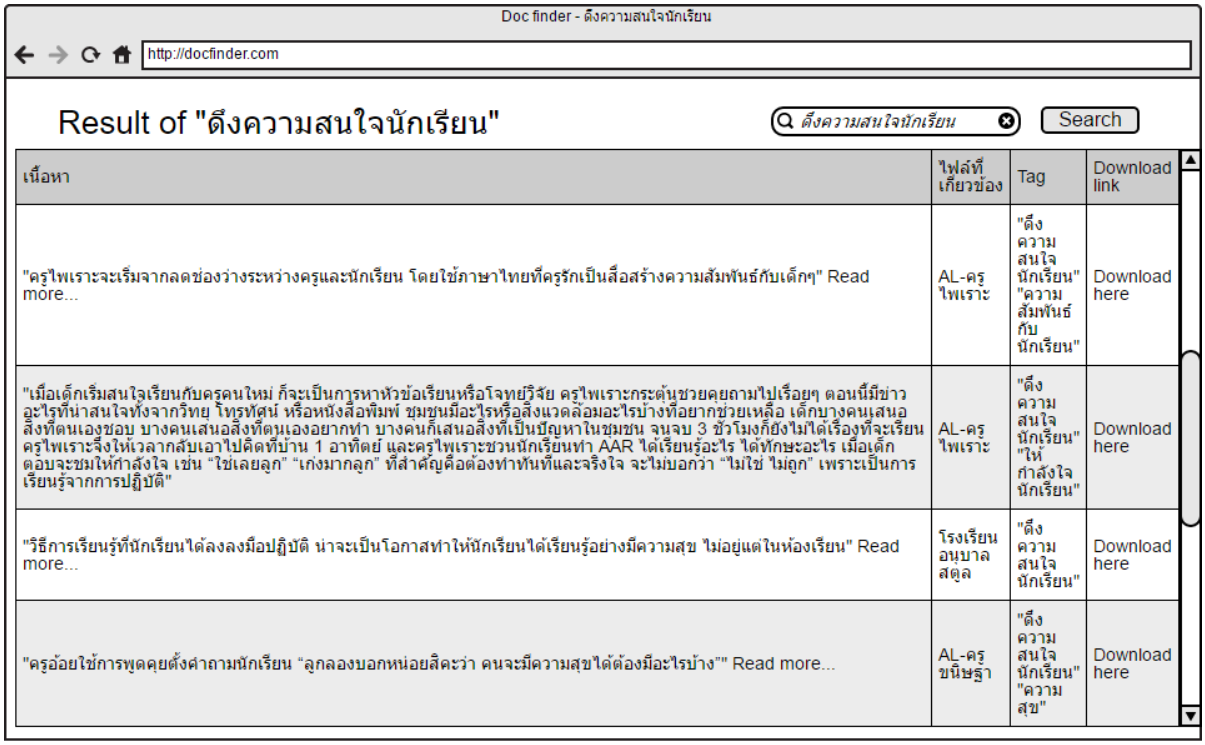
- ส่วนของผู้ใช้งาน ผู้ใช้สามารถเข้ามาใช้งานผ่าน web application ที่ทางกลุ่มพัฒนาขึ้นมา แล้วทำการค้นหาเนื้อหาที่เกี่ยวข้องกับสิ่งที่ผู้ใช้ต้องการ แล้วเนื้อหาส่วนนั้นก็จะปรากฏขึ้นมา และมีไฟล์เอกสารนั้นให้ผู้ใช้สามารถ download ไปอ่านได้ ยกตัวอย่างเช่น ครูสมศรีต้องการที่จะหาข้อมูลเรื่อง “การดึงความสนใจนักเรียน”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

โดย platform นี้จะมีจุดเด่นที่เราสามารถนำ model นี้ ไปประยุกต์ใช้กับหัวข้ออื่นๆได้ โดยไม่จำเป็นต้องออกแบบโปรแกรมใหม่ทั้งหมด เพียงแค่เตรียมเอกสารที่เกี่ยวข้องและกำหนดคำสำคัญของหัวข้อนั้นๆให้กับเอกสารตัวอย่างและให้ระบบทำการเรียนรู้ด้วยตัวเองกับเอกสารที่เหลือ

ในปัจจุบันนี้ มีโปรแกรมสำรหับการแปลง 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 และนำมาทำการวิเคราะห์ข้อมูลต่างๆ ได้ เช่น การนำบทความที่เขียนไว้และฐานข้อมูลมาสรุปผลร่วมกัน ซึ่งนอกจาก Deepdive แล้ว จะมีโปรแกรมสำหรับดึงข้อมูลข้อมูลจาก unstructured information ได้แก่ AlchemyLangage API 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 ที่เป็นโปรแกรมที่รับ text file และทำการตรวจสอบคำสำคัญ, สรุปของบทความหรือการสร้าง hashtag จาก model ของทางระบบที่สร้างไว้ ซึ่งโดยส่วนใหญ่ของโปรแกรมเหล่านี้ จะรองรับสำหรับภาษาในภาษาอังกฤษหรือภาษาที่รากศัพท์มาจากภาษาละติน เนื่องจากมี Library ในการจัดการทางภาษาศาสตร์จาก NLP Stanford

สำหรับการดึงข้อมูลต่างๆจากเอกสารนั้น จะแบ่งขั้นตอนต่างๆออกเป็น 3 ส่วน ได้แก่ 1.Preprocessing Data โดยในขั้นตอนนี้ สำหรับภาษาไทยนั้น จำเป็นต้องมีการตัดแบ่งคำ (Word Segmentation) สำหรับประโยคออกเพื่อทำการทำ NER (Name Entity Recognition) และทำการตัดคำต่างๆที่ไม่จำเป็นทิ้งไป ซึ่งสุดท้ายจะทำการทำ bag-of-word เพื่อนำไปต่อไป 2. Topic Discovery จะเป็นการ

ดึงคำสำคัญหรือความเกี่ยวข้องต่างๆที่อยู่ใน paragraph ออกมา เช่นการทำ TF-IDF เพื่อหาความถี่ของคำ และการลดมิติของคำให้เหลือเพียงคำสำคัญต่างๆโดยการใช้ Machine Learning: Clustering) เช่น Latent Dirichlet Allocation และ Latent semantic analysis 3. Machine Learning: Classification จะเป็นการสร้าง Machine สำหรับการจำแนกผลลัพธ์จากข้อมูลที่เข้ามา โดยจะแบ่งขั้นตอนการใช้งานเป็น 2 ขั้นตอน ได้แก่ 1.การทำ Training และ Testing Model โดยในขั้นตอนนี้จะนำคำต่างๆที่ได้จารขั้นตอนข้างต้นรวมกับ label ที่ผู้เชี่ยวชาญได้ระบุไว้มาสร้าง Model สำหรับการ Classification ออกมา 2. การ Prediction จากเอกสารต่างๆ เพื่อให้ได้ผลลัพธ์ออกมาเป็น tag เพื่อนำไปใช้ในการสืบค้นใน Database ต่อไป โดเทคนิคต่างๆของ Classification ได้แก่ One-vs-Rest, Neural Network, Decision Tree



7.2 เทคนิคหรือเทคโนโลยีที่ใช้

- Word Segmentation เป็นวิธีการในการแบ่งคำต่างๆออกจากประโยค โดยในภาษาไทยนั้น รูปแบบของประโยคจะเป็นคำต่อกันโดยไม่มีตัวระบุการจบคำหรือประโยค ทำให้จำเป็นจะต้องใช้โปรแกรมเฉพาะในการตัดคำ ซึ่งในปัจจุบันมีโปรแกรมสำหรับตัดคำภาษาไทยต่างๆได้แก่ LexTo เป็นโปรแกรมในการจัดคำที่จะใช้วิธี Dictionary base ในการที่จะเลือกแบ่งคำจากประโยค และ TLex เป็นโปรแกรมในการตัดคำภาษาไทยโดยใช้ machine learning ชื่อว่า Condition Random Fields
- bag-of-words model จะเป็นโมเดลในการทำ mapping ของคำต่างๆให้กลายเป็นตัวเลข เพื่อที่จะสามารถนำไปใช้ประโยชน์ในเชิงคณิตศาสตร์และการทำสถิติต่างๆต่อไป

- Term frequency – Inverse document frequency (TF-IDF) เป็นวิธีทางสถิติที่จะทำการตรวจสอบคำต่างๆในบทความเพื่อนำไปเปรียบเทียบกับบทความทั้งหมด เพื่อหาอัตราส่วนว่าคำๆนี้มีความสำคัญต่อบทความโดยรวมแค่ไหน โดย TF-IDF จะแบ่งขั้นตอนเป็น 2 ส่วนคือ Term frequency โดยในขั้นตอนนี้นั้นจะทำการนับจำนวนครั้งที่คำต่างๆปรากฏในบทความหนึ่งๆ และการทำ Inverse document frequency โดยในขั้นตอนนี้จะเป็นนำคำต่างๆในบทความมาเปรียบเทียบกับบทความทั้งหมดและคำนวณหาค่าน้ำหนักความสำคัญนั้นๆ จากบทความทั้งหมด โดยการทำ TF-IDF นั้นสามารถใช้ประโยชน์ในการหาคำสำคัญในบทความต่างๆซึ่งสามารถนำไปประยุคใช้ได้อย่างหลากหลายเช่นการทำ Search engine หรือการทำ Text Summarization
- Latent Dirichlet Allocation เป็น clustering algorithm ที่ใช้สำหรับการทำ topic discovery จากข้อมูลต่างๆ ที่ใส่เข้าไป โดยขั้นตอนการทำงานจะเป็นการตรวจสอบเนื้อหาในประโยคจต่างๆ แล้วนำไปจัดหมวดหมู่ตามจำนวน topic ที่ได้ตั้งไว้ โดยจะอ้างอิงตามหลักความน่าจะเป็นว่าประโยคต่างๆมีความเกี่ยวข้องกับ topic ใดมากที่สุด
- Latent semantic analysis เป็นหนึ่งในวิธีการทำ keyword extraction จากเอกสารโดยจะทำการเปลี่ยนคำต่างๆ ให้กลายเป็น vector แล้วทำการสร้าง matrix สำหรับเก็บจำนวน frequency ของคำ และใช้วิธีการทางคณิตศาสตร์เรียกว่า Singular value decomposition (SVD) ในการลดจำนวนมิติของ array ใน matrix เพื่อหาค่าที่มีความเกี่ยวข้องมากที่สุดจากในเอกสารนั้นๆ
- One-vs-Rest เป็น classification algorithm รูปแบบหนึ่งที่สามารถให้ผลลัพธ์การจัดกลุ่มได้หลายผลลัพธ์ (multiclass classification) โดยจะใช้วิธีการจัดกลุ่มข้อมูลนั้นๆ เข้าในแต่ละหมวดหมู่แล้วทำการเปรียบเทียบผลกับหมวดหมู่อื่นๆ แล้วเลือกผลลัพธ์ที่ทำให้การจัดกลุ่มมีความแม่นยำสูงที่สุด โดยการจัดกลุ่มอันนี้จะทำให้ข้อมูล 1 ย่อหน้าสามารถมี tag ได้หลายอย่าง
- Neural network เป็น classification algorithm ที่เลียนแบบการทำงานของระบบประสาทของมนุษย์ โดยมีการส่งข้อมูลที่ทำการเรียนรู้อยู่ในระบบเข้าสู่ node ต่างๆ และหาค่าน้ำหนักในแต่ละ node แล้วทำการส่งข้อมูลไปยัง node ย่อยๆ ต่างๆ ไปเรื่อยๆ จนได้ผลลัพธ์การจัดกลุ่มที่ดีที่สุด ซึ่งการทำ neural network นี้จะนำมาใช้เป็น 1 ใน algorithm ที่จะนำมาใช้ในการ tag ข้อมูลอัตโนมัติ
- Decision Tree เป็น classification algorithm ที่เป็น rule-based classification คือการสร้างต้นไม้ของกฎต่างๆ เพื่อที่จะจัดกลุ่มข้อมูล โดยต้นไม้จะมีความซับซ้อนเพิ่มมากขึ้นเรื่อยๆ ตามความซับซ้อนของข้อมูลที่เรียนรู้ โดย decision tree จะเป็นอีก 1 algorithm ที่จะใช้ในการ tag ข้อมูลอัตโนมัติ

7.3 เครื่องมือที่ใช้ในการพัฒนา

- Hadoop Distributed File System (HDFS) 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
- โปรแกรม Spark ML เป็น library ที่มีอยู่ในโปรแกรม Apache Spark ซึ่ง Spark ML เป็น library ที่ใช้ทำ 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
- โปรแกรม Apache Impala 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
- โปรแกรม Apache HBase เป็นโปรแกรมจัดการฐานข้อมูลแบบ NoSQL ที่เป็น Open source ที่ถูกใช้งานร่วมกัน Hadoop ecosystem โดยฐานข้อมูลแบบ NoSQL จะมีความยืดหยุ่นด้านโครงสร้างมากกว่าฐานข้อมูลแบบ SQL ดังนั้น ทางกลุ่มจึงนำ HBase มาใช้งานร่วมกันกับ Impala เพื่อเก็บข้อมูลที่เหมาะสมลงในฐานข้อมูลแต่ละโปรแกรม โดย HBase จะเก็บข้อมูลจำพวกเนื้อหาของแต่ละเอกสารที่ถูกแบ่งย่อหน้าแล้ว ซึ่งจำนวนย่อหน้าของแต่ละเอกสารจะมีไม่เท่ากัน ดังนั้นฐานข้อมูลแบบ NoSQL จึงเหมาะสมกับการเก็บข้อฒูลลักษณะนี้ ส่วน Impala ที่เป็นฐานข้อมูล

แบบ SQL จะจัดเก็บข้อมูลเรื่อง tag ของแต่ละย่อหน้าไว้ เพื่อให้สามารถทำการ Query ผ่านหน้าเว็บไซต์ได้อย่างรวดเร็ว

- โปรแกรมสำหรับการแปลงไฟล์ในรูปแบบ PDF ให้เป็น text file โดยทางกลุ่มเลือกใช้ PDFMiner ซึ่ง PDFMiner เป็น Python API ที่ใช้สำหรับการดึงข้อมูลต่างๆออกมาจาก PDF Document เช่น ตัวอักษรในภาษาต่างๆ เช่น ไทย อังกฤษ จีน และอื่นๆ หรือสามารถดึงภาพออกจาก PDF ได้ โดยสำหรับโปรเจคนี้จะเน้นที่การดึงข้อความออกจาก PDF Document เพื่อสำหรับนำไป preprocess ต่อ ซึ่งฟังก์ชั่นที่ใช้ในการดึงข้อความออกมานั้นคือ PDF2TXT โดยคำสั่งต่างๆของฟังชั่นนี้ สามารถเลือก page number, ชนิดของ output (text,tag,xml), ขนาดของ box ของคำใน pdf เป็นต้น โดยเหตุผลที่เราเลือกใช้ PDFMiner คือ การที่ตัวโปรแกรมสามารถ config ค่าต่างๆ ได้อย่างหลากหลาย ทำให้ผลการแปลงข้อมูลมีความแม่นยำมากยิ่งขึ้น เนื่องจากภาษาไทยเป็นภาษาที่มีโครงสร้างซับซ้อนกว่าภาษาอังกฤษมาก ทำให้ต้องมีการดัดแปลงตัวโปรแกรมเพื่อให้สามารถทำงานได้อย่างถูกต้อง
- โปรแกรม LexTo เป็นโปรแกรมที่ถูกพัฒนาด้วยภาษา Java โดยโปรแกรมนี้สามารถใช้ในการแบ่งคำต่างๆในภาษาไทยจากประโยคให้กลายเป็นคำซึ่งแบ่งด้วย delimiter ซึ่งคำต่างๆที่ใช้ในการแบ่งนั้น จะมี Dictionary ที่จะทำการเก็บคำทั้งหมดเอาไว้ แล้วโปรแกรมจะนำมาเปรียบเทียบเพื่อแบ่งคำตามที่ Dictionary ได้กำหนดไว้ ซึ่ง LexTo เป็นโปรแกรมตัดคำภาษาไทยแบบ Open Source ที่ทางกลุ่มสามารถนำมาใช้งานได้ และมีความแม่นยำในระดับที่พอรับได้ ทำให้ทางกลุ่มเลือกใช้โปรแกรม LexTo

7.4 รายละเอียดโปรแกรมที่จะพัฒนา

Input / Output Specification

Ingestion Web Application

- Input - PDF file อย่างเดียว หรือ PDF file Label จาก Expert สำหรับนำไปใช้ Training Model
- Output - Notification ว่ามีการรับ Input File สำเร็จแล้ว

Train Model

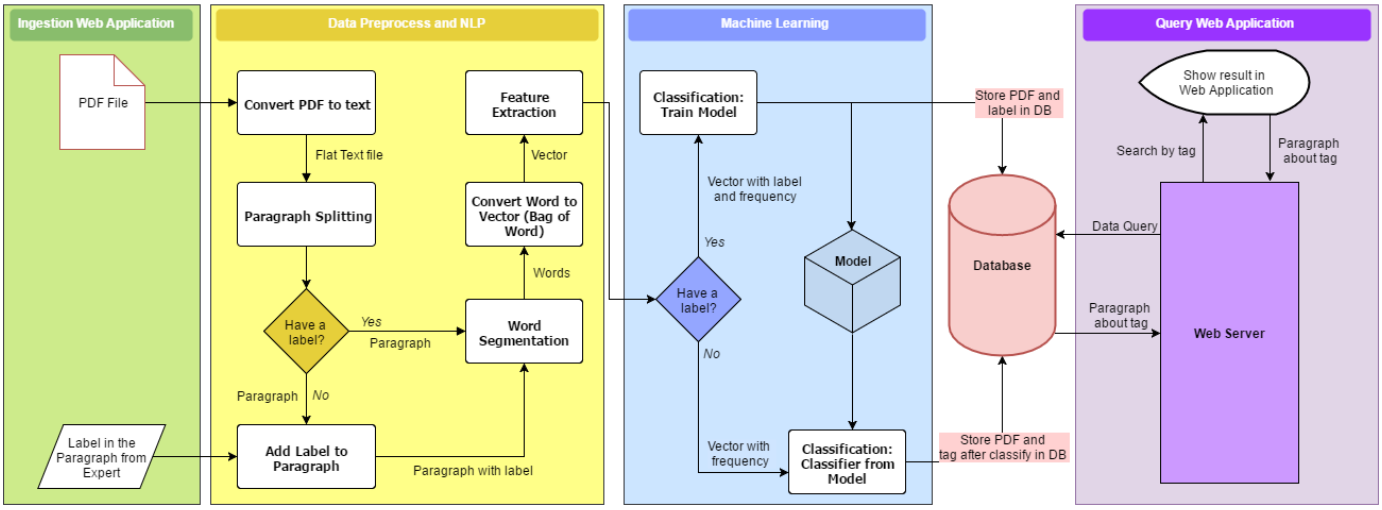
- Input – PDF File ที่มี Label จาก Expert
- Output - Model ที่สามารถคาดเดา tag จาก paragraph และเก็บข้อมูล PDF และ Tag ลงใน Database

Prediction

- Input - PDF File
- Output - เก็บข้อมูล PDF และ Tag ลงใน Database

Query Web Application

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



Functional Specification

Ingestion Web Application

- Upload PDF File มาเก็บไว้ในฐานข้อมูลและทำการส่งไปยังขั้นตอน Data Preprocess and NLP
- สามารถเลือกที่จะกรอก label ของ paragraph ต่างๆ สำหรับนำไปทำการ Training

Data Preprocess and NLP

- ทำการแปลง PDF ที่รับมาเปลี่ยนให้กลายเป็น xml file โดยการใช้ PDFMiner
- ทำการ preprocess xml file ที่ได้ โดยจะทำการระบุย่อหน้าจาก xml schema, ทำการตัด xml schema ต่างๆและดึงคำภาษาไทยออกมาและทำการแก้ภาษาไทยที่ผิดพลาดต่างๆจาก xml เช่น สระ อำ (อ า เป็น อำ)
- ทำการใส่ label ของคำเพิ่มเข้าไปใน paragraph สำหรับใช้เพื่อระบุในขั้นตอนต่อไป
- ทำการตัดคำจาก text file ด้วยโปรแกรม word segmentation สำหรับภาษาไทย เช่น LexTo
- ทำการกำจัดคำที่ไม่จำเป็นออกโดยการใช้ stop word remover
- สร้าง bag-of-word model สำหรับเก็บคำภาษาไทย และแปลงคำต่างๆจากในย่อหน้าให้กลายเป็น vector
- ใช้หลักการ TF-IDF ในการหาความถี่ของคำต่างๆ
- ใช้หลักการ LDA หรือ LSA กับ paragraph เพื่อช่วยลดมิติของคำ และนำคำที่ได้ไปรวมกับคำที่ label เพื่อกำหนดค่าที่ใช้สำหรับการทำ Classification

Classification: Training model

- ทำการ Train Model ด้วย paragraph, label คำจาก expert และ keyword จากการทำ LDA หรือ LSA ด้วยวิธีการ Classification โดยทดสอบกับ Technique ต่างๆ ได้แก่ neural network, One vs Rest, Decision Tree เป็นต้น
- นำ label, paragraph และ keyword เก็บลงใน Database

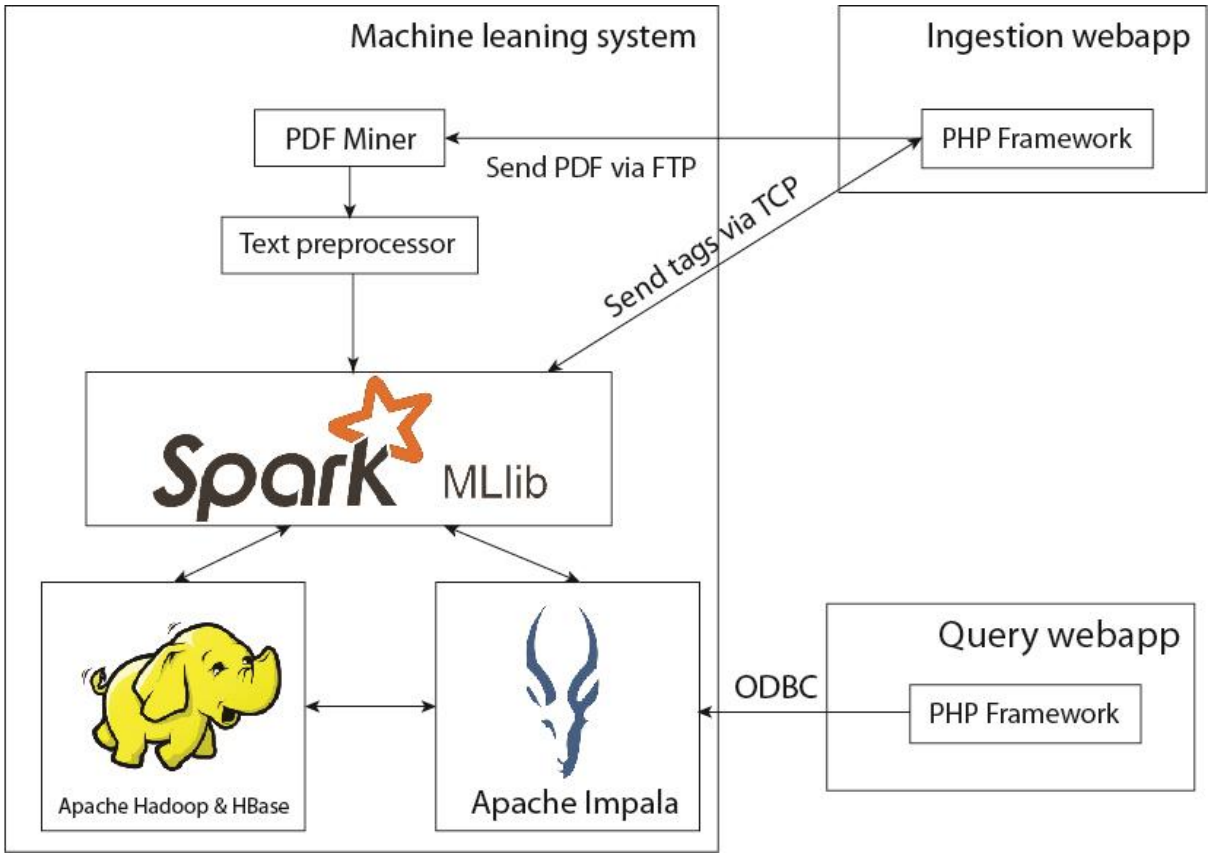
Classification: Prediction

- นำ keyword ของคำที่ได้จากการทำ LDA หรือ LSA มาทำการ Classify ผ่าน Model ที่ได้ทำมาในขั้นตอน Training Model เพื่อหา Tag ของเอกสารเหล่านั้น
- นำ tag ที่ได้จากการ Prediction, paragraph และ keyword ไปเก็บลงใน Database

Query Web Application

- รับคำที่ต้องการจะทำการค้นหาและทำการค้นหาคำในระบบ Database
- แสดงผลคำที่ต้องการค้นหา, paragraph ที่เกี่ยวข้อง, tag ที่เกี่ยวข้องกับ paragraph นั้นและ PDF file ที่สมบูรณ์

Architecture



7.5 ขอบเขตและข้อจำกัดของโปรแกรมที่พัฒนา

- เนื่องจากภาษาไทยเป็นภาษาที่มีความซับซ้อนสูง ทั้งทางด้านตัวอักษร ที่มีสระบน-ล่าง และทางด้านรูปประโยคที่ไท่ทีความแน่นอน ทำให้การเขียนโปรแกรมที่สามารถประมวลผลภาษาไทยได้อย่างสมบูรณ์แบบจึงเป็นเรื่องยาก ทำให้ความแม่นยำในการ tag และเลือกย่อหน้าที่มีความสำคัญกับเรื่องที่เลือก อาจจะต่ำกว่าการใช้งานกับภาษาอังกฤษ ที่มีรูปประโยคที่แน่นอนกว่า ทำให้สามารถใช้การดูรูปประโยคเข้ามาช่วยเสริมความหมายของคำได้ ซึ่งเป็นสิ่งที่ทำได้ยากมากในภาษาไทย
- ข้อมูลที่จะนำไปเข้าระบบ machine learning เพื่อให้ระบบทำการเรียนรู้ด้วยตนเองนั้น จะต้องใช้มนุษย์เป็นตัวช่วยในการกำหนดข้อมูลก่อนในเบื้องต้น เพราะฉะนั้น ถ้าเราต้องการให้ระบบเรียนรู้เนื้อหาเรื่องใหม่ๆ จะต้องมีการใช้ผู้เชี่ยวชาญที่เกี่ยวข้องกับเรื่องที่จะให้ระบบเรียนรู้มาช่วยทำการ label คำสำคัญก่อนที่จะนำข้อมูลเข้าไปในระบบ ดังนั้น ถ้าเกิดเราไม่สามารถหาผู้ที่จะมาระบุคำสำคัญให้ได้ เราก็จะไม่สามารถทำให้ระบบเรียนรู้หัวข้อใหม่ๆ ได้

- การระบุย่อหน้าจาก PDF นั้นสามารถทำได้ยากเนื่องจากการจะระบุย่อหน้าจาก PDF จำเป็นต้องใช้ค่าตำแหน่งของตัวอักษรต่างๆ เพื่อนำระบุว่าย่อหน้าควรจะอยู่ตำแหน่งไหน ซึ่ง PDF ที่ได้รับมานั้น มีรูปแบบการจัดหน้าและ font ที่แตกต่างกันรวมถึงรูปแบบคำภาษาไทยและภาษาอังกฤษในเอกสาร จะทำให้ตำแหน่งของคำเกิดการคลาดเคลื่อนซึ่งจะส่งผลให้ย่อหน้าที่ได้ออกมาอาจเกิดความผิดพลาดได้

บรรณานุกรม

- มาเริ่มเรียนรู้ Hadoop กันหน่อย, http://www.somkiat.cc/start-with-hadoop/ (Accessed 2016-9-23)
- Apache Spark, http://spark.apache.org (Accessed 2016-9-23)
- Pdfminer, http://euske.github.io/pdfminer/index.html
- PDFMiner, http://www.unixuser.org/~euske/python/pdfminer/
- One-vs-Rest classifier, https://spark.apache.org/docs/latest/ml-classification-regression.html#one-vs-rest-classifier-aka-one-vs-all
- Lexto , http://www.sansarn.com/lexto/
- Latent Dirichlet allocation(LDA), https://spark.apache.org/docs/1.6.0/ml-clustering.html#latent-dirichlet-allocation-lda
- Impala, http://impala.apache.org/
- Multiclass and multilabel algorithms, http://scikit-learn.org/stable/modules/multiclass.html
- Hbase, http://hbase.apache.org/
- LSA, https://en.wikipedia.org/wiki/Latent_semantic_analysis