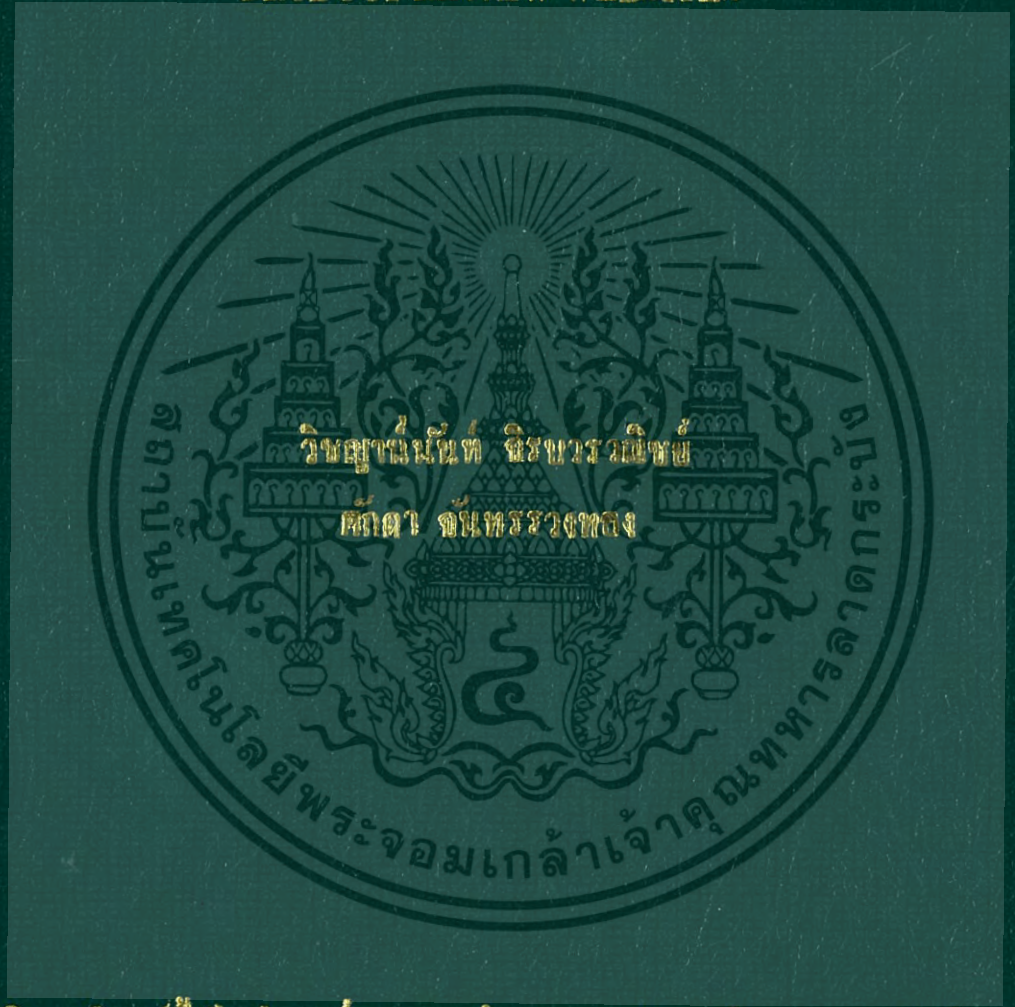
ระบบบริการส่วนต่อประสานโปรแกรมประยุกต์เพื่อสกัดคำที่บ่งบอกประเภทสินค้าจากเว็บไซต์วิจารณ์สินค้า

APPLICATION PROGRAMMING INTERFACE SERVICE SYSTEM FOR EXTRACTING BRANDS AND PRODUCT TYPES FROM PRODUCT REVIEW WEBSITES



วิชญานันท์ ฉิรพรวณิชย์

คัดตา ตันหรรวงทอง

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

สาขาวิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ภาคเรียนที่ 2 ปีการศึกษา 2558

ระบบบริการส่วนต่อประสานโปรแกรมประยุกต์เพื่อสังเคราะห์ชื่อและประเภทสินค้าจากเว็บไซต์วิจารณ์สินค้า

APPLICATION PROGRAMMING INTERFACE SERVICE SYSTEM FOR EXTRACTING BRANDS AND PRODUCT TYPES FROM PRODUCT REVIEW WEBSITES

โดย

วิชญาน์นันท์ จิรบวรวณิชย์

ศักดา จันทรรวงทอง

อาจารย์ที่ปรึกษา

ดร. กนกวรรณ อัจฉริยะชาญวณิช

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

สาขาวิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ภาคเรียนที่ 2 ปีการศึกษา 2558

APPLICATION PROGRAMMING INTERFACE SERVICE SYSTEM FOR EXTRACTING BRAND AND PRODUCT TYPE FROM PRODUCT REVIEW WEBSITES

VICHAYANAN JIRABOVOLVANIT

SAKDA JANTARAROUNGTHONG

A PROJECT SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

BACHELOR OF SCIENCE PROGRAM IN INFORMATION TECHNOLOGY

FACULTY OF INFORMATION TECHNOLOGY

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2/2015

ใบรับรองปริญญานิพนธ์ ประจำปีการศึกษา 2558

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง

ระบบบริการส่วนต่อประสานโปรแกรมประยุกต์เพื่อสังเคราะห์ชื่อและประเภทสินค้าจากเว็บไซต์วิจารณ์สินค้า

APPLICATION PROGRAMMING INTERFACE SERVICE SYSTEM FOR EXTRACTING BRANDS AND PRODUCT TYPES FROM PRODUCT REVIEW WEBSITES

นางสาววิชญาน์นันท์ จิรบวรวณิชย์ รหัสนักศึกษา 55070107

นายศักดา จันทรรวงทอง รหัสนักศึกษา 55070116

.....อาจารย์ที่ปรึกษา

(ดร. กนกวรรณ อัจฉริยะชาญวณิช)

หัวข้อโครงงาน	ระบบบริการส่วนต่อประสานโปรแกรมประยุกต์เพื่อสังเคราะห์ชื่อและประเภทสินค้าจากเว็บไซต์วิจารณ์สินค้า
นักศึกษา	นางสาววิชญาน์นันท์ จิรบวรวณิชย์ รหัสนักศึกษา 55070107
	นายศักดา จันทรรวงทอง รหัสนักศึกษา 55070116
ปริญญา	วิทยาศาสตรบัณฑิต
สาขาวิชา	เทคโนโลยีสารสนเทศ
ปีการศึกษา	2558
อาจารย์ที่ปรึกษา	ดร. กนกวรรณ อัจฉริยะชาญวณิช

บทคัดย่อ

เว็บไซต์วิจารณ์สินค้าในประเทศไทย จะมีรูปแบบการเขียนในภาษาผสม คือกล่าวถึงชื่อและแบรนด์สินค้าในภาษาอังกฤษ ส่วนเนื้อหาจะเป็นภาษาไทย ในปัจจุบันระบบค้นหาชื่อสินค้าที่เป็น API ยังไม่สามารถดึงชื่อสินค้าที่เป็นภาษาอังกฤษจากเว็บไซต์วิจารณ์สินค้าที่มีเนื้อหาหลักเป็นภาษาไทยได้ โครงงานนี้จึงมุ่งไปที่การนำเสนอวิธีการในการสกัดชื่อและประเภทสินค้าจากเว็บไซต์วิจารณ์สินค้า ทีมผู้วิจัยได้พัฒนา API เพื่อให้ผู้ที่สนใจสามารถใช้งานเพื่อสกัดชื่อและประเภทสินค้าได้ผ่านทางเว็บเบราว์เซอร์ จากการทดสอบพบว่าวิธีการที่ใช้รูปแบบของประโยคสามารถสกัดชื่อและประเภทสินค้าได้และมีความแม่นยำที่ดี ผลการทดลองการสกัดหาชื่อสินค้าจำนวน 960 เว็บเพจ ได้ความแม่นยำที่ 97.48% และการทดสอบหาประเภทสินค้าจำนวน 861 เว็บเพจได้ค่าความแม่นยำที่ 92.73% ผลงานของโปรเจคนี้คือระบบบริการส่วนต่อประสานโปรแกรมประยุกต์เพื่อสังเคราะห์ชื่อและประเภทสินค้าจากเว็บไซต์วิจารณ์สินค้า

Project Title	Application Programming Interface Service System for Extracting Brand and Product Type from Product Review Websites
Student	Ms. Vichayanan Jirabovolvanit Student ID 55070107
	Mr. Sakda Jantararuangthong Student ID 55070116
Degree	Bachelor of Science
Program	Information Technology
Academic Year	2015
Advisor	Dr. Kanokwan Atchariyachanvanich

ABSTRACT

Most of Thai product review websites display mixed contents by showing the product name and brand name in English language and review contents in Thai language. However, the current product extraction application programming interfaces (API) could not extract product name written in English language from Thai product review websites. Therefore, this paper proposed the approach to extract brand names and product names from mixed Thai-English language product review websites. our work reiterate method to extracting brand name and product type in review product website. we have created an API for extracting brand name and product type for other developer to use through web browser. The experimental result of extracting brand name and product name from 960 webpages revealed that proposed algorithm had a 97.48% of precision and the result of classifying product type from 861 webpages had a 92.73% of precision. In addition, the output of this project is the prototype of application programming interface service system for extracting brand and product type from product review websites

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยความช่วยเหลือและกรุณาของอาจารย์ที่ปรึกษา ดร. กนกวรรณ อัจฉริยะชาญวณิช ที่ให้แนวคิดริเริ่มในการทำระบบดังกล่าว และให้คำปรึกษาข้อเสนอแนะ ระหว่างการพัฒนาระบบ ทางผู้จัดทำขอขอบพระคุณอาจารย์ที่ปรึกษาเป็นอย่างสูงที่ทำให้โครงงานนี้สำเร็จไปด้วยดี รวมทั้งขอขอบคุณ คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่อำนวยความสะดวกด้านอุปกรณ์และสถานที่ในการพัฒนา

วิชญาน์นันท์ จิรบวรวณิชย์

ศักดา จันทรรวงทอง

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญรูป

รูปที่ หน้า

สารบัญรูป (ต่อ)

รูปที่ หน้า

สารบัญตาราง

ตารางที่ หน้า

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญ

เทคโนโลยีด้านเครือข่ายคอมพิวเตอร์ได้พัฒนาอย่างรวดเร็ว จึงเกิดการขยายตัวของอินเทอร์เน็ตอย่างแพร่หลาย ส่งผลให้เกิดข้อมูลข่าวสารมากมายกระจายอยู่ในอินเทอร์เน็ต ข้อมูลส่วนใหญ่มักจะเป็นข้อมูลสาธารณะทั่วไปที่ไม่มีมูลค่า แต่ถ้าสามารถนำข้อมูลเหล่านั้นเข้ามาผ่านกระบวนการวิเคราะห์ด้วยการทำเหมืองข้อมูลบนเว็บ จะทำให้เกิดข้อมูลใหม่ที่มีสามารถนำมาใช้ประโยชน์ต่อได้ทางสถิติและวิเคราะห์ทางการตลาด โดยการทำเหมืองข้อมูลบนเว็บจำเป็นต้องอาศัยความความรอบรู้ การทดลอง และความเชื่อมโยง จึงควรเข้าใจถึงขอบเขตของปัญหาโดยแท้จริงก่อน เพื่อให้การทำเหมืองข้อมูลบนเว็บ เกิดประโยชน์อย่างแท้จริง

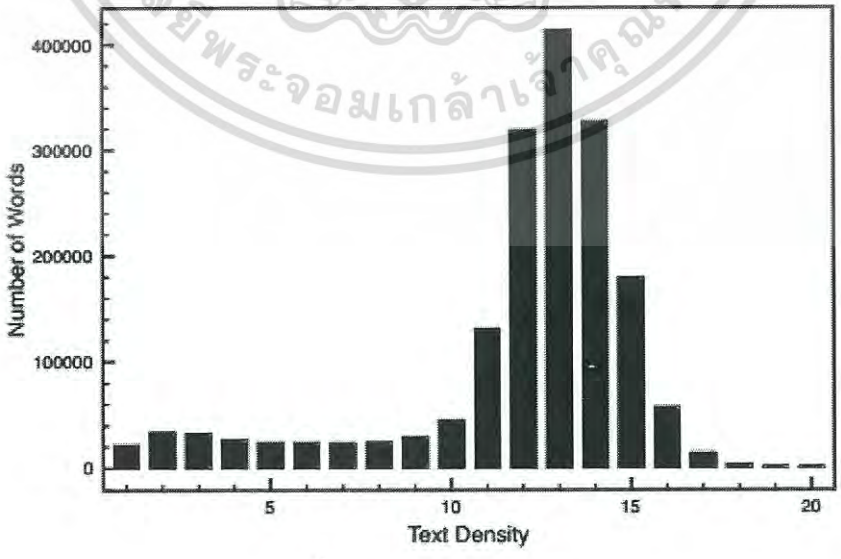
เนื่องจากอัตราการใช้อินเทอร์เน็ตในประเทศกำลังพัฒนาอย่างประเทศไทยสูงขึ้น ทำให้อัตราการเข้าถึงสินค้าผ่านอินเทอร์เน็ตเพิ่มมากขึ้นตามไปด้วย พฤติกรรมผู้บริโภคจึงเปลี่ยนไปจากเดิม การซื้อสินค้าออนไลน์กลายเป็นส่วนหนึ่งในชีวิตประจำวันของผู้คน เพราะเทคโนโลยีต่าง ๆ เข้ามาช่วยอำนวยความสะดวกในชีวิตประจำวัน ส่งผลให้ธุรกิจในปัจจุบันมีอัตราการแข่งขันสูงขึ้น การมีเครื่องมือที่สามารถวิเคราะห์ความต้องการของผู้บริโภคก็ถือเป็นสิ่งจำเป็นต่อที่ธุรกิจ การมีข้อมูลทำให้สร้างความได้เปรียบทางการแข่งขันเหนือคู่แข่ง แต่วิธีการในการทำเหมืองข้อมูลบนเว็บในประเทศไทยยังอยู่ในช่วงคิดค้นพัฒนา เพราะภาษาไทยจะมีความซับซ้อนละเอียดอ่อนกว่าถ้าเทียบกับภาษาอังกฤษ ทำให้การทำเหมืองข้อมูลในภาษาไทยยังไม่ค่อยมีความแม่นยำ และถ้าข้อมูลที่เป็นภาษาไทยอยู่บนเว็บไซต์ยิ่งมีความยุ่งยากในการเขียนระบบเพื่อสังเคราะห์ข้อมูลและจึงเป็นที่มาของงานวิจัยชิ้นนี้ที่จะสร้างระบบบริการส่วนต่อประสานโปรแกรมประยุกต์สำหรับค้นหาชื่อและประเภทสินค้าจากเว็บไซต์สำหรับโครงการวิเคราะห์เว็บไซต์ในอนาคตเพื่อให้ได้วิธีการวิเคราะห์ที่มีประสิทธิภาพดีกว่าเดิมสำหรับในเว็บภาษาไทย

1.2 ความมุ่งหมายและวัตถุประสงค์

1. เพื่อวิจัยและพัฒนาอัลกอริทึมการค้นหาชื่อสินค้าและประเภทจากโครงสร้างเว็บเพจ
2. เพื่อพัฒนาเว็บไซต์ระบบส่วนต่อประสานโปรแกรมประยุกต์สำหรับค้นหาชื่อสินค้าจากโครงสร้างเว็บไซต์วิจารณ์สินค้า

1.3 ทฤษฎีหรือแนวคิดที่ใช้ในการศึกษา

หน้าเว็บเพจทุกอันประกอบด้วยโครงสร้างรูปแบบของภาษาเอชทีเอ็มแอล (Hypertext Markup Language) ซึ่งเป็นภาษามาร์กอัปหลักในปัจจุบันที่ใช้ในการสร้างเว็บเพจ หรือข้อมูลอื่นที่เรียกดูผ่านทางเว็บเบราว์เซอร์ ซึ่งตัวโค้ดจะแสดงโครงสร้างของข้อมูล ในการแสดง หัวข้อ ลิงก์ ย่อหน้า รายการ รวมถึงการสร้างแบบฟอร์ม เชื่อมโยงภาพหรือวิดีโอด้วย โครงสร้างภาษาเอชทีเอ็มแอลจะอยู่ในลักษณะภายในวงเล็บสามเหลี่ยม เรียกว่า Tag มีที่ละเป็นคู่ในชื่อเดียวกันประกอบด้วย Start Tag และ End Tag โดยข้อมูลที่อยู่นอก Tag จะเรียกว่า Text ซึ่งเมื่อวิเคราะห์ความหนาแน่นของ Text (text density)กับจำนวน Tag ในหนึ่งหน้าเว็บเพจ ทำให้สามารถที่จะจำแนกรูปแบบอัตราส่วนเพื่อค้นหาเนื้อหาหลักของเว็บไซต์ได้ จากนั้นเราก็สามารถนำเนื้อหาที่ได้มาวิเคราะห์เพื่อหาชื่อและประเภทสินค้าต่อไป โดยใช้วิธีการวิเคราะห์แบบของประโยคและคำในการอธิบายกล่าวถึงสินค้าในเนื้อหา



รูปที่ 1.1 Text Density [1]

ที่มา : Boilerplate Detection using Shallow Text Features

1.4 ขอบเขตของงาน

1. สังเคราะห์ชื่อและแบรนสินค้าที่เป็นภาษาอังกฤษจากเว็บที่เป็นภาษาไทยเท่านั้น
2. เว็บที่ใช้ในการวิจัยเป็นเว็บที่อยู่ในรูปแบบประเภทของเว็บไซต์วิจารณ์สินค้า

1.5 ขั้นตอนของการศึกษา

1. ศึกษาโครงสร้างรูปแบบของภาษาเอชทีเอ็มแอล
2. ศึกษาระบบการสังเคราะห์ชื่อและประเภทสินค้าจากเว็บเพจที่มีอยู่ในปัจจุบัน
3. ศึกษาโปรแกรมที่ใช้สำหรับสกัดแบ่งคำภาษาไทย
4. คิดวิธีการและอัลกอริทึมพัฒนาส่วนต่อประสานออกแบบหน้าเว็บรูปแบบสำหรับนำเสนอผลลัพธ์จากการสังเคราะห์
5. แก้ไขและปรับปรุงข้อผิดพลาดและปัญหาที่เกิดขึ้น
6. สรุปผลการพัฒนาและจัดทำเอกสารประกอบการพัฒนา

1.6 ประโยชน์ที่คาดว่าจะได้รับ

1. เพื่อเป็นแนวทางในวิธีการสังเคราะห์ชื่อและประเภทสินค้าจากเว็บที่มีความแม่นยำเพิ่มขึ้น
2. เพื่อเป็นแนวทางในการสกัดข้อมูลสินค้าสำหรับโครงการวิเคราะห์เว็บไซต์ในอนาคต
3. ลดต้นทุนด้านค่าใช้จ่ายและเวลา ในการนำส่วนต่อประสานโปรแกรมประยุกต์ไปพัฒนาโปรแกรมส่วนอื่นต่อไปในอนาคต
4. ลดความเสี่ยงและช่วยในการตัดสินใจให้กับผู้ประกอบกิจการรายใหม่ หรือผู้ประกอบการเดิมที่มีความต้องการประกอบกิจการใหม่
5. ช่วยให้ผู้ประกอบกิจการเดิม นำข้อมูลที่ได้มาวางแผนกลยุทธ์โปรโมทส่งเสริมยอดขายสินค้าและบริการ

บทที่ 2
เทคโนโลยีและวรรณกรรมที่เกี่ยวข้อง

2.1 การทำเหมืองข้อมูลเว็บ (Web Mining) [2]

คือการใช้เทคนิคการทำเหมืองข้อมูลเพื่อค้นหาและสกัดข้อมูลสารสนเทศจากเอกสารเว็บและบริการบนเว็บโดยอัตโนมัติ เพื่อนำความรู้ที่ได้มาแก้ปัญหาที่ต้องการทั้งทางตรงและทางอ้อม จากงานวิจัยของการทำเหมืองข้อมูลเว็บ[3] นักวิจัยได้แบ่งประเภทของการทำเหมืองข้อมูลเว็บโดยพิจารณาจากข้อมูลที่นำมาวิเคราะห์ออกเป็น 3 ประเภท คือ Web Content Mining, Web Structure Mining และ Web Usage Mining

2.1.1 Web Content Mining

เป็นการค้นหาข้อมูลที่มีประโยชน์จากข้อมูลที่อยู่ภายในเว็บ เช่น ข้อความ รูปภาพ เป็นต้น โดย Web Content Mining สามารถแบ่งออกเป็น 2 ประเภทตามมุมมองคือ มุมมองทางด้านการสืบค้นสารนิเทศ (Information Retrieval) และมุมมองทางด้านฐานข้อมูล (Database) สำหรับเป้าหมายของ Web Content Mining จากมุมมองของการสืบค้นสารนิเทศคือการทำเหมืองข้อมูลเว็บเพื่อปรับปรุงการหาข้อมูลหรือกรองข้อมูลให้ผู้ใช้โดยพิจารณาจากข้อมูลที่ผู้ใช้อ้างอิงหรือร้องขอ ในขณะที่เป้าหมายของ Web Content Mining ในมุมมองของฐานข้อมูลส่วนใหญ่พยายามจำลองข้อมูลบนเว็บและรวมข้อมูลนั้น เพื่อให้การสอบถามทำงานดีขึ้นมากกว่าการใช้คำหลักเป็นตัวค้นหาเพียงอย่างเดียว

2.1.2 Web Structure Mining

เป็นวิธีการที่พยายามค้นหารูปแบบโครงสร้างการเชื่อมโยงที่สำคัญและซ่อนอยู่ในเว็บ ซึ่งรูปแบบนี้จะขึ้นอยู่กับรูปแบบการเชื่อมโยงเอกสารภายในเว็บ โดยนำรูปแบบที่ได้มาใช้เพื่อจัดกลุ่มเว็บเพจและใช้สร้างข้อมูลสารสนเทศที่เป็นประโยชน์ เช่น นำมาใช้ในการปรับโครงสร้างของเว็บให้สามารถให้บริการผู้ใช้ได้อย่างรวดเร็ว

2.1.3 Web Usage Mining

เป็นวิธีการที่พยายามค้นหาความหมายของข้อมูลที่สร้างจากช่วงการทำงานหนึ่งของผู้ใช้หรือสร้างจากพฤติกรรมของผู้ใช้เรียกอีกชื่อหนึ่งว่า Web Log Mining โดยในขณะที่ Web Content Mining และ Web Structure Mining ใช้ประโยชน์จากข้อมูลจริง หรือข้อมูลพื้นฐานบนเว็บแต่ Web Usage Mining ทำการค้นหาความรู้จากข้อมูลการติดต่อสื่อสารระหว่างกันของผู้ใช้ที่ติดต่อกับเว็บ โดย Web Usage Mining ทำการรวบรวมข้อมูลจากบันทึกในการดำเนินการต่าง ๆ เช่น บันทึกการ

ใช้งานของ Proxy (Proxy Server Log) ข้อมูลการลงทะเบียน (Registration Data) หรือข้อมูลอื่นอันเป็นผลจากการทำงานร่วมกันมาใช้วิเคราะห์ ดังนั้น Web Usage Mining จึงเป็นวิธีการทำงานที่เน้นใช้เทคนิคที่สามารถทำนายพฤติกรรมของผู้ใช้ในขณะที่ผู้ใช้ทำงานกับเว็บ

กระบวนการทำงานของ Web Usage Mining สามารถแบ่งออกเป็น 2 วิธีคือ

1. ทำการจับคู่ข้อมูลการใช้งานของเครื่องให้บริการเว็บให้อยู่ในรูปของตารางความสัมพันธ์ก่อนที่นำข้อมูล นี้มาปรับใช้ กับเทคนิคการทำเหมืองข้อมูลการใช้เว็บ
2. ใช้ประโยชน์จากข้อมูลในบันทึกการใช้งานโดยตรงซึ่งจะใช้เทคนิคการเตรียมข้อมูล (Preprocessing) เพื่อเตรียมข้อมูลก่อนหาความสัมพันธ์ (Pattern Discovery) และวิเคราะห์รูปแบบ (Pattern Analysis)

2.2 Web Mining กับการทำธุรกิจ e-Commerce [4]

ในการทำธุรกิจต่าง ๆ ไม่ว่าจะในรูปของห้างร้าน บริการสั่งของทางไปรษณีย์ หรือการทำธุรกิจแบบอิเล็กทรอนิกส์นั้น ปัจจัยหนึ่งที่มีความสำคัญอย่างมากต่อความสำเร็จของธุรกิจก็คือความเข้าใจในตัวลูกค้า หรือกลุ่มลูกค้า ยิ่งรู้ข้อมูลมากทำให้เข้าใจลูกค้าอย่างแท้จริงมากขึ้น โอกาสที่จะทำธุรกิจให้ตรงกับความต้องการของตลาดก็จะมีมากขึ้นไปด้วย

ข้อมูลของลูกค้าดังกล่าวมานี้ ความจริงแล้วมีให้นำมาใช้ได้มากมายอยู่แล้ว แต่อาจจะอยู่ในรูปที่เป็นได้ไม่ชัดเจน อันได้แก่ ข้อมูลที่รวบรวมไว้จากการบันทึกใน log file ของการใช้บริการเว็บ หรือข้อมูลจากการสมัครสมาชิกในรูปแบบต่าง ๆ เป็นต้น ข้อมูลเหล่านี้สามารถอำนวยประโยชน์ในการติดตามดูผู้ใช้ (user tracking) ยิ่งผู้ใช้เข้าใช้เว็บ บ่อยและนานขึ้น ยิ่งมีโอกาสทราบและรู้จักกับผู้ใช้มากขึ้นเท่านั้น สำหรับข้อมูลดังกล่าวเกี่ยวกับผู้ใช้จะมีการวิเคราะห์ออกมาใน 3 ลักษณะดังต่อไปนี้

1. Demographics เป็นข้อมูลเกี่ยวกับที่อยู่ หรือสถานที่ของผู้ใช้ในขณะที่เข้าใช้บริการ web ซึ่งจะสามารถประมวลเป็นสถิติบริเวณที่อยู่อาศัยของกลุ่มผู้ใช้ส่วนมากได้
2. Psychographics เป็นข้อมูลด้านจิตวิทยา ซึ่งแสดงถึงพฤติกรรม หรือค่านิยมในด้านต่าง ๆ ของผู้ใช้ โดยสามารถจะแบ่งแยกกลุ่มผู้ใช้ตามข้อมูลการเข้าใช้บริการ web ทั้งในแง่ของเวลาและเนื้อหา

3. Technolographics เป็นข้อมูลที่แสดงถึงระดับความรู้และความสนใจในเทคโนโลยีด้านต่าง ๆ ของผู้ใช้ รวมถึงเครื่องคอมพิวเตอร์ที่ติดต่อเข้ามาด้วย ซึ่งจะช่วยในการพัฒนาสินค้าและบริการให้น่าสนใจและเหมาะสมในแง่ของเทคโนโลยีได้ดีขึ้น

เมื่อนำข้อมูลที่วิเคราะห์แล้วทั้ง 3 ลักษณะนี้มาพิจารณาโดยละเอียด จะเกิดประโยชน์อย่างมากในการศึกษาเกี่ยวกับสภาพ และพฤติกรรมโดยรวมของประชากร ซึ่งจำนวนข้อมูลที่นำมาใช้วิเคราะห์มักจะมีจำนวนมากและให้ผลการวิเคราะห์ที่มีความแม่นยำสูง

2.3 Web Server API [5]

API ย่อมาจาก Application Programming Interface คือช่องทางการเชื่อมต่อระหว่างเว็บไซต์หนึ่งไปยังอีกเว็บไซต์หนึ่ง หรือเป็นการเชื่อมต่อระหว่างผู้ใช้งานกับ Server หรือจาก Server เชื่อมต่อไปหา Server ซึ่ง API นี้เปรียบได้เป็นภาษาคอมพิวเตอร์ที่ทำให้คอมพิวเตอร์สามารถสื่อสารและแลกเปลี่ยนข้อมูลกันได้อย่างอิสระโดยส่วนมากแล้วจะเห็นได้ว่า API ถูกใช้งานกันอย่างแพร่หลาย ที่เห็นได้อย่างชัดเจนนั่นก็คือ บริการของ Amazon นั่นมี API ที่เปิดให้ผู้ที่สนใจที่จะเป็นตัวแทนขายสินค้าหรือเจ้าของเว็บทั่วไป ได้นำสินค้าที่มีขายอยู่ใน Amazon ไปติดไว้ในเว็บไซต์หรือบล็อคของตัวเองได้ โดยเจ้าของเว็บไซต์หรือผู้สนใจจะได้รับคอมมิสชั่นเมื่อมีการคลิกซื้อสินค้าจากเว็บไซต์หรือบล็อคที่นำ API ไปติดตั้ง อีกบริการหนึ่งก็คือบริการของ PayPal API ซึ่งเจ้าของเว็บไซต์ที่ต้องการเพิ่มช่องทางการชำระเงินให้กับลูกค้าก็สามารถนำ PayPal API ไปติดตั้งที่เว็บไซต์ที่ต้องการได้ เพื่อเพิ่มความสะดวกสบายให้กับลูกค้าที่มาใช้บริการในเว็บไซต์นั่นเอง

2.4 การดึงข้อมูลด้วย Web Scraping

Web Scraping เป็นระบบที่ดึงข้อมูลจากเว็บเพจส่วนที่ต้องการมาเรียกใช้งาน โดยใช้โมดูล Requests ซึ่งเป็นโมดูลสำหรับงาน Requests กับ HTTP โดยโมดูลนี้ถูกออกแบบให้ใช้งานได้กับมนุษย์ โดยมีหน้าที่คล้ายกับไลบรารี Urllib ของไพทอน แต่ใช้งานได้สะดวกและง่ายกว่า ใช้ License: Apache 2.0 รองรับภาษาไพทอนเวอร์ชั่น 2 และ 3

2.5 เทคนิคการตัดคำภาษาไทย [6]

การเขียนรูปประโยคในภาษาไทยจะไม่มีการเว้นวรรคระหว่างคำต่าง ๆ เหมือนอย่างในภาษาอังกฤษ ดังนั้นการที่จะสกัดหาชื่อสินค้าจากประโยคบทความในเว็บเพจนั้นจึงมีขั้นตอนที่ยุ่งยากมากกว่า จึงต้องใช้วิธีการตัดแบ่งประโยคภาษาไทยให้เป็นคำแยกกัน

เทคนิคการตัดคำภาษาไทยที่มีในปัจจุบัน

การตัดคำไทยในปัจจุบัน มีด้วยกัน 3 เทคนิค คือ

1. ใช้กฎไวยากรณ์ทางภาษา (Rule - Based)

แนวทางนี้ทำได้ง่ายที่สุด ทำงานได้เร็วที่สุด แต่แบ่งคำพยางค์เดียวได้เท่านั้น ไม่สามารถจัดการกับคำหลายพยางค์ได้ อีกทั้งยังไม่สามารถแก้ปัญหาความกำกวมของพยัญชนะที่เป็นได้ทั้งพยัญชนะต้น และตัวสะกด ดังเช่น “ก” ใน “ตากลม” ได้

2. ใช้คำจากพจนานุกรม (Dictionary - Based)

คือการทำรายการคำเอาไว้ล่วงหน้า เมื่อต้องการแบ่งคำก็เปรียบเทียบข้อความที่ต้องการแบ่งกับรายการคำที่เก็บ ไว้ในพจนานุกรม วิธีนี้สามารถแก้ปัญหาคำหลายพยางค์ได้ แต่ยังไม่สามารถแก้ปัญหาคำกำกวมได้ทั้งหมด

3. ใช้เทคนิคการเรียนรู้ (Machine Learning or Corpus Based)

เป็นวิธีที่ได้รับความนิยมที่สุดในปัจจุบัน โดยการฝึกฝนระบบด้วยคลังข้อความขนาดใหญ่ที่มีการแบ่งคำไว้อย่างถูกต้องแล้ว เพื่อให้เครื่องได้เรียนรู้ด้วยตนเอง จากการเก็บสถิติ และคำนวณค่าความน่าจะเป็นของการปรากฏร่วมของคำที่อยู่ติด ๆ กัน ประสิทธิภาพของวิธีนี้ ขึ้นอยู่กับความถูกต้องและขนาดของคลังข้อความ

2.6 โปรแกรมสำหรับแบ่งคำภาษาไทย

ในปัจจุบันมีโปรแกรมสำหรับแบ่งคำภาษาไทยมากขึ้นกว่าแต่ก่อน แต่ความสามารถและโครงสร้างของโปรแกรมที่มีอยู่นั่นยังไม่สามารถรองรับได้ในทุกภาษาคอมพิวเตอร์ ทีมผู้วิจัยจึงได้ทำการเลือกโปรแกรมสำหรับแบ่งคำภาษาไทยที่สามารถรองรับภาษาไพทอนได้ มาจำนวน 4 โปรแกรมดังนี้

1. pyICU เวอร์ชั่น 1.9.2
2. Py-grabber
3. Eatiht เวอร์ชั่น 0.1.1.4
4. Goose Extractor

โดยคุณสมบัติที่ต้องการจากโปรแกรมดังกล่าวมีดังนี้

1. สามารถแบ่งคำภาษาไทยได้อย่างแม่นยำ
2. สามารถแบ่งคำภาษาอังกฤษที่ปนกับคำภาษาไทยได้
3. รองรับภาษาไพทอน เวอร์ชั่น3.4

2.7 เทคโนโลยีที่ใช้ในการพัฒนา

เทคโนโลยีพื้นฐานที่นำมาใช้ในการพัฒนาโครงการนี้ ประกอบไปด้วย

2.7.1 MySQL [7]

MySQL เป็นโปรแกรมจัดการฐานข้อมูล Relational Database Management System (RDBMS) เป็นฐานข้อมูลที่สามารถจัดเก็บ ค้นหา เรียงข้อมูล และดึงข้อมูล MySQL มีความสามารถให้ผู้ใช้งานเข้าดึงข้อมูลได้หลาย ๆ คนในเวลาเดียวกันได้และมีการเข้าถึงข้อมูลที่รวดเร็ว มีการกำหนดการเข้าใช้งานของผู้ใช้ในแบบต่าง ๆ อย่างเหมาะสม ปลอดภัย

รูปที่ 2.1 เทคโนโลยี MySQL [8]

ที่มา : https://en.wikipedia.org/wiki/File:MySQL.svg

2.7.2 Pymysql

Pymysql เป็น Package Index ของภาษาไพทอนที่ใช้ในการเชื่อมต่อฐานข้อมูล MySQL กับโปรแกรมไพทอนให้สื่อสารทำงานด้วยกันได้

รูปที่ 2.2 เทคโนโลยี Pymysql [9]

ที่มา : http://www.eceforge.com/wp-content/uploads/2012/07/sql_plus_python.png

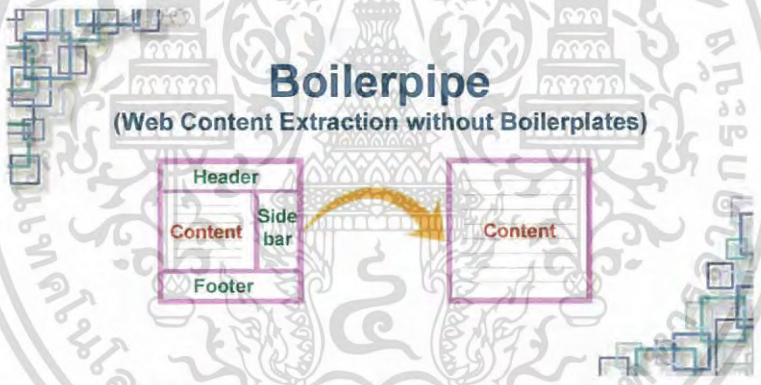
2.7.3 PyICU 1.9.2

PyICU หรือ Python extension wrapping IBM's International Component for Unicode C++ library เป็น API ใช้จัดการกับ String Unicode ในภาษาต่างๆรวมถึงภาษาไทย มีความสามารถดังนี้

1. ได้ผลลัพธ์การแบ่งคำภาษาไทยที่ถูกต้องแม่นยำ
2. สามารถใช้ได้กับ Sting ที่เป็นภาษาไทยปนกับภาษาอังกฤษ
3. เป็น Open Source สามารถใช้ได้ไม่มีค่าใช้จ่าย

2.7.4 Package Index Boilerpipe

Package Index Boilerpipe เป็น Library ที่มีอัลกอริทึมสำหรับค้นหาและทำลายส่วนเกินของใจความสำคัญหลักในเว็บเพจ เพื่อให้ได้ซึ่งข้อมูลเพื่อนำไปเข้าสู่กระบวนการต่อไป ภายใต้วิธีการของ Paper "Boilerplate Detection using Shallow Text Features"



รูปที่ 2.3 เทคโนโลยี Package Index Boilerpipe [10]

ที่มา : http://www.treselle.com/wp-content/uploads/freshizer/4d8de2ae9526bef9fe678425384a3ed0_boilerpipe1-863-430-c.jpg

2.7.5 VMware Workstation / Window Server 2012 R2 [11]

VMware Workstation คือ โปรแกรมจำลองเครื่องคอมพิวเตอร์ หลาย ๆ เครื่อง ในคอมพิวเตอร์เพียง 1 เครื่อง ในการใช้งานสามารถประยุกต์ใช้งานได้หลากหลาย Window Server 2012 R2 เป็น Window สำหรับทำเซิฟเวอร์

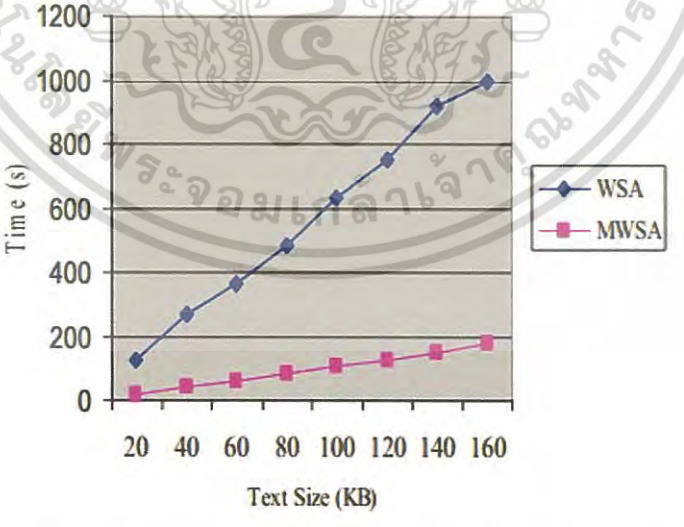
รูปที่ 2.4 เทคโนโลยี Server : VMware Workstation 12 / Window Server 2012 R2 [12]

ที่มา : http://itiscloudy.com/wp-content/uploads/2015/08/logo_winserver2012R2.png

2.8 งานวิจัยที่เกี่ยวข้อง

2.8.1 An Efficient Word Searching Algorithm through Splitting and Hashing the Offline Text

บทความวิจัยนี้กล่าวถึงอัลกอริทึมที่ใช้ในกระบวนการค้นหาคำหนึ่งคำจากฐานข้อมูลขนาดใหญ่ ด้วยวิธีการใหม่เรียกว่า modified word searching algorithm (MWSA) โดยพัฒนามาจากวิธีเดิม Word Searching Algorithm (WSA) มีหลักการโดยใช้วิธีค้นหาจาก ความยาวของคำและการนำคำมาเข้ารหัส Hash โดยใช้ SBDM Hash Function และเรียงในฐานข้อมูลโดยใช้ วิธีการ Insertion Sort



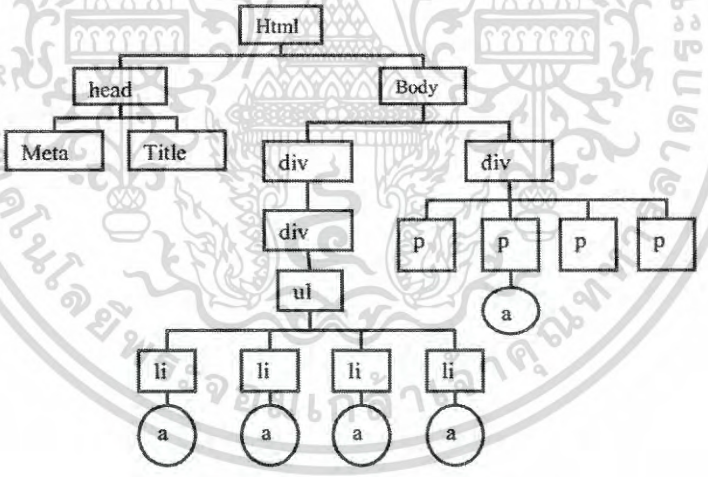
รูปที่ 2.5 เปรียบเทียบเวลาต่อขนาดข้อมูลระหว่าง WSA , MWSA [13]

2.8.2 Removing Noise Content from Online News Articles

ในหน้าเว็บข่าวต่าง ๆ จะมีสิ่งที่สนใจคือ เนื้อหาข่าว (Content) ออกจากสิ่งแปลกปลอมใน HTML เรียกว่า Noise เช่น ลิงค์, โฆษณา, ลิขสิทธิ์ เพราะมันจะทำให้การทำเหมืองข้อมูลผิดพลาด โดยวิธีพื้นฐานคือ

1. การหา Informative Tags โดยใช้เทคนิคที่หลากหลาย
2. เทคนิคการเก็บสถิติแต่ละ Tags
3. เปรียบเทียบรูปแบบกับข่าวอื่นในเว็บเดียวกันเพื่อหาส่วนที่เหมือนกัน

ในรูปที่ 2.6 แสดงถึง Dom tree หรือก็คือโครงสร้างของ HTML ในหน้าเว็บเพจนั้น ๆ รูปแบบเป็นต้นไม้มีกิ่งต่าง ๆ เรียกว่า โหนด ซึ่งในโหนดบางโหนดมีประเภทข้อมูลที่เหมือนกันในแต่ละหน้า เรียกว่า Static Noise Tags แต่ว่าในบางโหนดมีประเภทข้อมูลที่เหมือนกันในแต่ละหน้า เรียกว่า Static Noise Tags แต่ว่าในบางโหนด เช่น Recent Articles, Mostly Read Article, Relate News เป็นส่วนที่ไม่ซ้ำกันในเว็บ เรียกว่า Dynamic Noise Tags



รูปที่ 2.6 DOM tree of news web page [14]

ซึ่งผู้จัดทำโครงการของบทความนี้ได้เลือกใช้ 2 วิธี ที่ชื่อว่า StaDyNoT เพื่อลบ Static, Dynamic Noise ดังนี้

ขั้นแรกคือการ Mark Static Noise Tags โดยใช้ข้อมูลเว็บในเว็บเดียวกัน

ขั้นสองคือการ Mark Dynamic Noise Tags โดยใช้ Common Ancestar Technique บน Hyperlink Node ของ DOM tree

หากได้รับเว็บเพจ จะทำการทำข้อมูลให้สะอาด เช่น การ lower case, ตัวหน้า unFormatted Text จะง่ายที่สุดในการคำนวณ โดยใช้ตัววิเคราะห์พื้นฐานของ Precistion, Recall, F1

Approach: StaDyNoT

ในขั้นตอนนี้คือการนำ Static/Dynamic Tags ออก มี 2 ขั้นตอน

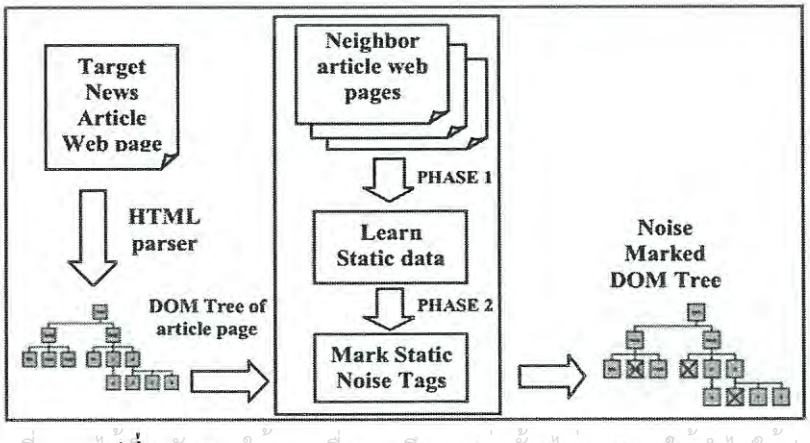
1. Marking Static Noise
2. Marking Dynamic Noise ในแต่ละสเตจจะ Mark Tags โดยไม่ทำการลบทิ้ง เพราะการลบ Noise อาจทำให้ DOM tree ไม่ถูกต้อง แล้วจะไม่สามารถนำไปใช้ใน Subsequent Stage ต่อได้

Mark Static Noise Tags

ในขั้นตอนนี้เป็นการหา Tags จาก "Neighbor Article Web Page" หรือในที่นี้คือ บทความอื่นในเว็บเดียวกัน และมี Category เดียวกัน โดยในรูปที่ 2.7 จะมีการแบ่งเป็น 2 ขั้นตอน ดังนี้

ขั้นที่ 1. คือการวิเคราะห์ NAWP เรียน Static Data

ขั้นที่ 2. Mark Static Tags ด้วย Noise Tags ในขั้นนี้คือการเปลี่ยนทุก ๆ Article ใน Pages NAWP ให้อยู่ในรูป DOM tree โดยแยก Tags name, attribute, data ทุกโหนดใน DOM tree ใส่ลง Hash Table โดยใช้ Key "tag-name: tag-attribute: tag-data" ด้วย Support Value1 ถ้ามี Key อยู่แล้วให้ +1 แทน สุดท้ายจะทำการนับค่าแต่ละค่าอีกครั้งหนึ่ง ใน Hash Table และ Output ว่าเท่ากันไหมใน NP (Neighbor Page) ด้วย Static Tags ถ้าสำรวจ DOM tree ของเพจเป้าหมายหากโหนดใด ตรงกับ Identified Static Noise Tags จะใส่ Marking ด้วย Attribute "NOISE" ใน Tags แทน

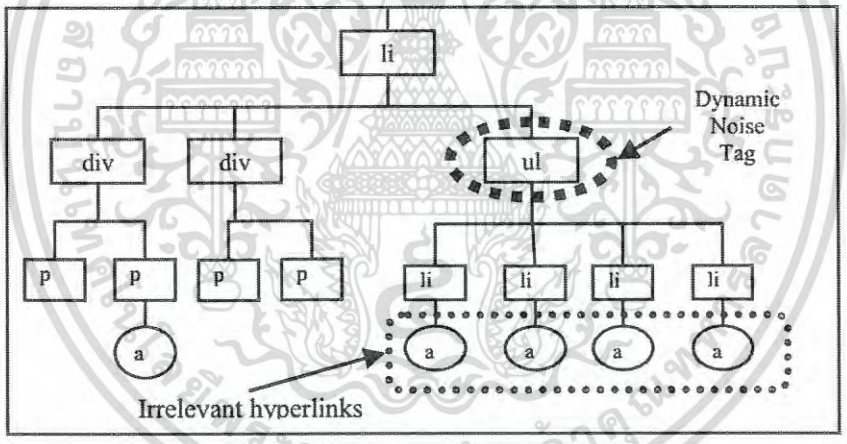


รูปที่ 2.7 The process of marking *Static Noise Tags* [14]

Mark Dynamic Noise Tags

ในขั้นตอนนี้จะพบว่ามี Noise Node ใน DOM tree มี Dynamic Contents เช่นโหนดเกี่ยวกับ "Recent News" ส่วนนี้จะประกอบด้วย Hyperlink และอยู่ใน <ul> Tags ซึ่งแต่ละ Tags มีเปอร์เซ็นต์สูงที่จะเป็น Hyperlink Content เรียกว่า Dynamic Noise Tags

ในรูปที่ 2.8 เสนอในตัวอย่าง Tags <ul> ถูกพบว่าเป็น Dynamic Tags บาง Hyperlinks อาจเกิดขึ้นใต้ Article Text Node แต่ว่าเปอร์เซ็นต์ของ Hyperlink ต่ำ ในแต่ละ Article Text Node โดยเล็งที่จะระบุและชี้ Dynamic Noise Tag ใน DOM tree ของเว็บเป้าหมาย โดยให้แต่ละโหนดที่เป็น Tags "a" ด้วย String (Part-String) โดย Part-String ที่โหนด n เป็น Path จาก Root ถึงโหนด n กับตำแหน่งที่อยู่ให้ทุกโหนด ยกตัวอย่างเช่น "A Path-String For Node With Tags 'p' " : 0(html) - 1(body) - 12(div) - 0(div) - 3(p) หลังจากได้รับ Part String แต่ละโหนดจะทำการพิจารณาโหนด Parent ของตัวมัน ใน DOM tree แล้ว Mark เป็น "Candidate Dynamic Noise Tags"



รูปที่ 2.8 The process of marking *Static Noise Tags* [14]

อย่างไรก็ตามอาจจะมี Hyperlink Node ใน Article Text node ที่ถูกระบุว่าเป็น Dynamic Noise ตาม LCA method ฉะนั้นจึงมี Filtering 2 วิธี

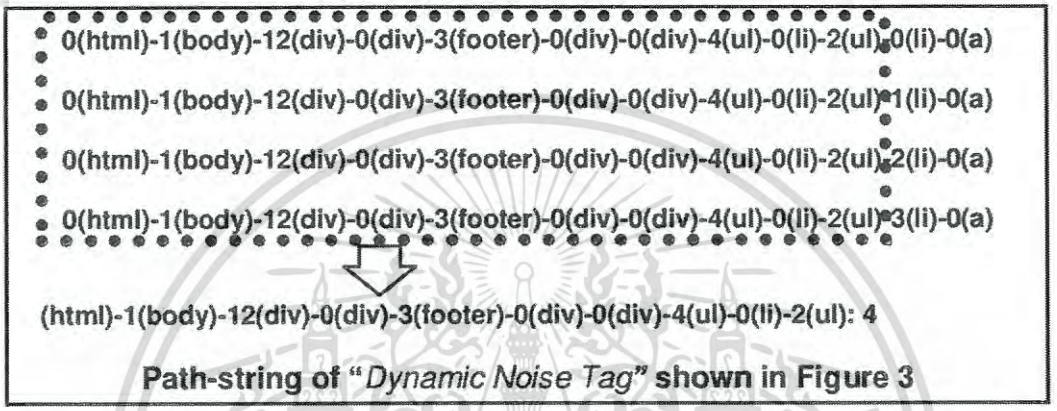
1. Candidate Dynamic Noise Tag with <p> tag cannot be a "Dynamic Noise Tag"
2. Candidate Noise จะเป็น Dynamic Noise เมื่อ

   2.1 เมื่อนับ Non-Link Text ในโหนดมีค่าน้อยกว่า Threshold α หรือ

   2.2 ถ้าส่วนหนึ่งของลิงค์โหนด n (LF_n) มีมากกว่า β ด้วยสูตร

$$\text{Link Fraction}(LF_n) = \frac{\text{Link Text count of node n}}{\text{Total Text count of node n}} \quad (2.1)$$

โดย α และ β คือ ค่าเริ่มต้นของ Non-Link Text และ Link Fraction สังเกตุว่าการกำหนดค่า $\alpha = 50$, $\beta = 20$ ในขั้นตอนนี้อาจ Re-Mark ส่วนหนึ่งของโหนดในสเตจ 1



รูปที่ 2.9 Ilustration of identifying Dynamic Noise nodes by applying LCA on path-strings [14]

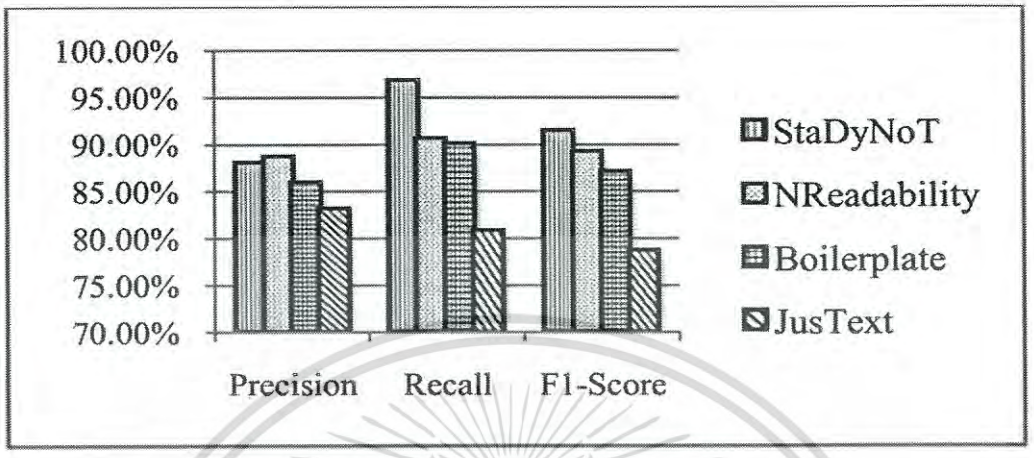
จะเห็นส่วนที่เหมือนกัน (path - string) ออกมาได้ว่า 0(html) - 1(body) - 12(div) - ... - 2(ul) : 4ตัว เป็น Path-String ของ Dynamic Noise Tags จากนั้นจะทำต่อ 2 ขั้นตอนคือ

1. Remove Noise Marked Tags จากที่ Mark DOM tree ด้วย Attribute "NOISE" ใน Tags จึงสามารถหาและลบ Noise ออกได้
2. Remove non - inFormative Formatting Tags คือ Tags ที่ไม่มีข้อมูลสำคัญ เช่น <img>, <link>, <script>, <input>, <form>

วิธีการข้างต้นเป็นการทำให้หน้าเพจเหลือเพียงแค่ Article Text Content เท่านั้น และเป็น DOM tree ที่มีแค่ Article Content และจากการทดสอบ StaDyNoT กับเทคนิคอื่นๆ 440 ข่าว ใน 1 ข่าว จะมีคำเฉลี่ย 544 คำ จาก 11 เว็บไซต์ ใช้ข่าว 10 ประเภท วัดด้วยตัวแปร Precistion, Recall, F1*** Scores ให้ผลที่ดีมาก เปรียบเทียบกับ StaDyNoT, NReadability, Boilerplate, JusText StaDyNot มี Recall, F1 สูงที่สุดแต่ Precistion เป็นรอง NReadability

วิธีของผู้จัดทำโครงการคือระบุ Static, Dynamic โดยใช้ Neighbor Web Pages เปรียบเทียบ โดย Static Tags จะเป็นตัวเดิมทุกครั้ง แต่ Dynamic จะเปลี่ยนรูปแบบข้อมูลไป ผู้จัดทำโครงการ

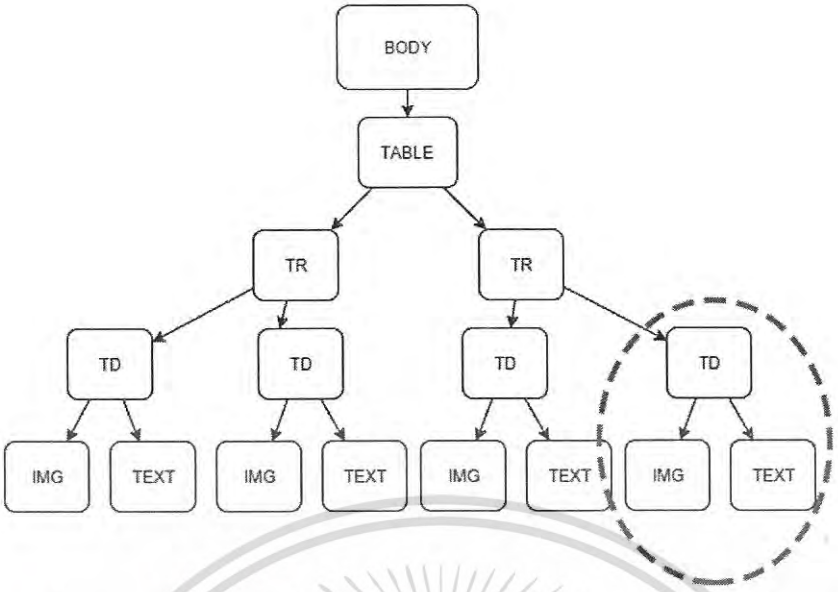
เสนอ 2 วิธีให้มีประสิทธิภาพ ในการระบุ Noise วิธีนี้จะได้ Recall Score ถึง 96% ของข้อมูล Article Content ในเว็บเพจ และ Precision 88% เป็นค่าที่รับได้



รูปที่ 2.10 Result comparison among all the techniques [14]

2.8.3 An algorithm of product information extraction from web pages

กล่าวถึงวิธีการค้นหาชื่อสินค้าและราคาจากเว็บเพจในเว็บไซต์ร้านค้าออนไลน์ โดยใช้ความรู้ในเรื่อง Document Object (DOM) วิธีนี้สามารถหาข้อมูลสินค้าได้จากโครงสร้าง DOM ของข้อมูลสินค้าในเว็บร้านค้าจะมีลักษณะโครงสร้างเดียวกัน ส่วน DOM ที่มีข้อมูลสินค้าภายในจะประกอบไปด้วยชื่อสินค้าที่เป็น Header, ราคาสินค้าที่เป็นตัวเลข, และรูปภาพสินค้าอย่างละหนึ่งอัน หากโปรแกรมวิเคราะห์เว็บเพจแล้วพบโครงสร้างแบบนี้ก็จะสามารถตีความได้ว่าภายในคือข้อมูลสินค้า [15]



รูปที่ 2.11 รูปภาพแสดงโครงสร้างของ DOM ที่ภายในบรรจุข้อมูลสินค้า [15]

2.8.4 A framework for laptop review analysis

นำเสนอวิธีวิเคราะห์เพื่อสรุปคุณสมบัติในด้านต่างๆออกมาเป็นข้อดีและข้อเสียของ Laptop จากเว็บไซต์แลกเปลี่ยนความคิดเห็น โดยแบ่งออกเป็นด้านประสิทธิภาพ, ด้านการออกแบบ และด้านความสามารถ โดยหลักการของ Part of Speech มาวิเคราะห์ประโยคเพื่อหาประธานและกรรม จากนั้นจึงนำ Emoticon Lexicon สำหรับบ่งบอกว่าประโยคนั้นๆเป็นประโยคด้านบวกหรือด้านลบ จากนั้นใช้ Machine Learning จำแนกประโยคว่ากล่าวถึงคุณสมบัติด้านไหนของ Laptop [16]

ตารางที่ 2.1 คำศัพท์ที่บ่งบอกคุณสมบัติของ Laptop ถูกแบ่งโดยวิธี Machine Learning

Performance	Design	Feature
Performance	Design	Feature
Ghz	Weight	USB
Mhz	Width	HDMI
CPU	Height	VGA
Memory	Size	Touchpad

บทที่ 3
การวิเคราะห์และระบบปัจจุบัน

3.1 การวิเคราะห์ระบบที่มีในปัจจุบัน

ปัจจุบันมีเว็บไซต์หลายแห่งสำหรับบริการส่วนต่อประสานโปรแกรมประยุกต์สำหรับสกัดและวิเคราะห์ข้อมูลจากเว็บไซต์มีทั้งแบบเชิงพาณิชย์และแบบไม่แสวงหาผลกำไร เพื่อใช้ในทางการศึกษาวิเคราะห์การตลาดในธุรกิจ ซึ่งแต่ละระบบนั้นมีความสามารถแตกต่างกันออกไป ทีมผู้วิจัยจึงทำการทดสอบเว็บไซต์ที่ให้บริการสกัดเนื้อหามาทดสอบจำนวน 5 เว็บไซต์ดังนี้

1. Diffbot
2. AlchemyAPI
3. Boilerpipe
4. Embed.ly
5. Readability

โดยทำการทดสอบคุณสมบัติของเว็บไซต์ที่สามารถเปรียบเทียบกันได้จาก

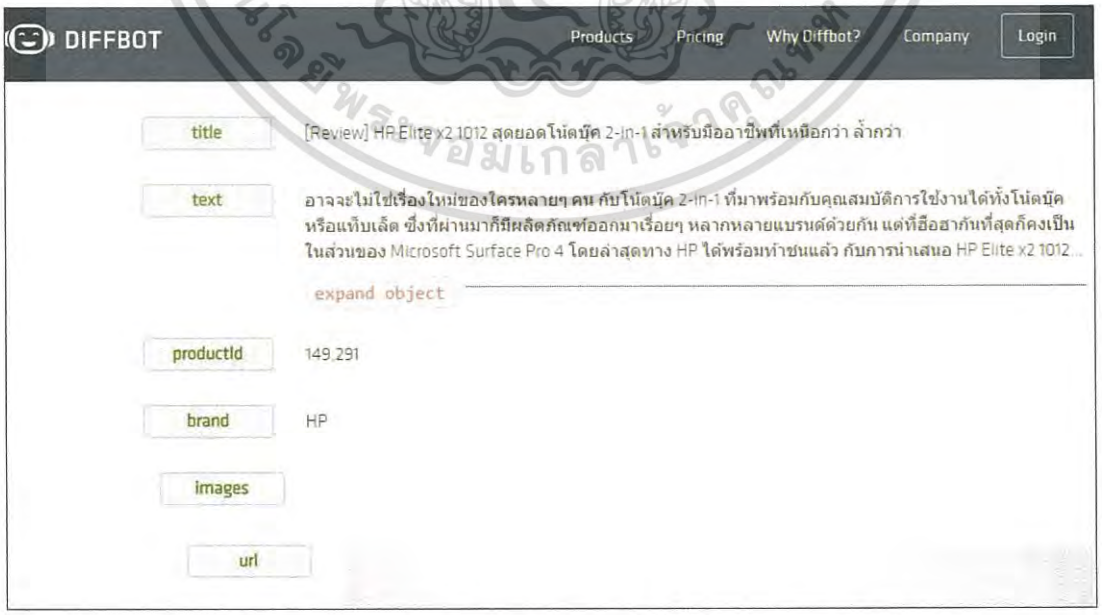
1. Article extraction API คือสามารถสกัดเนื้อหาหลักออกจากหน้าเว็บได้
2. Product extraction API คือสามารถสกัดชื่อสินค้าและผลิตภัณฑ์ออกจากหน้าเว็บได้
3. Returns clean plaintext คือผลลัพธ์จากการสกัดเนื้อหาหลักออกจากหน้าเว็บสามารถนำ Tag ออกจากเนื้อหาได้อย่างสมบูรณ์
4. Language detection คือสามารถระบุภาษาของเนื้อหาในหน้าเว็บนั่นได้
5. Support Thai Language คือสามารถสกัดเนื้อหาหลักออกจากหน้าเว็บภาษาไทยได้อย่างถูกต้อง

จากการทดสอบระบบที่สามารถสกัดเนื้อหาหลักออกจากหน้าเว็บในปัจจุบันได้ผลดังนี้

ตารางที่ 3.1 ผลการทดสอบระบบที่มีในปัจจุบัน

	diffbot	AlchemyAPI	boilerpipe	embed.ly	Readability
1. Article extraction API	✓	✓	✓	✓	✓
2. Product extraction API	✓				
3. Returns clean plaintext	✓	✓	✓		
4. Language detection	✓	✓		✓	
5. Support Thai Language	✓				

จากตารางที่ 3.1 การทดสอบคุณสมบัติในข้อ 1 – 4 ทีมผู้วิจัยได้ดึงผลลัพธ์จากการทดสอบมาจากเว็บ diffbot.com ที่ทำการทดสอบไว้อยู่แล้ว และจากคุณสมบัติในข้อ 5 จะเห็นได้ว่ามีเพียง Diffbot เท่านั้นที่รองรับภาษาไทย แต่ยังไม่ค่อยมีความแม่นยำสำหรับเว็บประเภทวิจารณ์สินค้า ในการหาชื่อสินค้าและไม่สามารถบอกประเภทสินค้าได้



รูปที่ 3.1 ผลการทดสอบ Diffbot API ด้วยเว็บไซต์วิจารณ์สินค้า

ในรูปที่ 3.1 แสดงถึงผลการทดสอบ Diffbot API ด้วยเว็บไซต์ประเภทวิจารณ์สินค้า ในส่วนของ Product API จะเห็นได้ว่าผลลัพธ์ที่ออกมาไม่สามารถบอกถึงชื่อสินค้าได้ บอกได้เพียงแค่แบรนด์ของสินค้านั้นๆ จึงพอสรุปได้ว่า Diffbot API ยังไม่มีความสามารถในการหาชื่อสินค้า

3.2 การวิเคราะห์ความต้องการของระบบ

3.2.1 ความต้องการที่เป็นฟังก์ชั่นหลักของระบบ (Functional Requirement)

Client System : สามารถทดลองหาชื่อกับประเภทสินค้าและใจความหลักจากเว็บไซต์

Client User : สามารถหาชื่อกับประเภทสินค้าและใจความหลักจากเว็บไซต์ผ่าน API โดยใช้ภาษากลาง เป็นภาษา JSON

3.2.2 ความต้องการที่ไม่ใช่ฟังก์ชั่นหลักของระบบ (Non-Functional Requirement)

- เว็บไซต์และระบบมี User Experience (UX) ที่ใช้งานง่าย โดยมีหลักการออกแบบที่ผู้ใช้สามารถทำความเข้าใจได้ง่ายที่สุด
- ระบบตอบสนองการเรียกใช้งานได้เร็ว
- มีระบบตรวจจับการโจมตีแบบ Distributed Denial of Service (DDOS)

3.3 แผนภาพยูสเคส (Use-Case Diagram)

เป็นแผนภาพที่แสดงความสามารถของระบบที่ผู้ใช้สามารถทำได้ ดังรูปที่ 3.2

รูปที่ 3.2 แผนภาพยูสเคสของระบบส่วนต่อประสานโปรแกรมประยุกต์เพื่อการสังเคราะห์ชื่อและประเภทสินค้าจากเว็บไซต์

ตารางที่ 3.2 คำอธิบายยูสเคส หาชื่อและประเภทสินค้า

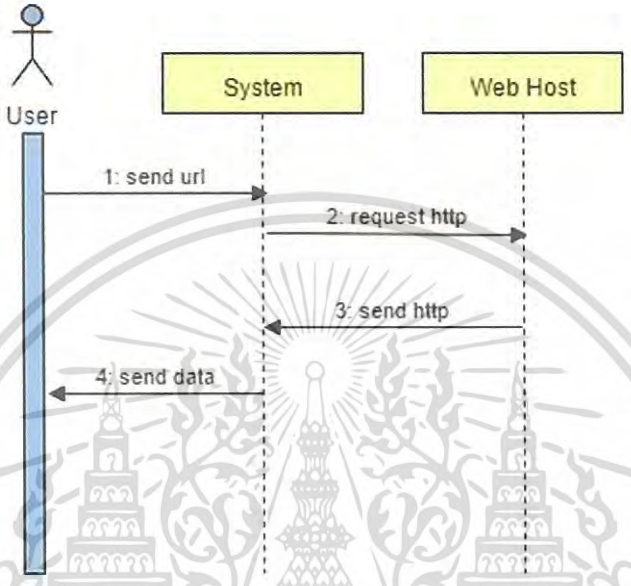
Use Case Name:	หาชื่อและประเภทสินค้า	
Triggering Event:	เมื่อผู้ใช้ต้องการหาชื่อและประเภทสินค้าจากหน้าเว็บ	
Brief Description:	เป็นการหาชื่อและประเภทสินค้าจาก URL ที่กรอกเข้ามา	
Actors:	Client User ,Client System	
Precondition:	ผู้ใช้ต้องส่ง URL เข้าสู่ระบบ	
Post conditions:	ระบบแสดงชื่อและประเภทสินค้าของ URLที่ส่งเข้ามา	
Flow of Events	Actor	System
	1. ส่ง URL เข้าระบบ	2. ตรวจสอบ URL 3. ดึงหน้าเว็บจาก URL 4. ประมวลผล 5. แสดงชื่อและประเภทสินค้า

ตารางที่ 3.3 คำอธิบายยูสเคส แยกเนื้อหาหลักออกจากหน้าเว็บ

Use Case Name:	แยกเนื้อหาหลักออกจากหน้าเว็บ	
Triggering Event:	เมื่อผู้ใช้ต้องการแยกเนื้อหาหลักออกจากหน้าเว็บ	
Brief Description:	เป็นการแยกเอาเฉพาะเนื้อหาหลักของหน้าเว็บนั้นโดยตัดส่วนที่ไม่เกี่ยวข้องกับเนื้อหาออก	
Actors:	Client User, Client System	
Precondition:	ผู้ใช้ต้องส่ง URL เข้าสู่ระบบ	
Post conditions:	ระบบแสดงเนื้อหาหลักของ URL ที่ส่งเข้ามา	
Flow of Events	Actor	System
	1. ส่ง URL เข้าระบบ	2. ตรวจสอบ URL 3. ดึงหน้าเว็บจาก URL 4. ประมวลผล 5. แสดงเนื้อหาหลักของหน้าเว็บ

3.4 แผนภาพ Sequence Diagram

เป็นแผนภาพแสดงขั้นตอนการทำงานของระบบตั้งแต่การรับ Input บทบาทของแต่ละส่วนไปจนถึง Output แสดงชื่อและประเภทสินค้า ดังรูปที่ 3.4



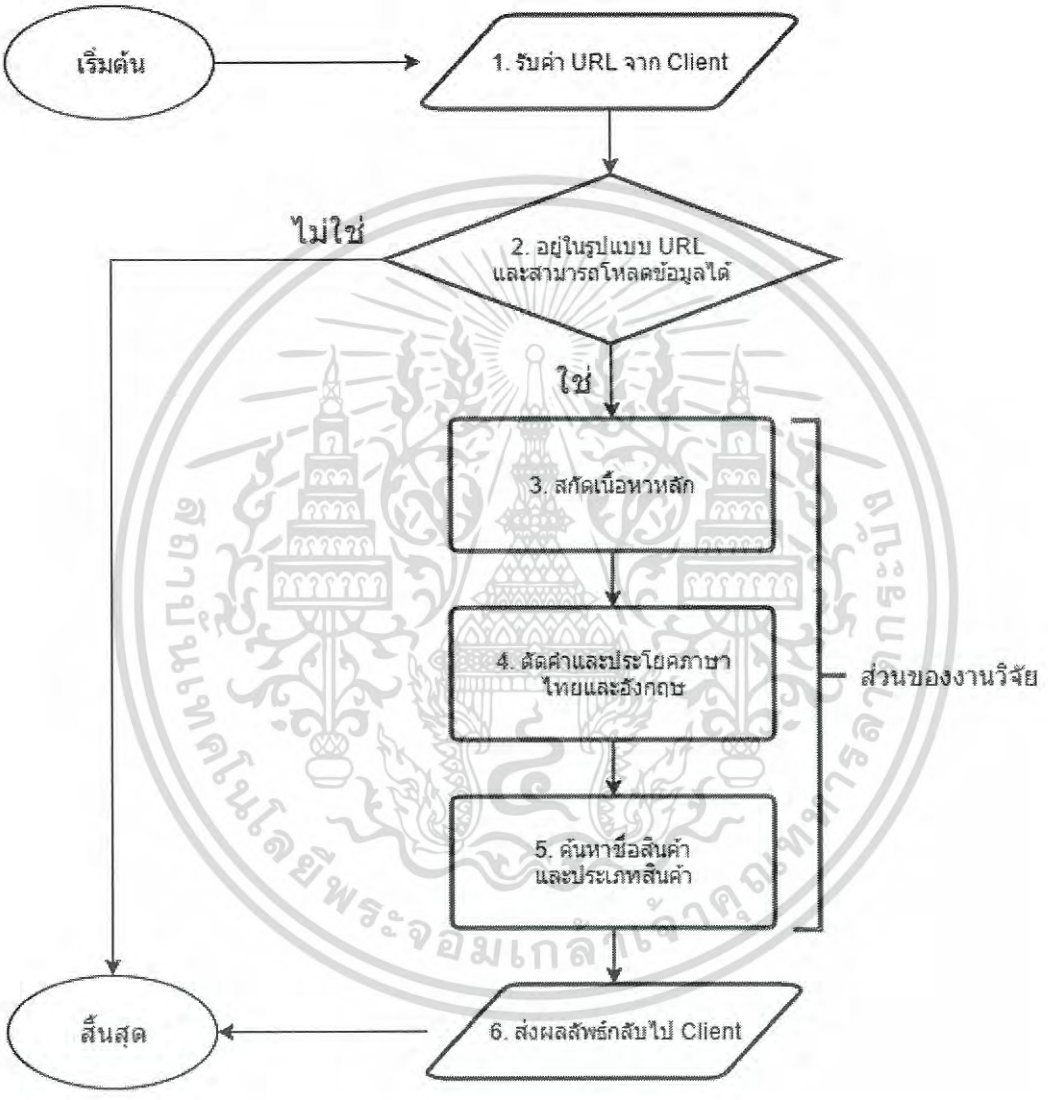
รูปที่ 3.3 Sequence Diagram ของระบบส่วนต่อประสานโปรแกรมประยุกต์เพื่อการสังเคราะห์ชื่อและประเภทสินค้าจากเว็บไซต์

3.5 ขั้นตอนการทำงานของระบบ

ดังแสดงในรูปที่ 3.4 มีขั้นตอนดังนี้

1. Web Server รับ Request Input จาก Client User หรือ Client System โดยรับ Website URL ห่อหุ้มด้วย JSON เข้าสู่ระบบ
2. ระบบ Decode JSON เป็น Plaintext URL
3. Web Scraper เช็คว่า URL นี้มีจริง แล้วเข้าไปดึงข้อมูล HTML ใน Internet
4. ระบบทำการสกัด Main Article Content จากโครงสร้าง Webpage ด้วย Boilerpipe ได้ Plaintext
5. นำข้อมูล Main Article Content มาสกัดตัดแยกคำภาษาไทย แบ่งคำเก็บในรูปแบบ list
6. นำข้อมูล list มาเข้ากระบวนการวิเคราะห์หาชื่อและประเภทสินค้า

7. กรณี Client System จะนำชื่อและประเภทสินค้าเข้ารหัสในรูปแบบภาษา JSON ส่งกลับไปให้ Client System , กรณี Client User จะนำข้อมูลชื่อและประเภทสินค้าแสดงที่หน้าเว็บ API



รูปที่ 3.4 กระบวนการดำเนินงานของระบบบริการส่วนต่อประสานโปรแกรมประยุกต์เพื่อการสังเคราะห์ชื่อและประเภทสินค้าจากเว็บไซต์

บทที่ 4

ระบบส่วนต่อประสานโปรแกรมประยุกต์ เพื่อการสังเคราะห์ชื่อและประเภทสินค้าจากเว็บไซต์

4.1 คัดเลือกโปรแกรมสำหรับแบ่งคำภาษาไทย

จากบทที่ 2 ที่กล่าวถึงโปรแกรมสำหรับแบ่งคำภาษาไทยในปัจจุบัน หลังจากทีมผู้วิจัยทดลองใช้ทั้ง 4 โปรแกรม มีเพียง Package Index ในภาษาไพทอน ที่ชื่อว่า PyICU เวอร์ชั่น 1.9.2 ที่รองรับภาษาไพทอน เวอร์ชั่น3.4 ทีมผู้วิจัยจึงเลือกโปรแกรมดังกล่าวเป็นเครื่องมือช่วยสำหรับการจัดการสตริง ในการเขียนโปรแกรมเพื่อแบ่งข้อความ String ในภาษาไทยออกจากกัน และได้ผลจากการทดลองใช้ดังนี้

ตัวอย่างที่หนึ่ง :

source : ระบบสกัดตัดคำไทย

result : ระบบ|สกัด|ตัด|คำ|ไทย

ตัวอย่างที่สอง :

source : บริการของกูเกิลเปิดแนวทางใหม่ให้ผู้ใช้ปรับค่าของเครื่องเซิร์ฟเวอร์ที่ต้องการใช้งานได้เอง โดยไม่ต้องรอประเภทเครื่องที่กูเกิลจัดมาให้

result : บริการ|ของ|กู|เกิล|เปิด|แนวทาง|ใหม่|ให้|ผู้|ใช้|ปรับ|ค่า|ของ|เครื่อง|เซิร์ฟเวอร์|ที่|ต้องการ|ใช้|งาน|ได้|เอง|โดย|ไม่|ต้อง|รอ|ประเภท|เครื่อง|ที่|กู|เกิล|จัด|มา|ให้

ตัวอย่างที่สาม :

source : แนวทางนี้ทำให้อายุการใช้งานแบตเตอรี่สำหรับรถยนต์ที่ปกติไม่ยาวนักเพราะหากแบตเตอรี่เริ่มเสื่อมจะกระทบกับระยะทางที่วิ่งได้ สามารถนำมาใช้ได้อีกนานเท่าตัวจึงจะปลดระวาง

result : แนวทาง|นี้|ทำให้|อายุ|การ|ใช้|งาน|แบตเตอรี่|สำหรับ|รถยนต์|ที่|ปกติ|ไม่|ยาว|นัก|เพราะ|หาก|แบตเตอรี่|เริ่ม|เสื่อม|จะ|กระทบ|กับ|ระยะ|ทาง|ที่|วิ่ง|ได้|สามารถ|นำ|มา|ใช้ได้|อีก|นาน|เท่า|ตัว|จึง|จะ|ปลด|ระวาง

จากผลลัพธ์ที่แสดงข้างต้นพิสูจน์ได้ว่า การแบ่งคำภาษาไทยของโปรแกรม PyICU เวอร์ชั่น 1.9.2 มีความแม่นยำในระดับที่ดีมากสามารถแบ่งคำออกถูกต้องเกือบทั้งหมด

4.2 เชื่อมต่อฐานข้อมูล MYSQL

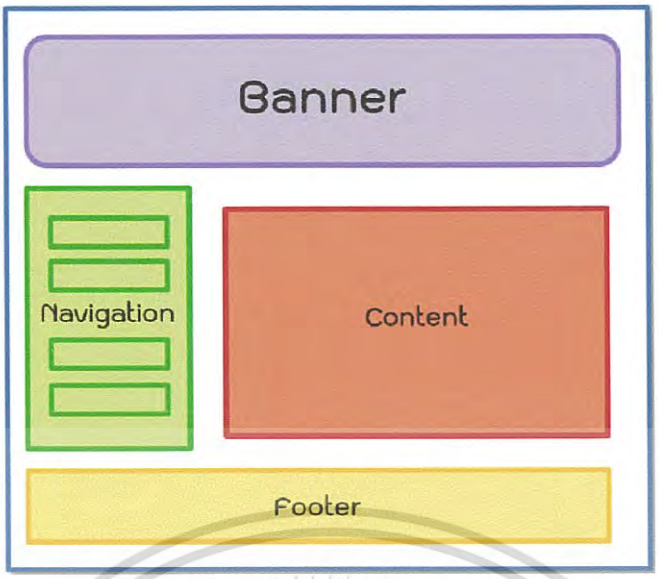
เก็บ Logs ข้อมูลไว้ทดสอบ Web API ลงใน MYSQL Database โดยใช้ Python Package Index ชื่อว่า Pymysql เป็นส่วนติดต่อกับฐานข้อมูล

```
>>> conn = pymysql.connect(host='localhost', port=3306, user='root', passwd='root', db='project' , autocommit='True')
```

4.3 การสกัดเนื้อหาหลักจากเว็บ

เป็นขั้นตอนการดึงข้อมูล HTML ด้วยโปรโตรคอล HTTP/1.1 มาเก็บไว้ในระบบ แล้วระบบจะทำการสกัดเอาใจความสำคัญหลักของหน้าเว็บออกมา ในรูปที่ 4.1 แสดงถึงส่วนประกอบพื้นฐานของหน้าเว็บ โดยทั่วไปแล้วจะประกอบด้วย

1. ส่วนหัวหรือส่วนด้านบนของเว็บ (Header) คือส่วนที่ใช้ในการแสดงผลโลโก้และเมนูหลัก
2. ส่วนเนื้อหา (Content) คือส่วนที่แสดงเนื้อหาหลัก
3. ส่วนการเชื่อมโยง (Navigation) คือส่วนที่เป็นเมนูสำหรับลิงค์ไปยังหน้าอื่น ๆ ของเว็บ จะแสดงผลอยู่ทางด้านซ้าย หรือด้านขวาของหน้าเว็บก็ได้
4. ส่วนท้ายของหน้า (Footer) คือส่วนที่จะให้ข้อมูลเพิ่มเติมเกี่ยวกับเนื้อหาและเว็บไซต์ หรืออาจเป็นที่รวมของลิงค์ที่เกี่ยวกับนโยบายทางกฎหมาย ลิขสิทธิ์ ความเป็นตัวส่วนตัว และวิธีการติดต่อกับผู้ดูแลเว็บไซต์



รูปที่ 4.1 ส่วนประกอบพื้นฐานของหน้าเว็บ [17]

ที่มา : https://sites.google.com/a/samakkhi.ac.th/kru-nitchakan-jinaprom/_

/rsrc/1439794157528/-khwam-ru-beuxng-tn-keiyw-kab-websit/form.png

ในรูปที่ 4.2 จะเป็นแสดงให้เห็นว่าบนหน้าเว็บจะมีการแบ่งพื้นที่ตามส่วนประกอบพื้นฐาน ดังนั้นขั้นตอนการสกัดเนื้อหาหลักจากเว็บจะทำการดึงเฉพาะส่วนที่เป็นใจความสำคัญหลักของเพจ (Main Article Content) คือบล็อคในกรอบสีแดง และตัดส่วนอื่นที่ไม่ใช่เนื้อหาหลักออก เช่น Menu Bar, Nav Bar, Login, Relate Content, Footer, Ads เป็นต้น

รูปที่ 4.2 : แสดงบล็อคต่าง ๆ ในหน้าเว็บทั่วไป

โดยใช้วิธีการ Boilerplate [1] จาก library ใน PyPi ชื่อว่า Boilerpipe ในการสกัด Main Article Content จากโครงสร้างเว็บเพจได้ดังตัวอย่าง

ตัวอย่างที่หนึ่ง :

Extract : "www.blognone.com/node/74971"

Result : Facebook เองนั้นเคยมีระบบเปิดให้ผู้ใช้บริจาคเงินเพื่อช่วยเหลือภัยพิบัติต่าง ๆ (เช่น เหตุการณ์น้ำท่วมในพม่า) และ Facebook เองก็คงเล็งเห็นว่าฟีเจอร์นี้จะช่วยให้ผู้ใช้ได้ตระหนักถึงภัยต่าง ๆ ที่เกิดขึ้นในโลกและเปิดช่องทางให้คนสามารถบริจาคเงินช่วยได้ง่ายกว่าเดิม ด้วยการทำหน้ารับบริจาค และมีปุ่มรับบริจาคที่ผู้ใช้สามารถบริจาคเงินเท่าไหร่ก็ได้ (ดูตัวอย่างหน้าบริจาค) หรือหากไม่บริจาคก็สามารถแชร์บอกต่อให้คนมาร่วมบริจาคด้วยก็ได้เช่นกัน ปัจจุบันนี้ Facebook ยังเปิดให้ใช้ฟีเจอร์นี้กับพาร์ตเนอร์บางส่วนเท่านั้น (เข้าใจว่าเพื่อป้องกันการเปิดรับบริจาคจากผู้ไม่ประสงค์ดี) แต่องค์กรไหนสนใจฟีเจอร์นี้ สามารถไปลงทะเบียนได้ ที่นี่

ตัวอย่างที่สอง :

Extract : "www.dek-d.com/admission/39090"

Result : ด้วยคุณสมบัติด้านกีฬาฯ วิ่ง+บาส ส่งผลให้พละกินเรียบทั้งโควตา รับตรง และแอดมิชชั่น ซึ่งคือความได้เปรียบกว่าเด็กมัธยมทั่วไป คิดเล่น ๆ หากพละคือเด็ก ม.6 จะต้องแอดมิชชั่นในปีนี้ ขณะนี้เดือน พ.ย. นักเรียน ม.6 ส่วนใหญ่กำลังรอลุ้นผลคะแนนสอบ GAT PAT อย่างใจจดใจจ่อ

สรุปผลจากการทดสอบ

จากการทดสอบการสกัดเนื้อหาหลักจากเว็บไซต์ที่เป็นภาษาไทยจำนวน 5 เว็บไซต์ โดยทำการเลือกเว็บเพจมาจำนวนทั้งหมด 20 เว็บเพจ จึงพบว่าผลลัพธ์การคำนวณค่า Precistion และ Recall เท่ากับ 0.89 และ 0.76 ตามลำดับ

ตารางที่ 4.1 ผลการทดสอบการ Extract Main Article Content ด้วย Boilerpipe

รายการทดสอบ	Precision	Recall
Extract Main Article Content	89%	76%

4.4 การหาชื่อและแบรนด์สินค้า

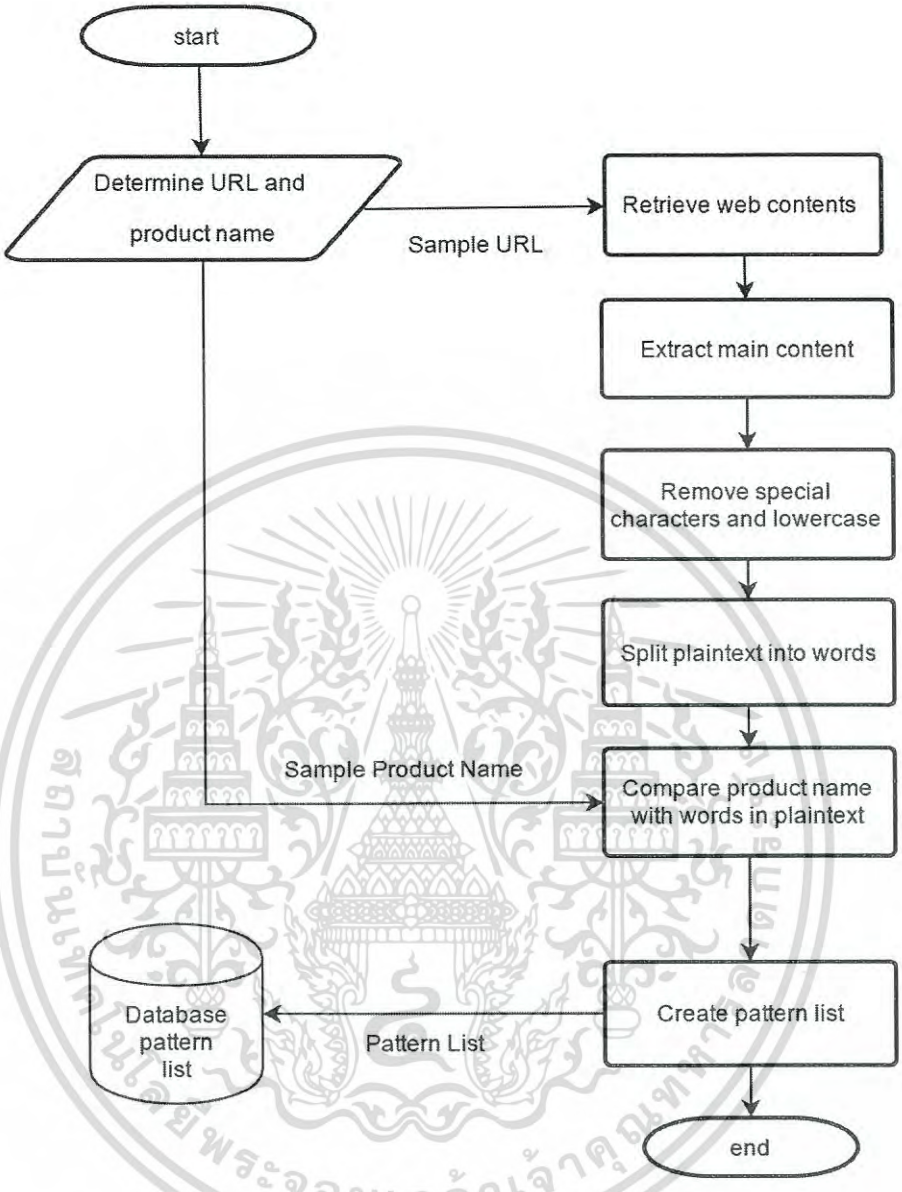
การหาชื่อและแบรนด์สินค้าเป็นวิธีการที่ทำให้รู้ว่าสินค้าที่กำลังถูกโฆษณาในแต่ละเว็บเพจคือสินค้าอะไร โดยจะแบ่งออกเป็น 2 กระบวนการ

4.4.1 การเรียนรู้รูปแบบประโยคแบบขั้นบันได (Staircase Pattern Learning)

การหารูปแบบประโยคอัตโนมัติเป็นวิธีการที่สอนวิธีคิดให้กับโปรแกรม โดยการหารูปแบบประโยคที่สามารถเชื่อมโยงไปถึงชื่อและแบรนด์สินค้าที่อยู่ในใจความสำคัญของเพจ จากนั้นก็เก็บรูปแบบประโยคเหล่านั้นไว้บนฐานข้อมูล

วิธีการนี้ประกอบด้วย 6 ขั้นตอน คือ

1. ดึงข้อมูลของแต่ละเว็บจาก URL โดยใช้โปรโตคอล http แล้วนำข้อมูลที่ได้มาเก็บไว้บนระบบ
2. สกัดใจความหลักจากเว็บเพจโดยใช้วิธีการ Boilerpipe ในการลบส่วนที่ไม่เกี่ยวข้องกับเนื้อหาหลัก เพื่อสกัดเอาแค่ใจความสำคัญของข้อมูลในหน้า html ทำให้ได้ผลลัพธ์ออกมาเป็นข้อมูลที่เป็นตัวอักษร
3. แทนที่อักษรพิเศษ ,เว้นวรรคใหญ่ ,เว้นบรรทัด และรหัสอักษรด้วยเว้นวรรคเล็ก และแปลงตัวอักษรภาษาอังกฤษทั้งหมดให้เป็นตัวพิมพ์เล็ก
4. ใช้ไลบรารี่ PyICU ในการแยกประโยคภาษาไทยให้ออกมาเป็นคำที่มีความหมายโดยมีรากฐานคำศัพท์มาจาก Regular Expression ส่วนประโยคภาษาอังกฤษจะแยกเป็นคำไว้อยู่แล้ว
5. เปรียบเทียบชื่อสินค้าในตัวอย่างกับชื่อสินค้าที่ได้จากระบบ
6. จดจำคำที่อยู่ด้านหน้าและหลังของชื่อสินค้าจำนวนอย่างละห้าคำ
7. นำคำที่ได้มาสร้างเป็นรูปแบบของประโยคแล้วเก็บไว้ในฐานข้อมูล



รูปที่ 4.3 Flow Chart การเรียนรู้รูปแบบประโยคแบบขั้นบันได (Staircase Pattern Learning)

รูปที่ 4.4 แสดงให้เห็นถึง อัลกอริทึมของการเรียนรู้รูปแบบประโยคแบบขั้นบันได (Staircase Pattern Learning)

$\sum w_i$ คือคำทุกคำที่สกัดได้จากเนื้อหลักบนเว็บ

w คือคำที่สกัดได้จากเนื้อหลักบนเว็บ

i คือตำแหน่งของคำ

w_i คือคำในตำแหน่งที่ i

sp คือตัวอย่างชื่อสินค้า

k คือค่าคงที่สำหรับกำหนดความยาวสูงสุดของคำที่ Pattern list หนึ่งอันสามารถมีได้

```
for i = 0 ; i < length of p_i ; i + +{
        n = 0;
        while(sp[n] == w_i + n){
        n = n + 1;
        if (n == length of sp)
        for(j = 0 ; j < k ; j + +){
                add (w_{i+n+j}) to temp
                add (w_{i−j−1} ) to temp
        }
        add temp to patterm list
        break;
}
```

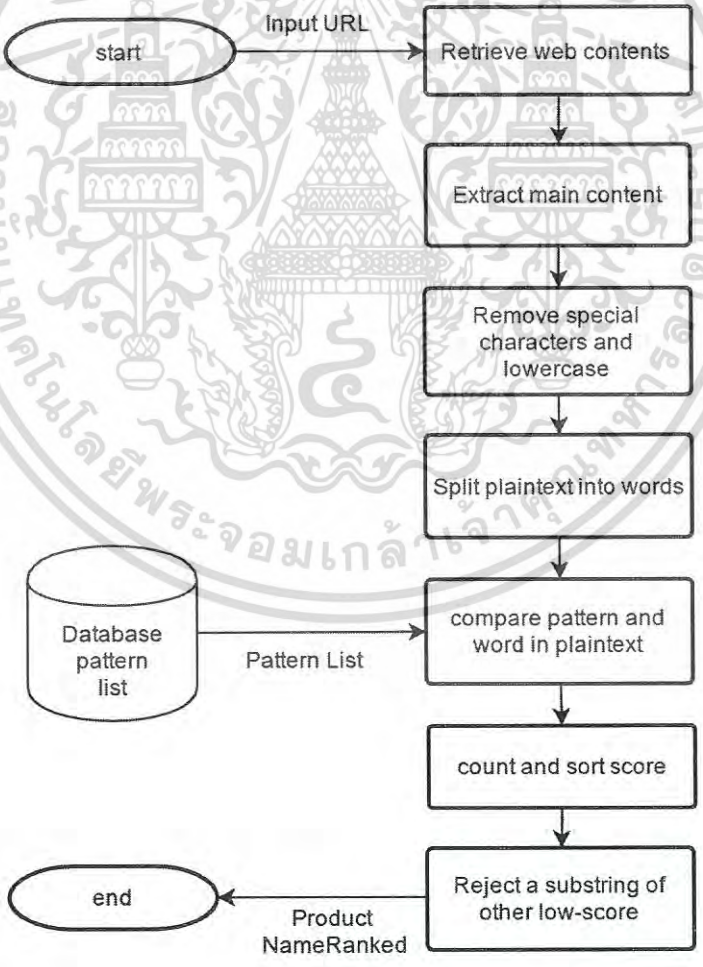
รูปที่ 4.4 อัลกอริทึมการเรียนรู้รูปแบบประโยคแบบขั้นบันได (Staircase Pattern Learning)

4.4.2 การจับคู่คำในรูปแบบขั้นบันได (Staircase Pattern Matching)

การจับคู่ของคำในรูปแบบขั้นบันได เป็นวิธีวิเคราะห์แล้วให้โอกาสที่แต่ละคำในเว็บเพจมีโอกาสที่จะเป็นชื่อสินค้าและแบรนด์ ซึ่งประกอบด้วย 7 ขั้นตอน

1. ดึงข้อมูลของแต่ละเว็บจาก URL โดยใช้โปรโตคอล http แล้วนำข้อมูลที่ได้มาเก็บไว้บนระบบ
2. สกัดใจความหลักจากเว็บเพจโดยใช้วิธีการ Boilerpipe ในการลบส่วนที่ไม่เกี่ยวข้องกับเนื้อหาหลัก เพื่อสกัดเอาแค่ใจความสำคัญของข้อมูลในหน้า html ทำให้ได้ผลลัพธ์ออกมาเป็นข้อมูลที่เป็นตัวอักษร
3. แทนที่อักษรพิเศษ ,เว้นวรรคใหญ่ ,เว้นบรรทัด และรหัสอักษรด้วยเว้นวรรคเล็ก และแปลงตัวอักษรภาษาอังกฤษทั้งหมดให้เป็นตัวพิมพ์เล็ก

4. ใช้ไลบรารี่ PyICU ในการแยกประโยคภาษาไทยให้ออกมาเป็นคำที่มีความหมายโดยมีรากฐานคำศัพท์มาจาก Regular Expression ส่วนประโยคภาษาอังกฤษจะแยกเป็นคำไว้อยู่แล้ว
5. ดึงข้อมูลรูปแบบของประโยคทั้งหมดในฐานข้อมูลมาเก็บไว้ในหน่วยความจำของระบบแล้วเทียบคำทีละคำกับข้อมูลใจความหลักของเว็บเพจหากตรงกันก็จะให้คะแนนคำๆนั้น (ขึ้นอยู่กับว่าเป็นรูปแบบประโยคแบบก่อนหรือหลังคำ) ซึ่งคะแนนดังกล่าวทำให้รู้ว่าโอกาสที่คำๆนั้นจะเป็นชื่อและแบรนด์สินค้ามากเพียงใด
6. เรียงลำดับคะแนนจากมากไปน้อย
7. ตัดคำที่มีจำนวนพยางค์น้อยกว่า 2 พยางค์ และตัดคำที่เป็น Substring ของคำที่มีคะแนนน้อยกว่า



รูปที่ 4.5 Flow Chart การจับคู่คำในรูปแบบขั้นบันได (Staircase Pattern Matching)

รูปที่ 4.6 แสดงให้เห็นถึง อัลกอริทึมของการจับคู่คำในรูปแบบขั้นบันได (Staircase Pattern Matching)

$\sum w_i$ คือคำทุกคำที่สกัดได้จากเนื้อหลักบนเว็บ

w คือคำที่สกัดได้จากเนื้อหลักบนเว็บ

i คือตำแหน่งของคำ

w_i คือคำในตำแหน่งที่ i

k คือค่าคงที่สำหรับกำหนดความยาวสูงสุดของคำที่ Pattern list หนึ่งอันสามารถมีได้

```
for ( j ; j < length of p_j ; j ++)
        (for i = 0 ; i < length of w_i ; i ++){
        n = 0;
        do{
        if (w_{i+n} != p_j[n])break;
        else{
                n = n + 1;
                if (n == length of p_j)
                        Add (w_{i+n}) to product list
        }while( n < length of p_j );
}
```

รูปที่ 4.6 อัลกอริทึมการจับคู่คำในรูปแบบขั้นบันได (Staircase Pattern Matching)

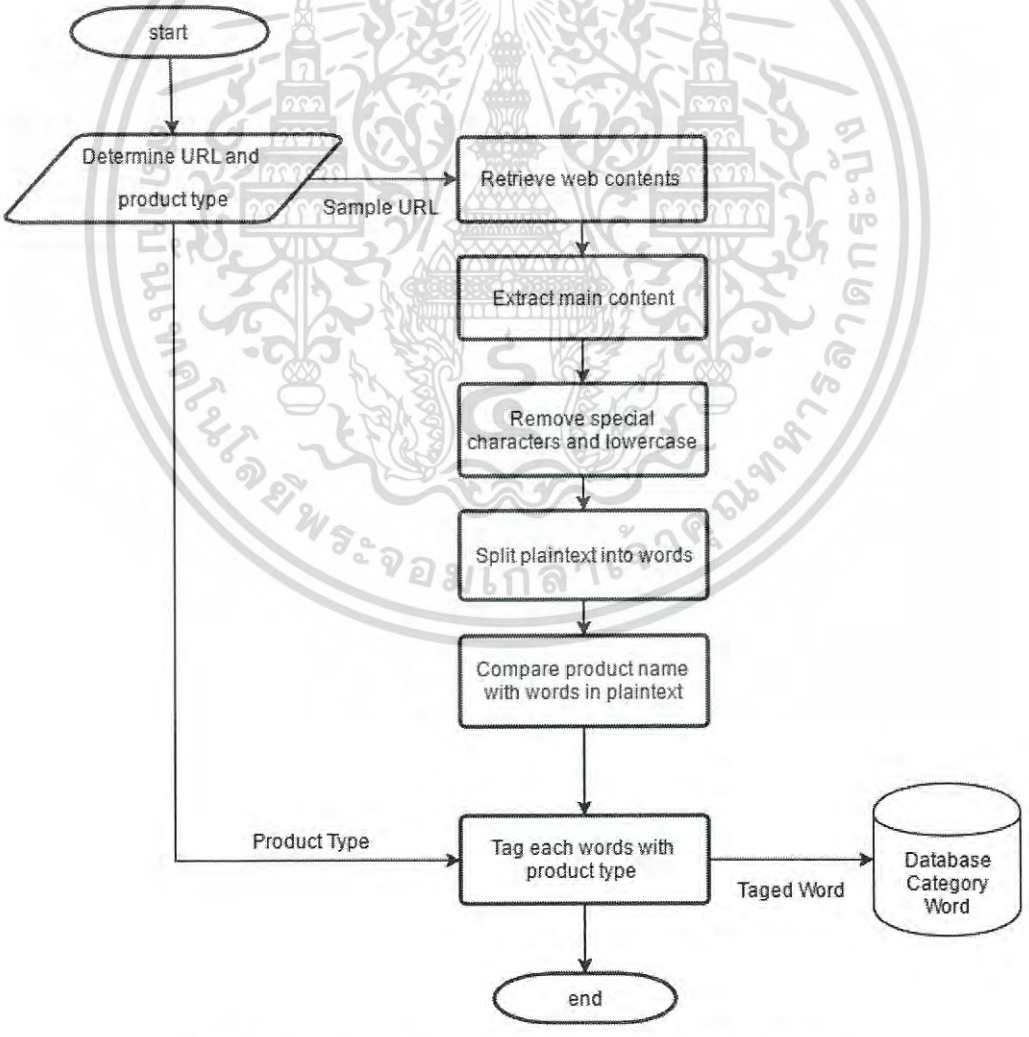
4.5 การหาประเภทสินค้า

ทีมผู้วิจัยแบ่งประเภทเว็บเพจวิจารสินค้าส่วนใหญ่ออกเป็น 4 ประเภท คือ เทคโนโลยี , เครื่องสำอาง , การท่องเที่ยว และร้านอาหาร ประเภทของเว็บเพจจะบ่งบอกถึงประเภทของสินค้าที่นำเสนอ วิธีที่เราใช้แบ่งออกเป็น 2 ขั้นตอนคือ 1. Tagging Type to Word 2. Finding Type from Tagged Word

4.5.1 Tagging Type to Word

เป็นขั้นตอนสอนโปรแกรมเรียนรู้ว่าคำต่างๆแต่ละคำที่ถูกเขียนในเว็บเพจนั้นเป็นเว็บประเภทอะไร แล้วทำการเก็บเป็นสถิติลงในฐานข้อมูล โดยมีขั้นตอนทั้งหมด 5 ขั้นตอน ดังนี้

1. ดึงข้อมูลของแต่ละเว็บจาก URL โดยใช้โปรโตคอล http แล้วนำข้อมูลที่ได้มาเก็บไว้บนระบบ
2. สกัดใจความหลักจากเว็บเพจโดยใช้วิธีการ Boilerpipe ในการลบส่วนที่ไม่เกี่ยวข้องกับเนื้อหาหลัก เพื่อสกัดเอาแค่ใจความสำคัญของข้อมูลในหน้า html ทำให้ได้ผลลัพธ์ออกมาเป็นข้อมูลที่เป็นตัวอักษร
3. แทนที่อักษรพิเศษ ,เว้นวรรคใหญ่ ,เว้นบรรทัด และรหัสอักษรด้วยเว้นวรรคเล็ก และแปลงตัวอักษรภาษาอังกฤษทั้งหมดให้เป็นตัวพิมพ์เล็ก
4. ใช้ไลบรารี่ PyICU ในการแยกประโยคภาษาไทยให้ออกมาเป็นคำที่มีความหมายโดยมีรากฐานคำศัพท์มาจาก Regular Expression ส่วนประโยคภาษาอังกฤษจะแยกเป็นคำไว้อยู่แล้ว
5. เก็บคำที่แท็กประเภทเว็บไซต์ลงฐานข้อมูล

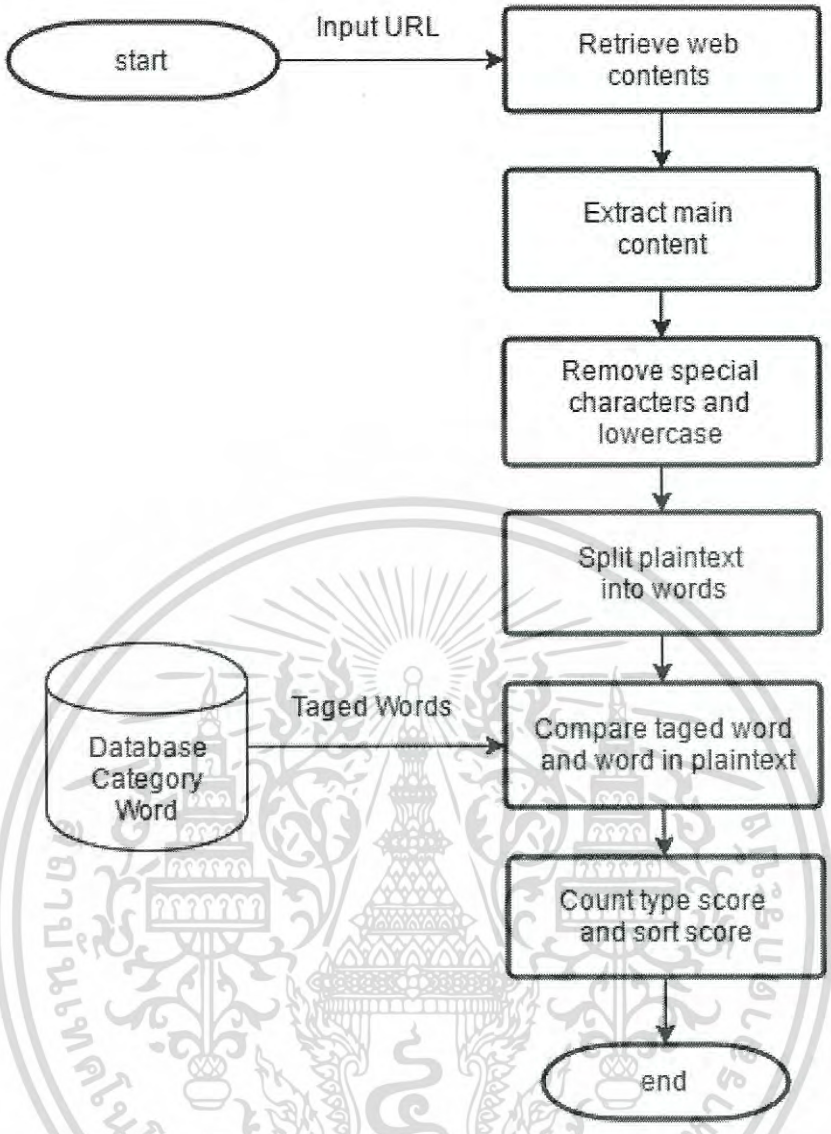


รูปที่ 4.7 Flow Chart การทำงานของ Tagging Type to Word

4.5.2 Finding Website Type from Tagged Word

เป็นขั้นตอนการหาประเภทของเว็บไซต์โดยใช้ข้อมูลของคำที่มีแท็กที่ได้บันทึกไว้จากฐานข้อมูล ประกอบด้วย 6 ขั้นตอน ดังนี้

1. ดึงข้อมูลของแต่ละเว็บจาก URL โดยใช้โปรโตคอล http แล้วนำข้อมูลที่ได้มาเก็บไว้บนระบบ
2. สกัดใจความหลักจากเว็บเพจโดยใช้วิธีการ Boilerpipe ในการลบส่วนที่ไม่เกี่ยวข้องกับเนื้อหาหลัก เพื่อสกัดเอาแค่ใจความสำคัญของข้อมูลในหน้า html ทำให้ได้ผลลัพธ์ออกมาเป็นข้อมูลที่เป็นตัวอักษร
3. แทนที่อักษรพิเศษ ,เว้นวรรคใหญ่ ,เว้นบรรทัด และรหัสอักษรด้วยเว้นวรรคเล็ก และแปลงตัวอักษรภาษาอังกฤษทั้งหมดให้เป็นตัวพิมพ์เล็ก
4. ใช้ไลบรารี่ PyICU ในการแยกประโยคภาษาไทยให้ออกมาเป็นคำที่มีความหมายโดยมีรากฐานคำศัพท์มาจาก Regular Expression ส่วนประโยคภาษาอังกฤษจะแยกเป็นคำไว้อยู่แล้ว
5. นับจำนวนคำที่มี Tag ประเภทของเว็บไซต์ภายในเว็บเพจโดยดึงคำที่ติด Tag มาจากฐานข้อมูล
6. เรียงลำดับคะแนนว่าเว็บเพจนี้ประกอบด้วยคำในประเภทไหนมากที่สุดได้ประเภทของเว็บ



รูปที่ 4.8 Flow Chart การทำงานของ Finding Website Type from Tagged Word

4.6 โครงสร้างฐานข้อมูล

4.6.1 ฐานข้อมูลรวบรวมประโยคหน้าและหลัง keyword ชื่อสินค้า

เป็นตารางฐานข้อมูลสำหรับเก็บรวบรวมข้อมูลจากกระบวนการเรียนรู้รูปแบบประโยค โดยเก็บชุดคำที่อยู่หน้าและหลังชื่อสินค้าที่เป็น keyword ทั้งหมดจำนวน 5 คำ

ตารางที่ 4.2 ตารางสำหรับเก็บชุดคำก่อน keyword

ชื่อตาราง : pattern			
Column name	Data type	Constraint	อธิบาย
no	INT(11)	PK,NN,AI	ลำดับข้อมูล
date	DATETIME	-	เวลาและวันที่ที่บันทึกข้อมูล
url	VARCHAR(255)	-	url เว็บเพจที่บันทึก
number_word	VARCHAR(20)	-	จำนวนประโยคที่ซ้ำ
word_1	VARCHAR(45)	-	คำที่ 5 นับจากด้านซ้ายของ keyword
word_2	VARCHAR(45)	-	คำที่ 4 นับจากด้านซ้ายของ keyword
word_3	VARCHAR(45)	-	คำที่ 3 นับจากด้านซ้ายของ keyword
word_4	VARCHAR(45)	-	คำที่ 2 นับจากด้านซ้ายของ keyword
word_5	VARCHAR(45)	-	คำที่ 1 นับจากด้านซ้ายของ keyword

ตัวอย่างแถวข้อมูลของตาราง pattern

1. {1172, 2016-01-28 13:26:02, 'http://www.techxcite.com/topic/24522.html', 4, 'ได้', 'รู้', 'กัน', 'แล้ว', 'กับ'}
2. {2504, 2016-03-08 16:27:58, 'http://notebookspec.com/acer-ut220hql-android-smart-monitor-review/237275/', 19, 'มุม', 'มอง', 'หน้า', 'จอ', 'ของ'}

ตารางที่ 4.3 ตารางสำหรับเก็บชุดคำหลัง keyword

ชื่อตาราง : pattern_after			
Column name	data type	constraint	อธิบาย
no	INT(11)	PK,NN,AI	ลำดับข้อมูล
date	DATETIME	-	เวลาและวันที่ที่บันทึกข้อมูล
url	VARCHAR(255)	-	url เว็บเพจที่บันทึก
number_word	VARCHAR(20)	-	จำนวนประโยคที่ซ้ำ
word_1	VARCHAR(45)	-	คำที่ 1 นับจากด้านขวาของ keyword
word_2	VARCHAR(45)	-	คำที่ 2 นับจากด้านขวาของ keyword
word_3	VARCHAR(45)	-	คำที่ 3 นับจากด้านขวาของ keyword
word_4	VARCHAR(45)	-	คำที่ 4 นับจากด้านขวาของ keyword
word_5	VARCHAR(45)	-	คำที่ 5 นับจากด้านขวาของ keyword

ตัวอย่างแถวข้อมูลของตาราง pattern_after

1. {251, 2016-01-26 15:20:11, 'http://www.techxcite.com/topic/23643.html, 23, 'นั้น', 'หน้าตา', 'ก็', 'อาจ', 'จะ'}
2. {141, 2016-01-26 15:13:37, 'http://www.beartai.com/lifestyle/movies/77591', 4, 'มี', 'กำหนด', 'ฉาย', 'ใน', 'โรง'}

4.6.2 ฐานข้อมูลรูปแบบของประโยคหน้าและหลัง keyword ชื่อสินค้าที่สรุปแล้ว

เป็นฐานข้อมูลเก็บรูปแบบประโยคจากตาราง pattern และ pattern_after ที่ผ่านกระบวนการเรียนรู้ประโยคแบบขั้นบันได (Staircase Pattern Learning) จากโปรแกรมเรียบร้อยแล้วเก็บในรูปแบบที่สามารถเอาไปคำนวณในกระบวนการ การจับคู่คำในรูปแบบขั้นบันได (Staircase Pattern Matching)

ตารางที่ 4.4 ตารางสำหรับเก็บชุดคำที่ถูกสรุปแล้ว สำหรับชุดคำก่อน keyword

ชื่อตาราง : pattern_sum_set			
Column name	data type	constraint	อธิบาย
no	INT(11)	PK,NN,AI	ลำดับข้อมูล
word_1	VARCHAR(45)	-	คำที่ 5 นับจากด้านซ้ายของ keyword
word_2	VARCHAR(45)	-	คำที่ 4 นับจากด้านซ้ายของ keyword
word_3	VARCHAR(45)	-	คำที่ 3 นับจากด้านซ้ายของ keyword
word_4	VARCHAR(45)	-	คำที่ 2 นับจากด้านซ้ายของ keyword
word_5	VARCHAR(45)	-	คำที่ 1 นับจากด้านซ้ายของ keyword
count	VARCHAR(20)	-	จำนวนรูปแบบที่ซ้ำกัน
หมายเหตุ เพิ่ม Table constraint : ADD UNIQUE INDEX `idx_name` (`word_1` ASC, `word_2` ASC, `word_3` ASC, `word_4` ASC, `word_5` ASC);			

ตัวอย่างแถวข้อมูลของตาราง pattern_sum_set

1. { 26, null, null, null, null, 'ของ', 584 }
2. { 1309, null, null, 'หน้า', 'จอ', 'ของ', 44 }
3. { 1576, null, null, 'ใช้' 'งาน' 'โปรแกรม', 10}

ตารางที่ 4.5 ตารางสำหรับเก็บชุดคำที่ถูกสรุปแล้ว สำหรับชุดคำหลัง keyword

ชื่อตาราง : pattern_a_sum_set			
Column name	data type	constraint	อธิบาย
no	INT(11)	PK,NN,AI	ลำดับข้อมูล
word_1	VARCHAR(45)	-	คำที่ 5 นับจากด้านขวาของ keyword
word_2	VARCHAR(45)	-	คำที่ 4 นับจากด้านขวาของ keyword
word_3	VARCHAR(45)	-	คำที่ 3 นับจากด้านขวาของ keyword
word_4	VARCHAR(45)	-	คำที่ 2 นับจากด้านขวาของ keyword
word_5	VARCHAR(45)	-	คำที่ 1 นับจากด้านขวาของ keyword
count	VARCHAR(20)	-	จำนวนรูปแบบที่ซ้ำกัน
หมายเหตุ เพิ่ม Table constraint : ADD UNIQUE INDEX `idx_name` (`word_1` ASC, `word_2` ASC, `word_3` ASC, `word_4` ASC, `word_5` ASC);			

ตัวอย่างแถวข้อมูลของตาราง pattern_a_sum_set

1. { 1951, null, 'นี้', 'มา', 'พร้อม', 'กับ', 18 }
2. { 821, null, null, 'ได้', 'ว่า', 'เป็น', 11}
3. { 170, null, null, null, null, 'ใช้', 28}

4.6.3 ฐานข้อมูลเก็บสถิติคำตามประเภทเว็บไซต์

เป็นฐานข้อมูลสำหรับเก็บสถิติโดยเก็บทุกคำในหน้าเว็บจากกระบวณน Tagging Type to Word โดยแท็กประเภทเว็บไปกับคำนับเก็บไว้เป็นสถิติ เพื่อนำไปใช้ต่อในกระบวนการ Finding Website Type from Tagged Word

ตารางที่ 4.6 ตารางสำหรับเก็บคำที่มีสถิติบ่งบอกประเภทของชื่อสินค้า

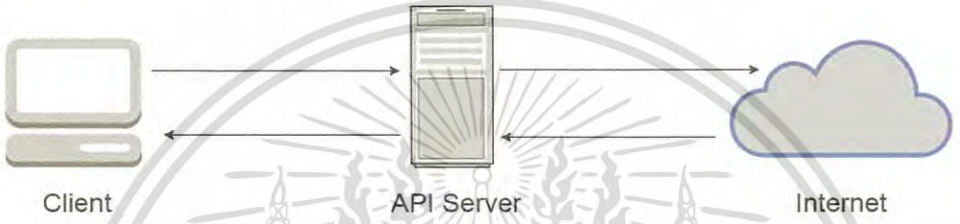
ชื่อตาราง : category_word			
Column name	data type	constraint	อธิบาย
no	INT(11)	PK,NN,AI	ลำดับข้อมูล
word	VARCHAR(200)	-	ชื่อคำที่เก็บสถิติ
it_count	VARCHAR(20)	-	จำนวนโอกาสที่พบในประเภท เทคโนโลยี
cos_count	VARCHAR(20)	-	จำนวนโอกาสที่พบในประเภท เครื่องสำอาง
travel_count	VARCHAR(20)	-	จำนวนโอกาสที่พบในประเภทการท่องเที่ยว
rest_count	VARCHAR(20)	-	จำนวนโอกาสที่พบในประเภทร้านอาหาร
total_count	VARCHAR(20)	-	จำนวนครั้งที่พบคำทั้งหมด

ตัวอย่างแถวข้อมูลของตาราง category_word

1. { 1586,'qualcomm', 1.0, 0.0, 0.0, 0.0, 55 }
2. { 467, 'blush', 0.0, 1.0, 0.0, 0.0, 28 }
3. { 6233, 'สำรวจ', 0.17, 0.0, 0.56, 0.28, 18 }
4. { 736, 'เที่ยว', 0.02, 0.02, 0.29, 0.68, 256}

4.7 โครงสร้างของระบบ Web Server

Web Server สำหรับให้บริการระบบส่วนต่อประสานโปรแกรมประยุกต์เพื่อการสังเคราะห์ชื่อและประเภทสินค้าจากเว็บไซต์วิจารณ์สินค้า เป็นเทคโนโลยี VMware ที่ติดตั้งใน Server หลักของคณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าลาดกระบัง มี Domain Name = http://it-kmitl.ml โดยติดตั้ง Django: Web framework เป็น Web Service ทำงานบนพื้นฐานของภาษาไพทอน ให้บริการรับ Request มาจากอุปกรณ์ Client ด้วยภาษา JSON เป็นภาษากลางในการติดต่อ



รูปที่ 4.9 แสดงลำดับชั้นของ Server , API , Client

4.8 หน้าจอแสดงผลลัพธ์สำหรับผู้ใช้ API

หลังจากที่ผู้ใช้ป้อน URL ที่ต้องการลงไป ระบบจะทำการดึงข้อมูลเว็บมาวิเคราะห์แล้วแสดงผลลัพธ์ผ่านเว็บเบราว์เซอร์ของผู้ใช้ โดยข้อมูลที่แสดงประกอบด้วย 1. URL ที่วิเคราะห์ 2. ประเภทของสินค้า 3. จำนวนชื่อสินค้าที่หาเจอ 4. จำนวนคะแนนรวมทั้งหมด 5. เวลาที่ระบบทำงาน 6. ตารางอันดับคำที่ได้คะแนนโอกาสเป็นชื่อสินค้ามากที่สุด 5 อันดับ

API for Extracting Brand Name and Product Name

API Analysis from URL http://review.thaiware.com/751.html EXTRACT

Information

Product type : technology
Total product extracted : 18
Total score : 94
Calculating time 0.15 second

Total of top-five product name

Rank	Result Product Name	Score	Percent
1	dell venue 11 pro	49	52%
2	venue 11 pro	9	9%
3	dell venue	7	7%
4	microsd slot	3	3%
5	intel hd 5300	2	2%

รูปที่ 4.10 หน้าจอแสดงผลลัพธ์ทางเว็บเบราว์เซอร์

บทที่ 5

ผลการดำเนินงาน การวิเคราะห์และสรุปผล

5.1 ผลการทดลอง

ข้อมูลที่ใช้ในการทดลองคือบทความรีวิวสินค้าและบริการบนเว็บไซต์มีชื่อเสียงในประเทศไทยที่ถูกเขียนขึ้นตั้งแต่วันที่ 1 ม.ค. 2558 ถึง 30 มี.ค. 2559 โดยจะแยกข้อมูลออกเป็น 2 ชุดด้วยกัน ข้อมูลชุดแรกสำหรับหาชื่อสินค้า จะมีเว็บสินค้าและบริการปะปนกันจำนวน 960 เว็บเพจ และข้อมูลชุดที่สองสำหรับหาประเภทสินค้าจะมีการแยกหมวดหมู่เว็บเพจไว้ทั้งหมด 4 หมวดหมู่ จำนวนหมวดหมู่ละ 150 เว็บเพจ รวมเป็นจำนวน 450 เว็บเพจ แยกออกเป็นสินค้า 2 หมวดหมู่และบริการ 2 หมวดหมู่ หมวดหมู่สินค้า ประกอบด้วย 1. สินค้าในหมวดหมู่เทคโนโลยี คือ โน๊ตบุ๊ค , โทรศัพท์มือถือ ,กล้องถ่ายรูป ,แก็ดเจ็ต 2. สินค้าในหมวดหมู่เครื่องสำอางค์ หมวดหมู่บริการประกอบด้วย 1. ร้านอาหารและคาเฟ่ 2. ที่พัก คือ โรงแรมขนาดเล็ก (Hostel) ,โรงแรมขนาดใหญ่ (Hotel) ,รีสอร์ท ซึ่งสินค้าและบริการที่เลือกมาในหมวดหมู่ คือสินค้าและบริการที่ผู้บริโภคมักจะดูจากเว็บไซต์วิจารณ์หรือรีวิวก่อนทำการตัดสินใจซื้อหรือเข้ารับบริการ

ทีมผู้วิจัยได้ใช้ค่า Precision เป็นตัวชี้วัดค่าความถูกต้องของข้อมูล โดยค่า Precision นี้คำนวณได้จากสมการ

$$Precision = \frac{Number\ of\ correctly\ extracted\ webpages}{Number\ of\ correctly\ extracted\ webpages + number\ of\ incorrectly\ extracted\ webpages} \quad (5.1)$$

5.1.1 การหาความถูกต้องของการสกัดชื่อและแบรนด์สินค้า

ใช้ข้อมูลชุดแรกในการทดสอบ โดยจะแยกข้อมูลเพื่อนำมาใช้สอนวิธีคิดการค้นหาชื่อสินค้าให้กับโปรแกรม 63% คิดเป็นจำนวน 602 เว็บเพจ สำหรับการทดสอบความถูกต้อง 37% คิดเป็นจำนวน 358 เว็บเพจ โปรแกรมจะเรียนรู้จากค่าความถูกต้องของรูปแบบประโยคจากเนื้อหาหลักของเว็บไซต์วิจารณ์สินค้า ค่าความถูกต้องคือค่าที่ได้จากการเปรียบเทียบชื่อสินค้าที่แท้จริงกับชื่อสินค้าที่ได้จากการคำณวน และอัตราส่วนความถูกต้องของชื่อสินค้าจะถูกนำมาใช้ในการพัฒนาและใช้ในการทดสอบวัดผล

ผลการทดลองแสดงในตาราง 5.1 คือผลการทดลองการหาความถูกต้องของการสกัดชื่อและแบรนด์สินค้า ด้วยวิธีการเรียนรู้รูปแบบประโยคแบบขั้นบันได (Staircase Pattern Learning) และวิธีการจับคู่คำในรูปแบบขั้นบันได (Staircase pattern matching)

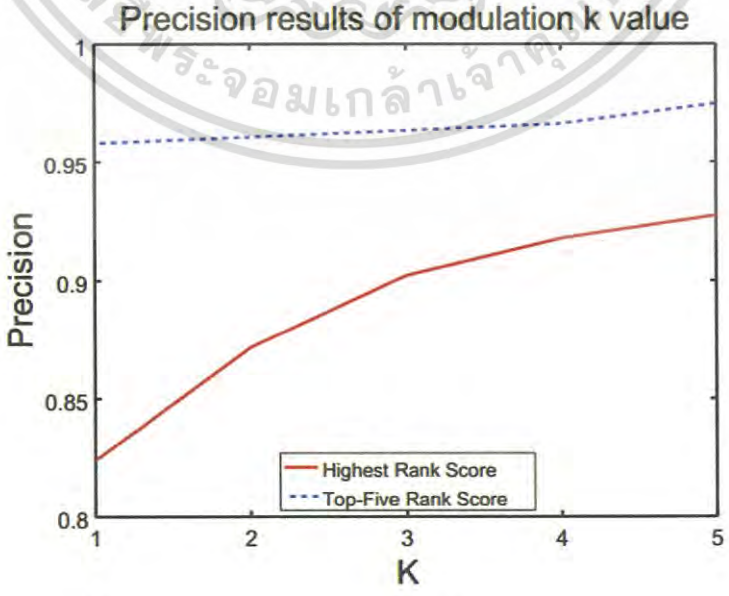
ตารางที่ 5.1 ผลการทดสอบการสกัดชื่อและแบรนด์สินค้า

Precision Type	Highest Rank Score	Top-Five Rank Score
Number of webpages	358	358
Number of correctly extracted webpages	332	349
Number of incorrectly extracted webpages	26	9
Precision (%)	92.73%	97.48%

ในรูปที่ 5.1 เป็นการแสดงค่าความถูกต้องที่เปลี่ยนแปลงไปหลังจากเปลี่ยนค่า K

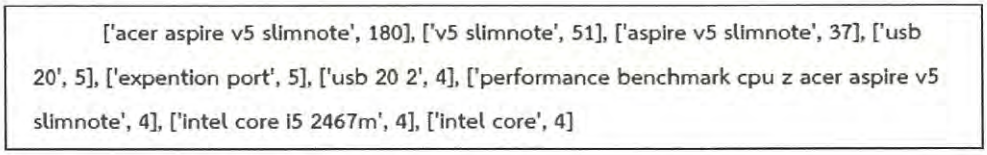
K คือค่าคงที่ สำหรับกำหนดความยาวของคำที่ Pattern list หนึ่งอันสามารถมีได้ ในกระบวนการของการเรียนรู้รูปแบบประโยคแบบขั้นบันได

ถ้า Pattern list มีจำนวนคำมาก จะยิ่งทำให้เกิดความแตกต่างระหว่าง Pattern อื่นมากขึ้น ซึ่งความแตกต่างเป็นสิ่งสำคัญที่ทำให้รู้ว่าคำที่โปรแกรมหามาได้นั่น ใช่ชื่อสินค้าหรือไม่ ถ้า K มีค่ามากจะทำให้ผลลัพธ์ที่ได้ออกมาดีขึ้น แต่ก็ใช้เวลาในการประมวลผลมากขึ้นตาม จากการทดลองทีมผู้วิจัยจึงพบว่า ค่า K ที่เหมาะสมที่สุดคือ 5 นั่นเอง



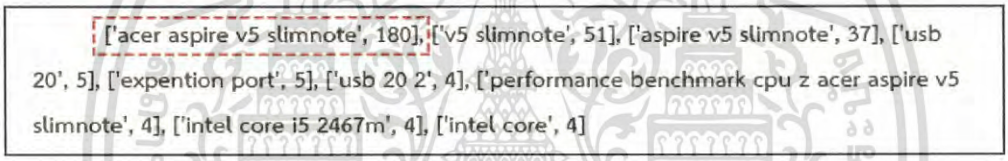
รูปที่ 5.1 ค่าความถูกต้องจากการเปลี่ยนแปลงของค่า K

รูปที่ 5.2 แสดงถึงตัวอย่างผลลัพธ์การสกัดชื่อและแบรนด์สินค้าด้วยวิธีการจับคู่คำในรูปแบบขั้นบันได (Staircase pattern matching) "Acer Aspire V5 slimnote" คือชื่อและแบรนด์สินค้าที่โปรแกรมหาออกมา ส่วน "180" คือคะแนนที่บอกถึงโอกาสที่คำๆนั้นจะเป็นชื่อและแบรนด์สินค้า



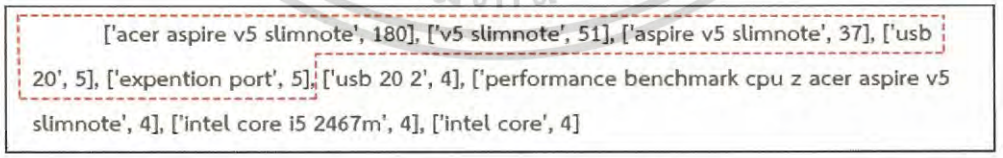
รูปที่ 5.2 ตัวอย่างผลลัพธ์การสกัดชื่อและแบรนด์สินค้าด้วยวิธีการจับคู่คำในรูปแบบขั้นบันได (Staircase pattern matching)

รูปที่ 5.3 จะเห็นได้ว่า "Acer Aspire V5 Slimnote" มีคะแนนของโอกาสที่จะเป็นชื่อสินค้ามากที่สุด โปรแกรมก็จะทำการเปรียบเทียบชื่อดังกล่าวกับชื่อสินค้าจริงในตัวอย่างข้อมูลที่หาไว้ในตอนแรกหรือไม่ หากตรงกันก็จะจัด URL นี้ไว้ในประเภท top rank score



รูปที่ 5.3 ค่าที่สูงที่สุดจากตัวอย่างผลลัพธ์การสกัดชื่อและแบรนด์สินค้าด้วยวิธีการจับคู่คำในรูปแบบขั้นบันได (Staircase pattern matching)

ถ้าชื่อสินค้าที่มีคะแนนสูงสุดไม่ตรงกับชื่อสินค้าจริง แต่ชื่อสินค้าที่มีคะแนน 5 อันดับแรกยังตรงกับชื่อสินค้าจริง ก็จะจัด URL นี้ไว้ในประเภท top- five rank scored



รูปที่ 5.4 ค่าที่สูงที่สุด 5 อันดับแรกจากตัวอย่างผลลัพธ์การสกัดชื่อและแบรนด์สินค้าด้วยวิธีการจับคู่คำในรูปแบบขั้นบันได (Staircase pattern matching)

5.1.2 การหาความถูกต้องของการสกัดประเภทสินค้า

เราได้ใช้ข้อมูลชุดสองสำหรับการทดสอบ โดยจะแยกข้อมูลเพื่อนำมาใช้สอนวิธีจำแนกประเภทให้กับโปรแกรม 52% คิดเป็นจำนวน 450 เว็บเพจ และสำหรับการทดสอบความถูกต้อง 48% คิดเป็นจำนวน 411 เว็บเพจ โปรแกรมจะเรียนรู้จากคำที่ถูกใช้ในเว็บแต่ละประเภท เมื่อทดสอบแล้วจะได้ค่าความถูกต้องสำหรับวัดผล โดยค่าความถูกต้องมาจาก จำนวนเว็บไซต์ที่โปรแกรมวิเคราะห์ออกมาได้ถูกต้องต่อจำนวนเว็บไซต์ที่วิเคราะห์ทั้งหมด ผลการทดลองดังกล่าวแสดงในตาราง 5.2

ตารางที่ 5.2 ผลการทดสอบการสกัดประเภทสินค้า

Precision Type	Type Equal
Number of webpages	411
Number of correctly extracted webpages	358
Number of incorrectly extracted webpages	53
Precision (%)	87.10%

5.2 สรุปผลการวิจัยและดำเนินงาน

จากการศึกษาโครงสร้างของเว็บไซต์ในประเทศไทยตลอดระยะเวลาที่ผ่านมา ทำให้ทีมผู้วิจัยสังเกตุได้ว่าเว็บไซต์วิจารณ์สินค้า มีการวางโครงสร้างของบล็อกที่บรรจุข้อมูลเนื้อหาสำคัญของเพจเพียงบล็อกเดียวต่อหนึ่งหน้าเว็บเพจ โดยภายในมีข้อมูลของสินค้า รูปภาพสินค้า คำอธิบายสินค้า เพื่อนำเสนอแก่ผู้เข้าชมเว็บไซต์

จากการศึกษาวิธีการค้นหาชื่อและประเภทสินค้าจากประโยคเนื้อหาในภาษาไทยนั้นมีความยากต่อการค้นหา เพราะว่าภาษาไทยมีรูปแบบการเขียนประโยคที่ติดกัน อ่านออกเสียงได้หลายแบบ มีกฎการเขียนที่ละเอียดอ่อน ทีมผู้วิจัยจึงต้องแบ่งคำภาษาไทยออกจากประโยค ด้วยวิธีการและเทคนิคต่าง ๆ ที่ผู้ทำวิจัยทั่วประเทศไทยคิดค้นขึ้นมา

ในส่วนของการทดลองวิธีการที่ใช้หาชื่อและแบรนด์สินค้าจากเนื้อหาหลักบนเว็บไซต์วิจารณ์สินค้าที่เขียนในรูปแบบภาษาภาษาไทยและภาษาอังกฤษด้วยวิธีการเรียนรู้รูปแบบประโยคแบบขั้นบันได(Staircase Pattern Learning) และวิธีการจับคู่คำในรูปแบบขั้นบันได (Staircase pattern matching) สามารถหาชื่อและแบรนด์สินค้าได้อย่างแม่นยำ แต่มีข้อจำกัดในการคำนวณ คือ

1.) เว็บไซต์วิจารณ์สินค้าจะต้องมีชื่อและแบรนด์สินค้าอยู่ในส่วนของเนื้อหาหลักของเว็บ 2.) ชื่อและแบรนด์สินค้าจะต้องเป็นภาษาอังกฤษและเนื้อหาหลักจะต้องเป็นภาษาไทยเท่านั้น

ส่วนการทดลองแบ่งประเภทสินค้า จะมี 2 ขั้นตอนคือ การเรียนรู้ประเภทของสินค้ากับคำศัพท์จากเว็บเพจที่กล่าวถึงสินค้า (Tagging Type to Word) และวิธีการหาประเภทสินค้าจากคำศัพท์ในฐานข้อมูล(Finding Website Type from Tagged Word) ก็ให้ผลลัพธ์ในการแบ่งประเภทที่ค่อนข้างแม่นยำ โดยวิธีการนี้จะมีประสิทธิภาพเพิ่มมากขึ้นตามจำนวนเว็บเพจที่นำมาสอนโปรแกรม

5.3 ปัญหาและอุปสรรค

ทีมผู้วิจัยใช้เครื่องมือเพื่อศึกษาด้วยภาษาไพทอนที่เป็นภาษาโปรแกรมระดับสูงเพื่อใช้งานทั่วไป โดยเลือกใช้โปรแกรมไพทอน เวอร์ชั่น 3.4 โดยปัญหาที่พบคือ ภาษาไพทอนที่นิยมใช้ในปัจจุบัน มี 2 เวอร์ชั่น คือ เวอร์ชั่น 2.7 และ เวอร์ชั่น 3.4 ซึ่งทั้ง 2 เวอร์ชั่นจะมีรูปแบบการเขียนที่แตกต่างกัน ทำให้ไม่สามารถใช้งานร่วมกันได้ และมีปัญหาที่ไม่สามรถนำไลบรารี่ที่ถูกพัฒนาไว้ในเวอร์ชั่นเก่ามาใช้ได้ ปัญหาของ Microsoft Visual C++ 2010 สำหรับติดตั้งไลบรารี่ส่วนใหญ่ของภาษาไพทอนจากแหล่งรวมไลบรารี่ PyPI มีบัคสำหรับผู้ใช้งาน Window8/8.1/10 ทำให้ไม่สามารถติดตั้งไลบรารี่ที่ต้องการได้

ปัญหาของภาษาไทยเมื่อใช้งานไลบรารี่ต่าง ๆ ที่ถูกพัฒนาในภาษาอังกฤษ ทำให้โปรแกรมไม่สามารถใส่ค่า Input ที่เป็นภาษาไทยได้ จึงทำให้ประมวลผลออกมาได้ผลลัพธ์ที่ไม่ถูกต้องแม่นยำ จึงไม่สามารถนำไปใช้งานได้ เช่น Unicode จะไม่รองรับ Unicode cp-874 ซึ่งเป็น Unicode ของภาษาไทย

5.4 งานวิจัยในอนาคต

วิธีการเรียนรู้รูปแบบประโยคแบบขั้นบันได (Staircase Pattern Learning) เป็นวิธีสำหรับหารูปแบบของภาษาท้องถิ่นในการพูดถึงชื่อสินค้า ถ้าหากมีจำนวนเว็บไซต์ที่มากขึ้นและมีความหลากหลายเพียงพอ ก็จะสามารถหารูปแบบของประโยคที่สามารถเชื่อมโยงไปถึงที่ถูกใช้ในเว็บไซต์ชนิดอื่น เช่น เว็บข่าวสินค้า, เว็บขายสินออนไลน์, เว็บไซต์สังคมออนไลน์

ในอนาคตหากมีข้อมูลเว็บไซต์ของคนส่วนมากใช้งาน เช่น จากระบบเก็บ Logs ในองค์กร ก็จะสามารถนำ Logs เหล่านั้นมาวิเคราะห์เพื่อหาว่าในขณะนี้คนส่วนใหญ่ในองค์กรกำลังสนใจสินค้าประเภทไหน ชื่อและแบรนอะไร เพื่อนำข้อมูลที่ได้นี้ไปใช้ประโยชน์ได้ต่อไปเช่น ทำการตลาดภายในองค์กร

บรรณานุกรม

[1] Christian Kohlschütter, Peter Fankhauser, Wolfgang Nejdl "**Boilerplate Detection using Shallow Text Features**" in Proc. WSDM 10, 2010

[2] อดุลย์ ยิ้มงาม. "**การทำเหมืองข้อมูล Data Mining**"[Online]. Available: http://compcenter.bu.ac.th/news-information/data-mining. 2011

[3] Ir R. Kosala , Ac. Be , R. Kosala , Hendrik Blockeel , Frank Neven. **"Web Mining"** [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=FD09B0D2ECF538BE512E214E08C632C9?doi=10.1.1.12.3256&rep=rep1&type=pdf.2000

[4] nmcxpress in Data Mining. **"Web Mining"** [Online]. Available: https://nmcxpress.wordpress.com/2012/03/11/web-mining/. 2012

[5] เกร็ดความรู้.net. **"API คืออะไร ทำหน้าที่อะไร ประโยชน์ของ API มีอะไรบ้าง"** [Online]. Available: http://www.xn--12cg1cxchd0a2gzc1c5d5a.net/api/. 2016

[6] Human Language Technology Laboratory , National Electronics and Computer Technology Center.**"แหล่งรวบรวมความรู้"** [Online]. Available: http://thailang.nectec.or.th/best/?q=node/4 .2016

[7] Admin. **"MySQL คืออะไร?"** [Online]. Available: http://www.phpdevthailand.com/archive/19/MySQL_%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3?. 2012

[8] Wikipedia. **"File:MySQL.svg"** [Online]. Available: https://en.wikipedia.org/wiki/File:MySQL.svg. 2008

[9] admin. **"Unix Sockets and PyMySQL"** [Online]. Available: http://www.eceforge.com/wp-content/uploads/2012/07/sql_plus_python.png. 2012

[10] Treselle Engineering. **"Boilerpipe – Web Content Extraction without Boilerplates"** [Online] . Available: http://www.treselle.com/blog/boilerpipe-web-content-extraction-without-boiler-plates/. 2014

บรรณานุกรม (ต่อ)

[11] MawtoSoftware. **"VMware Workstation Pro 12.1.1 [Full + Key] One2up พร้อมวิธีใช้โปรแกรมจำลองวินโดว์ Apr2016"** [Online]. Available: https://www.mawtoload.com/vmware-workstation-pro-12-full-key-one2up/. 2016

[12] coreacademy. **"logo_winserver2012R2"** [Online]. Available: http://coreacademy.com.pk/ca/wp-content/uploads/2016/01/logo_winserver2012R2-1-200x182.png. 2012

[13] Bharat Singh, Ishadutta Yadav, Suneeta Agarwal, Rajesh Prasad. **"An Efficient Word Searching Algorithm through Splitting and Hashing the Offline Text"** Ph.D. Thesis Of Computer Science & Engineering Motilal Nehru National Institute of Technology Allahabad-211004, INDIA. 2009

[14] Jayendra Barua, Dhaval Patel, Ankur Kumar Agrawal. **"Removing Noise Content from Online News Articles "** Computer Society of India Mumbai, India, India. 2014

[15] W. Choochaiwattana, "**An algorithm of product information extraction from web pages: a document object model Analysis approach**" in Proc. ICICM 2012, 2012

[16] B. Mehta and M. Narvekar, **"DOM tree based approach for web content extraction"** in Proc.ICCICT, 2015

[17] Nitchakan Jinaprom. **"องค์ประกอบของเว็บเพจ ขนาด 1024x768 pixels"** [Online]. Available: https://sites.google.com/a/samakkhi.ac.th/kru-nitchakan-jinaprom/_/rsrc/1439794157528/-khwam-ru-beuxng-tn-keiyw-kab-websit/form.png. 2015

[18] COUCHBASE. **"Sub-Document API"** [Online]. Available: http://images.cbauthx.com/server/4.1/20160520-233733/sub-doc-api-1.png. 2016

[19] Open source license, **"User Guide, ICU Architectural Design, ICU API compatibility"** [Online]. Available: www.userguide.icu-project.org/design#TOC-ICU-API-compatibility, 2009.

ภาคผนวก

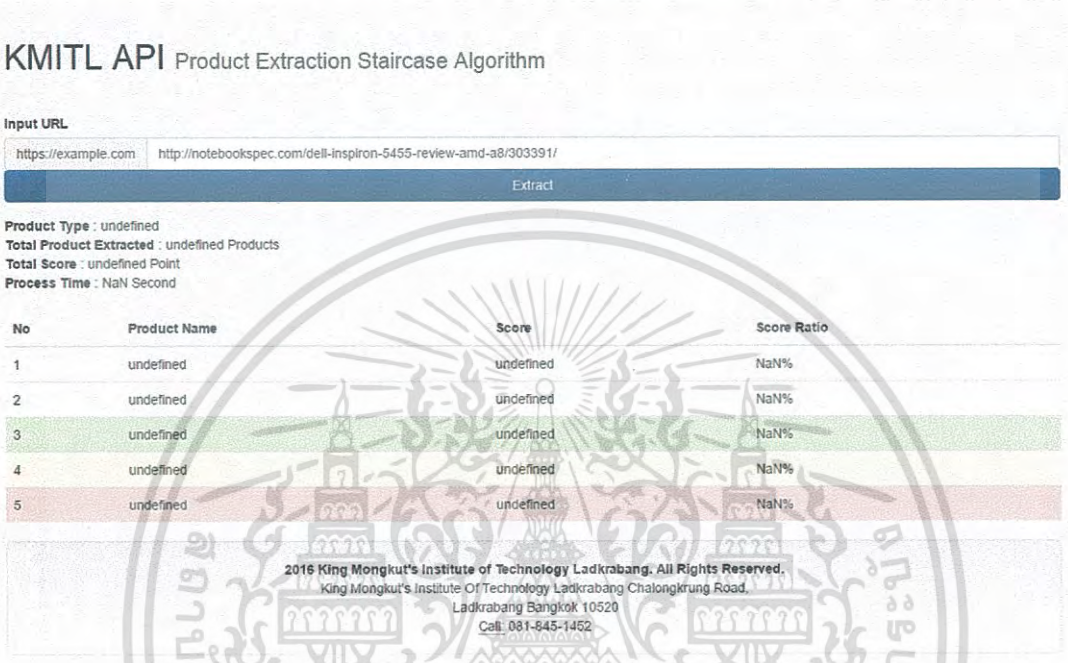
ภาคผนวก ก

คู่มือการใช้งานระบบ

คู่มือการใช้งานระบบ

การใช้งาน API

1. เข้า Web Browser ไปที่ http://161.246.38.222:8000/index/ จะแสดงหน้าระบบดังรูปที่ ก.1



รูปที่ ก.1 หน้าระบบ API

2. ใส่ URL ของเว็บเพจที่ต้องการจะค้นหาชื่อและประเภทสินค้า

รูปที่ ก.2 Input Text

3. กดปุ่ม Extract เพื่อสกัดชื่อและประเภทสินค้า

รูปที่ ก.3 ปุ่ม Extract

4. ผลลัพธ์ในการสกัดหาชื่อและประเภทสินค้าจะแสดงดังรูปที่ ก.4

Product Type คือ ประเภทสินค้าบนเว็บเพจ

Total Product Extract คือ จำนวนชื่อสินค้าทั้งหมดที่หาได้จากเว็บเพจ

Total Score คือ คะแนนรวมของชื่อสินค้าที่เป็นไปได้

Process Time คือ เวลาในการประมวลผล

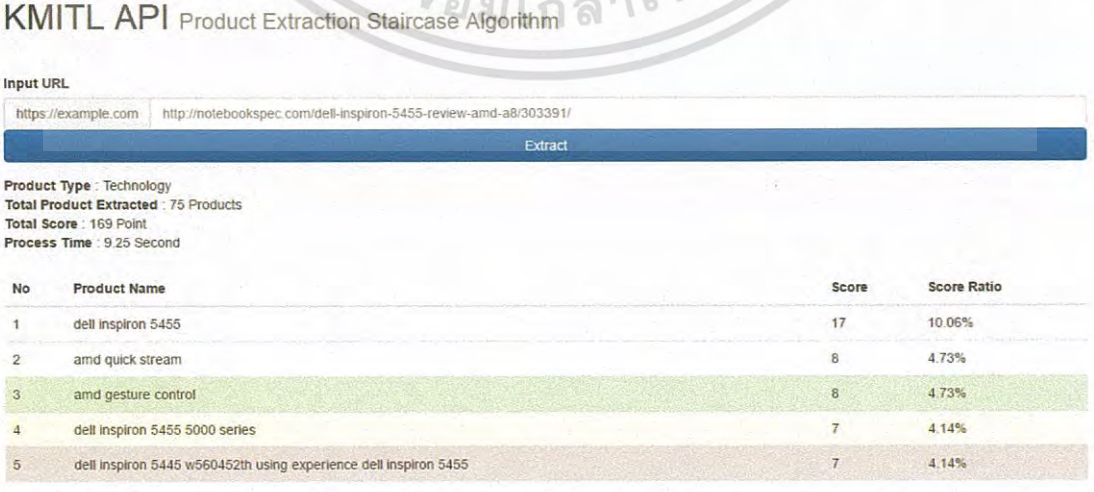
ในส่วนของตารางจะแสดงผลลัพธ์ของชื่อสินค้าที่เป็นไปได้ 5 อันดับแรก

No คือ ลำดับของชื่อสินค้า

Product Name คือ ชื่อสินค้าที่สกัดออกมา

Score คือ คะแนนความเป็นไปได้ของชื่อสินค้า

Score Ratio คือ อัตราส่วนของคะแนนความเป็นไปได้ของชื่อสินค้าจากเว็บเพจ

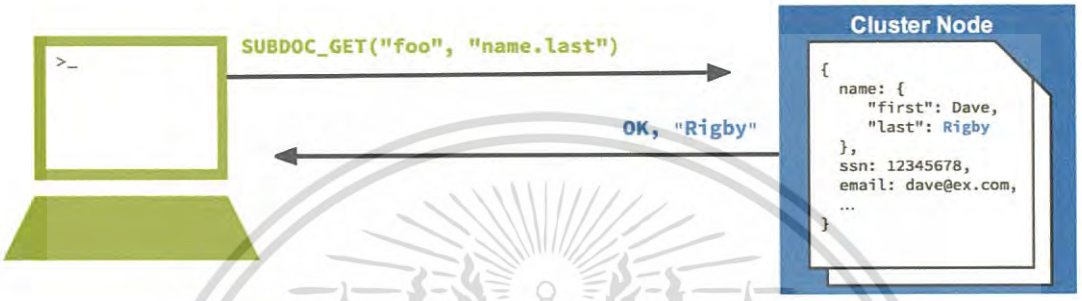


รูปที่ ก.4 ส่วนแสดงผลลัพธ์การสกัดชื่อและประเภทสินค้า

วิธีการใช้งานผ่าน JSON API

1. ส่ง JSON Request มาที่ http://161.246.38.222:8000/APIProcess/ โดยใส่ Parameter Value ชื่อ "url" เข้ามา 1 ค่า และเข้ารหัสให้อยู่ในรูปแบบของภาษา JSON

2. รอรับ JSON จาก Server



รูปที่ ก.5 กระบวนการทำงานของ Web API [18]

ที่มา : http://images.cbauthx.com/server/4.1/20160520-233733/sub-doc-api-1.png

ประวัติผู้เขียน

ชื่อ – นามสกุล	นางสาววิชญาน์นันท์ จิรบวรวณิชย์
วัน เดือน ปีเกิด	30 เมษายน 2537 ที่กรุงเทพมหานคร
ที่อยู่	868/216 ซอยวานิช 2 ถนนทรงวาด แขวงตลาดน้อย เขตสัมพันธวงศ์ จังหวัดกรุงเทพมหานคร 10100
ประวัติการศึกษา	2558 วิทยาศาสตรบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
อีเมล	Vichayanan_Jir@hotmail.com

ชื่อ – นามสกุล	นายศักดา จันทรรวงทอง
วัน เดือน ปีเกิด	19 มิถุนายน 2536 ที่เชียงใหม่
ที่อยู่	85/194 หมู่บ้านนัทวัน ถนนอุทยานอักษะ แขวงศาลายา เขตพุทธมณฑล จังหวัดนครปฐม 73170
ประวัติการศึกษา	2558 วิทยาศาสตรบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
อีเมล	Sakda_jan@hotmail.com

ระบบบริการส่วนต่อประสานโปรแกรมประยุกต์เพื่อสังเคราะห์ชื่อและประเภทสินค้าจากเว็บไซต์วิจารณ์สินค้า

วิชญาน์นันท์ จิรบวรวณิชย์ และ ศักดา จันทรรวงทอง

คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

Emails: s5070107@kmitl.ac.th, s5070116@kmitl.ac.th

บทคัดย่อ

เว็บไซต์วิจารณ์สินค้าในประเทศไทย จะมีรูปแบบการเขียนในภาษาผสม คือกล่าวถึงชื่อและแบรนด์สินค้าในภาษาอังกฤษ ส่วนเนื้อหาจะเป็นภาษาไทย ในปัจจุบันระบบค้นหาชื่อสินค้าที่เป็น API ยังไม่สามารถดึงชื่อสินค้าที่เป็นภาษาอังกฤษจากเว็บไซต์วิจารณ์สินค้าที่มีเนื้อหาหลักเป็นภาษาไทยได้ โครงงานนี้จึงมุ่งไปที่การนำเสนอวิธีการในการสกัดชื่อและประเภทสินค้าจากเว็บไซต์วิจารณ์สินค้า ทีมผู้วิจัยได้พัฒนา API เพื่อให้ผู้ที่สนใจสามารถใช้งานเพื่อสกัดชื่อและประเภทสินค้าได้ผ่านทางเว็บบราวเซอร์ จากการทดสอบพบว่าวิธีการที่ใช้รูปแบบของประโยคสามารถสกัดชื่อและประเภทสินค้าได้และมีความแม่นยำที่ดี ผลการทดลองการสกัดหาชื่อสินค้าจำนวน 960 เว็บเพจได้ความแม่นยำที่ 97.48% และการทดสอบหาประเภทสินค้าจำนวน 861 เว็บเพจได้ค่าความแม่นยำที่ 92.73%

คำสำคัญ – สกัดชื่อสินค้า; รูปแบบการจัดคู่คำ;

1. บทนำ

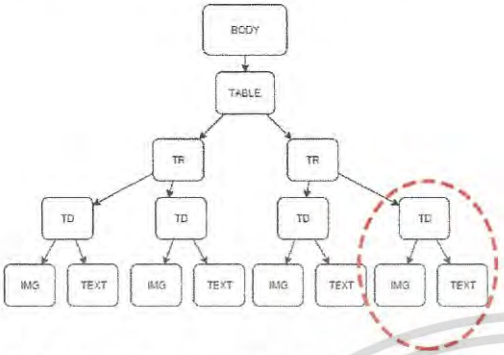
เนื่องจากอัตราการใช้อินเทอร์เน็ตในประเทศกำลังพัฒนาอย่างประเทศไทยสูงขึ้น ทำให้อัตราการเข้าถึงสินค้าผ่านอินเทอร์เน็ตเพิ่มมากขึ้นตามไปด้วย พฤติกรรมผู้บริโภคจึงเปลี่ยนไปจากเดิม การซื้อสินค้าออนไลน์กลายเป็นส่วนหนึ่งในชีวิตประจำวันของผู้คน เพราะเทคโนโลยีต่างๆ เข้ามาช่วยอำนวยความสะดวกในชีวิตประจำวัน ส่งผลให้ธุรกิจในปัจจุบันมีอัตราการแข่งขันสูงขึ้น การมีเครื่องมือที่สามารถวิเคราะห์ความต้องการของผู้บริโภคก็ถือเป็นสิ่งจำเป็นต่อที่ธุรกิจ การมีข้อมูลทำให้สร้างความได้เปรียบทางการแข่งขันเหนือคู่แข่ง แต่วิธีการในการทำเหมืองข้อมูลบนเว็บในประเทศไทยยังอยู่ในช่วงคิดค้นพัฒนา เพราะภาษาไทยจะมีความซับซ้อนละเอียดอ่อนกว่าถ้าเทียบกับภาษาอังกฤษ ทำให้การทำเหมืองข้อมูลในภาษาไทยยังไม่ค่อยมีความแม่นยำ และถ้าข้อมูลที่เป็นภาษาไทยอยู่บนเว็บไซต์ยิ่งมีความยุ่งยากในการเขียนระบบเพื่อสังเคราะห์ข้อมูลและจึงเป็นที่มาของงานวิจัยชิ้นนี้ที่จะสร้างระบบบริการส่วนต่อประสานโปรแกรมประยุกต์สำหรับค้นหาชื่อและประเภทสินค้าจากเว็บไซต์สำหรับโครงการวิเคราะห์เว็บไซต์ในอนาคตเพื่อให้ได้วิธีการวิเคราะห์ที่มีประสิทธิภาพดีกว่าเดิมสำหรับในเว็บภาษาไทย

2. วรรณกรรมที่เกี่ยวข้อง

2.1 An algorithm of product information extraction from web pages

กล่าวถึงวิธีการค้นหาชื่อสินค้าและราคาจากเว็บเพจในเว็บไซต์ร้านค้าออนไลน์ โดยใช้ความรู้ในเรื่อง Document Object (DOM) วิธีนี้สามารถหาข้อมูลสินค้าได้จากโครงสร้าง DOM ของข้อมูลสินค้าในเว็บร้านค้าจะมีลักษณะโครงสร้างเดียวกัน ส่วน DOM ที่มีข้อมูลสินค้าภายในจะประกอบไปด้วยชื่อสินค้าที่เป็น Header, ราคา

สินค้าที่เป็นตัวเลข, และรูปภาพสินค้าอย่างละหนึ่งอัน หากโปรแกรมวิเคราะห์เว็บเพจแล้วพบโครงสร้างแบบนี้ก็จะสามารถตีความได้ว่าภายในคือข้อมูลสินค้า [1]



รูปที่ 2.1 : รูปภาพแสดงโครงสร้างของ DOM ที่ภายในบรรจุข้อมูลสินค้า

2.2 A framework for laptop review analysis

นำเสนอวิธีวิเคราะห์เพื่อสรุปคุณสมบัติในด้านต่างๆ ออกมาเป็นข้อดีและข้อเสียของ Laptop จากเว็บไซต์แลกเปลี่ยนความคิดเห็น โดยแบ่งออกเป็นด้านประสิทธิภาพ, ด้านการออกแบบ และด้านความสามารถ โดยหลักการของ Part of Speech มาวิเคราะห์ประโยคเพื่อหาประธานและกรรม จากนั้นจึงนำ Emoticon Lexicon สำหรับบ่งบอกว่าประโยคนั้นๆเป็นประโยคด้านบวกหรือด้านลบ จากนั้นใช้ Machine Learning จำแนกประโยคว่ากล่าวถึงคุณสมบัติด้านไหนของ Laptop [2]

ตารางที่ 2.1 คำศัพท์ที่บ่งบอกคุณสมบัติของ Laptop ถูกแบ่งโดยวิธี Machine Learning

Performance	Design	Feature
Performance	Design	Feature
Ghz	Weight	USB
Mhz	Width	HDMI
CPU	Height	VGA
Memory	Size	Touchpad

3. การวิเคราะห์และระบบปัจจุบัน

3.1 การวิเคราะห์ระบบที่มีในปัจจุบัน

ปัจจุบันมีเว็บไซต์หลายแห่งสำหรับบริการส่วนต่อประสานโปรแกรมประยุกต์สำหรับสกัดและวิเคราะห์ข้อมูลจากเว็บไซต์มีทั้งแบบเชิงพาณิชย์และแบบไม่แสวงหาผลกำไร เพื่อใช้ในทางการศึกษาวิเคราะห์การตลาดในธุรกิจ ซึ่งแต่ละระบบนั้นมีความสามารถแตกต่างกันออกไป ผู้จัดทำโครงการจึงทำการทดสอบเว็บไซต์ที่ให้บริการสกัดเนื้อหามาทดสอบจำนวน 5 เว็บไซต์ดังนี้

ตารางที่ 3.1 ผลการทดสอบระบบที่มีในปัจจุบัน

Features	APIs			
	Diffbot	*Alchemy API*	*Embed.ly*	*Readability*
1. Article extraction	✓	✓	✓	✓
2. Product extraction	✓			
3. Clean plaintext	✓	✓		
4. Language detection	✓	✓	✓	
5. Thai Supporting	✓			

จากตารางที่ 3.1 การทดสอบคุณสมบัติในข้อ 1 – 4 ผู้จัดทำโครงการได้ดึงผลลัพธ์จากการทดสอบมาจากเว็บ diffbot.com ที่ทำการทดสอบไว้อยู่แล้ว และจากคุณสมบัติในข้อ 5 จะเห็นได้ว่ามีเพียง Diffbot เท่านั้นที่รองรับภาษาไทย แต่ยังไม่ค่อยมีความแม่นยำสำหรับเว็บประเภทวิจารณ์สินค้า ในการหาชื่อสินค้าและไม่สามารถบอกประเภทสินค้าได้

รูปที่ 3.1 ผลการทดสอบ Diffbot API ด้วยเว็บไซต์วิจารณ์สินค้า

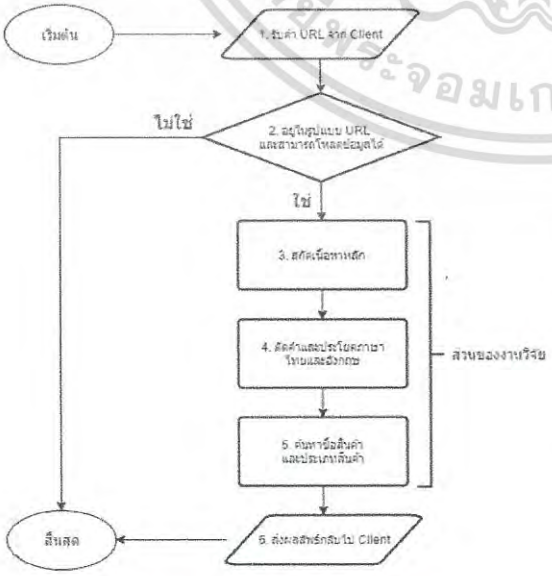
ในรูปที่ 3.1 แสดงถึงผลการทดสอบ Diffbot API ด้วยเว็บไซต์ประเภทวิจารณ์สินค้า ในส่วนของ Product API จะเห็นได้ว่าผลลัพธ์ที่ออกมาไม่สามารถบอกถึงชื่อสินค้าได้ บอกได้เพียงแค่แบรนด์ของสินค้า

นั้นๆ จึงพอสรุปได้ว่า Diffbot API ยังไม่มีความสามารถในการหาชื่อสินค้า

3.2 กระบวนการดำเนินงานของระบบ

ดังแสดงในรูปที่ 3.2 มีขั้นตอนดังนี้

1. Application Programming Interface Web Service รับคำสั่ง Request โดยประกอบด้วย URL, Output Type ในรูปแบบภาษา JSON จากเครื่อง Client
2. ระบบแปลงภาษา JSON มาเก็บในตัวแปรของภาษาระบบ แล้วตรวจสอบว่า URL ที่ถูกส่งเข้ามามีรูปแบบใน Format ของ URL หรือไม่ หากไม่ใช่ก็จบการทำงาน
3. ระบบทำการ Extract Main Article Content จากโครงสร้าง HTML ด้วยวิธีการ Boilerplate โดยใช้หลักการของ Text Density
4. ตัดคำและประโยคภาษาอังกฤษและภาษาไทยด้วย PyICU
5. นำ Main Article Content มาวิเคราะห์เพื่อหาชื่อและประเภทของสินค้าภายในเว็บไซต์
6. นำผลลัพธ์ที่ได้ เข้ารหัสให้อยู่ในรูปแบบ JSON แล้วส่งกลับไปให้เครื่อง Client



รูปที่ 3.2 กระบวนการดำเนินงานของระบบ

4. ระบบส่วนต่อประสานโปรแกรมประยุกต์เพื่อการสังเคราะห์ชื่อและประเภทสินค้าจากเว็บไซต์

4.1 การสกัดเนื้อหาหลักจากเว็บ

เป็นขั้นตอนการดึงข้อมูล HTML ด้วยโปรโตรคอล HTTP/1.1 มาเก็บไว้ในระบบ แล้วระบบจะทำการสกัดเอาใจความสำคัญหลักของหน้าเว็บออกมา ในรูปที่ 4.2 จะเป็นแสดงให้เห็นว่าขั้นตอนการสกัดเนื้อหาหลักจากเว็บจะทำการดึงเฉพาะส่วนที่เป็นใจความสำคัญหลักของเพจ (Main Article Content) คือบล็อคในกรอบสีแดง และตัดส่วนอื่นที่ไม่ใช่เนื้อหาหลักออก เช่น Menu Bar, Nav Bar, Login, Relate Content, Footer, Ads เป็นต้น

รูปที่ 4.1 : แสดงบล็อคต่าง ๆ ในหน้าเว็บทั่วไป

โดยใช้วิธีการ Boilerplate จาก library ใน PyPi ที่มีชื่อว่าBoilerpipe ในการสกัด Main Article Content จากโครงสร้างเว็บเพจ จากการทดสอบการสกัดเนื้อหาหลักจากเว็บไซต์ที่เป็นภาษาไทยจำนวน 5 เว็บไซต์ โดยทำการเลือกเว็บเพจมาจำนวนทั้งหมด 20 เว็บเพจ จึงพบว่าผลลัพธ์การคำนวนค่า Precistion และ Recall เท่ากับ 0.89 และ 0.76 ตามลำดับ

ตารางที่ 4.1 ผลการทดสอบการ Extract Main Article Content ด้วย Boilerpipe

รายการทดสอบ	Precision	Recall
Extract Main Article Content	89%	76%

4.2 การหาชื่อและแบรนด์สินค้า

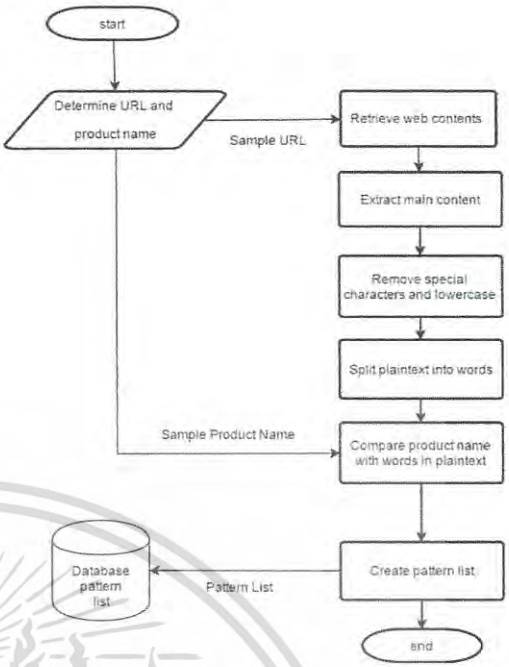
การหาชื่อและแบรนด์สินค้าเป็นวิธีการที่ทำให้รู้ว่าสินค้าที่กำลังถูกโฆษณาในแต่ละเว็บเพจคือสินค้าอะไร โดยจะแบ่งออกเป็น 2 กระบวนการ

4.2.1 การเรียนรู้รูปแบบประโยคแบบขั้นบันได (Staircase Pattern Learning)

การหารูปแบบประโยคอัตโนมัติเป็นวิธีการที่สอนวิธีคิดให้กับโปรแกรม โดยการหารูปแบบประโยคที่สามารถเชื่อมโยงไปถึงชื่อและแบรนด์สินค้าที่อยู่ในใจความสำคัญของเพจ จากนั้นก็เก็บรูปแบบประโยคเหล่านั้นไว้บนฐานข้อมูล

วิธีการนี้ประกอบด้วย 6 ขั้นตอน คือ

1. ดึงข้อมูลของแต่ละเว็บจาก URL โดยใช้โปรโตคอล http แล้วนำข้อมูลที่ได้มาเก็บไว้บนระบบ
2. สกัดใจความหลักจากเว็บเพจโดยใช้วิธีการ Boilerpipe ในการลบส่วนที่ไม่เกี่ยวข้องกับเนื้อหาหลัก เพื่อสกัดเอาแค่ใจความสำคัญของข้อมูลในหน้า html ทำให้ได้ผลลัพธ์ออกมาเป็นข้อมูลที่เป็นตัวอักษร
3. แทนที่อักษรพิเศษ ,เว้นวรรคใหญ่ ,เว้นบรรทัด และรหัสอักษรด้วยเว้นวรรคเล็ก และแปลงตัวอักษรภาษาอังกฤษทั้งหมดให้เป็นตัวพิมพ์เล็ก
4. ใช้ไลบรารี่ PyICU ในการแยกประโยคภาษาไทยให้ออกมาเป็นคำที่มีความหมายโดยมีรากฐานคำศัพท์มาจาก Regular Expression ส่วนประโยคภาษาอังกฤษจะแยกเป็นคำไว้อยู่แล้ว
5. เปรียบเทียบชื่อสินค้าในตัวอย่างกับชื่อสินค้าที่ได้จากระบบ
6. จดจำคำที่อยู่ด้านหน้าและหลังของชื่อสินค้าจำนวนอย่างละห้าคำ
7. นำคำที่ได้มาสร้างเป็นรูปแบบของประโยคแล้วเก็บไว้ในฐานข้อมูล



รูปที่ 4.2 Flow Chart การเรียนรู้รูปแบบประโยคแบบขั้นบันได (Staircase Pattern Learning)

รูปที่ 4.3 แสดงให้เห็นถึง อัลกอริทึมของการเรียนรู้รูปแบบประโยคแบบขั้นบันได (Staircase Pattern Learning)

$\sum w_i$ คือคำทุกคำที่สกัดได้จากเนื้อหลักบนเว็บ, w คือคำที่สกัดได้จากเนื้อหลักบนเว็บ, i คือตำแหน่งของคำ, w_i คือคำในตำแหน่งที่ i, sp คือตัวอย่างชื่อสินค้า, k คือค่าคงที่สำหรับกำหนดความยาวสูงสุดของคำที่ Pattern list หนึ่งอันสามารถมีได้

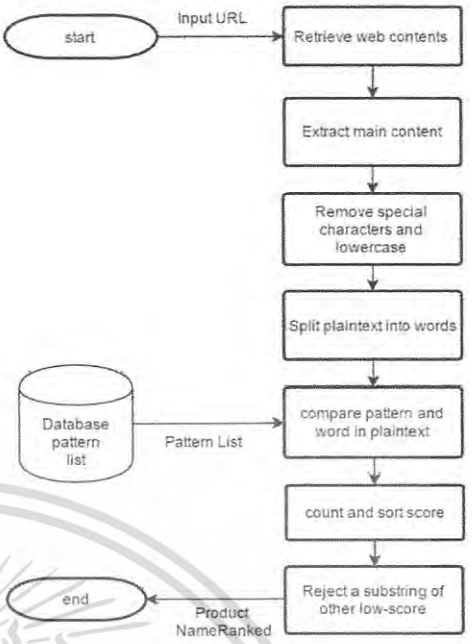
```
for i = 0 ; i < length of p_i ; i++{
    n = 0;
    while(sp[n] == w_i + n){
    n = n + 1;
    if (n == length of sp)
    for(j = 0 ; j < k ; j++){
        add (w_{i+n+j}) to temp
        add (w_{i-j-1}) to temp
    }
    add temp to patterm list
    break;
}
```

รูปที่ 4.3 อัลกอริทึมการเรียนรู้รูปแบบประโยคแบบขั้นบันได (Staircase Pattern Learning)

4.2.2 การจับคู่คำในรูปแบบขั้นบันได (Staircase Pattern Matching)

การจับคู่ของคำในรูปแบบขั้นบันได เป็นวิธีวิเคราะห์แล้วให้โอกาสที่แต่ละคำในเว็บเพจมีโอกาสที่จะเป็นชื่อสินค้าและแบรนด์ ซึ่งประกอบด้วย 7 ขั้นตอน

1. ดึงข้อมูลของแต่ละเว็บจาก URL โดยใช้โปรโตคอล http แล้วนำข้อมูลที่ได้มาเก็บไว้บนระบบ
2. สกัดใจความหลักจากเว็บเพจโดยใช้วิธีการ Boilerpipe ในการลบส่วนที่ไม่เกี่ยวข้องกับเนื้อหาหลัก เพื่อสกัดเอาแค่ใจความสำคัญของข้อมูลในหน้า html ทำให้ได้ผลลัพธ์ออกมาเป็นข้อมูลที่เป็นตัวอักษร
3. แทนที่อักษรพิเศษ ,เว้นวรรคใหญ่ ,เว้นบรรทัด และรหัสอักษรด้วยเว้นวรรคเล็ก และแปลงตัวอักษรภาษาอังกฤษทั้งหมดให้เป็นตัวพิมพ์เล็ก
4. ใช้ไลบรารี่ PyICU ในการแยกประโยคภาษาไทยให้ออกมาเป็นคำที่มีความหมายโดยมีรากฐานคำศัพท์มาจาก Regular Expression ส่วนประโยคภาษาอังกฤษจะแยกเป็นคำไว้อยู่แล้ว
5. ดึงข้อมูลรูปแบบของประโยคทั้งหมดในฐานข้อมูลมาเก็บไว้ในหน่วยความจำของระบบแล้วเทียบคำทีละคำกับข้อมูลใจความหลักของเว็บเพจหากตรงกันก็จะให้คะแนนคำๆนั้น (ขึ้นอยู่กับว่าเป็นรูปแบบประโยคแบบก่อนหรือหลังคำ) ซึ่งคะแนนดังกล่าวทำให้รู้ว่าโอกาสที่คำๆนั้นจะเป็นชื่อและแบรนด์สินค้ามากเพียงใด
6. เรียงลำดับคะแนนจากมากไปน้อย
7. ตัดคำที่มีจำนวนพยางค์น้อยกว่า 2 พยางค์ และตัดคำที่เป็น Substring ของคำที่มีคะแนนน้อยกว่า



รูปที่ 4.4 Flow Chart การจับคู่คำในรูปแบบขั้นบันได (Staircase Pattern Matching)

รูปที่ 4.5 แสดงให้เห็นถึง อัลกอริทึมของการจับคู่คำในรูปแบบขั้นบันได (Staircase Pattern Matching)

$\sum w_i$ คือคำทุกคำที่สกัดได้จากเนื้อหลักบนเว็บ, w คือคำที่สกัดได้จากเนื้อหลักบนเว็บ, i คือตำแหน่งของคำ, w_i คือคำในตำแหน่งที่ i, k คือค่าคงที่สำหรับกำหนดความยาวสูงสุดของคำที่ Pattern list หนึ่งอันสามารถมีได้

```
for ( j ; j < length of p_j ; j++)
    (for i = 0 ; i < length of w_i ; i++){
    n = 0;
    do{
    if (w_{i+n} != p_j[n])break;
    else{
            n = n + 1;
            if (n == length of p_j)
                Add (w_{i+n}) to product list
    }while( n < length of p_j );
}
```

รูปที่ 4.5 อัลกอริทึมการจับคู่คำในรูปแบบขั้นบันได (Staircase Pattern Matching)

4.3 การหาประเภทสินค้า

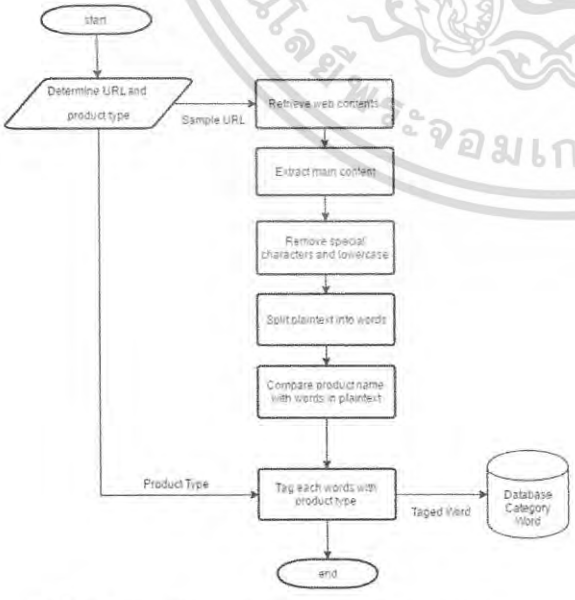
ทีมผู้วิจัยแบ่งประเภทเว็บเพจวิจารสินค้าส่วนใหญ่ออกเป็น 4 ประเภท คือ เทคโนโลยี , เครื่องสำอาง , การท่องเที่ยว และร้านอาหาร ประเภทของเว็บเพจจะบ่ง

บอกถึงประเภทของสินค้าที่นำเสนอ วิธีที่เราใช้แบ่งออกเป็น 2 ขั้นตอนคือ 1. Tagging Type to Word 2. Finding Type from Tagged Word

4.3.1 Tagging Type to Word

เป็นขั้นตอนสอนโปรแกรมเรียนรู้ว่าคำต่างๆแต่ละคำที่ถูกเขียนในเว็บเพจนั้นเป็นเว็บประเภทอะไร แล้วทำการเก็บเป็นสถิติลงในฐานข้อมูล โดยมีขั้นตอนทั้งหมด 5 ขั้นตอน

1. ดึงข้อมูลของแต่ละเว็บจาก URL โดยใช้โปรโตคอล http แล้วนำข้อมูลที่ได้มาเก็บไว้บนระบบ
2. สกัดใจความหลักจากเว็บเพจโดยใช้วิธีการ Boilerpipe ในการลบส่วนที่ไม่เกี่ยวข้องกับเนื้อหาหลัก เพื่อสกัดเอาแค่ใจความสำคัญของข้อมูลในหน้า html ทำให้ได้ผลลัพธ์ออกมาเป็นข้อมูลที่เป็นตัวอักษร
3. แทนที่อักษรพิเศษ ,เว้นวรรคใหญ่ ,เว้นบรรทัด และรหัสอักษรด้วยเว้นวรรคเล็ก และแปลงตัวอักษรภาษาอังกฤษทั้งหมดให้เป็นตัวพิมพ์เล็ก
4. ใช้ไลบรารี่ PyICU ในการแยกประโยคภาษาไทยให้ออกมาเป็นคำที่มีความหมายโดยมีรากฐานคำศัพท์มาจาก Regular Expression ส่วนประโยคภาษาอังกฤษจะแยกเป็นคำไว้อยู่แล้ว
5. เก็บคำที่แท็กประเภทเว็บไซต์ลงฐานข้อมูล

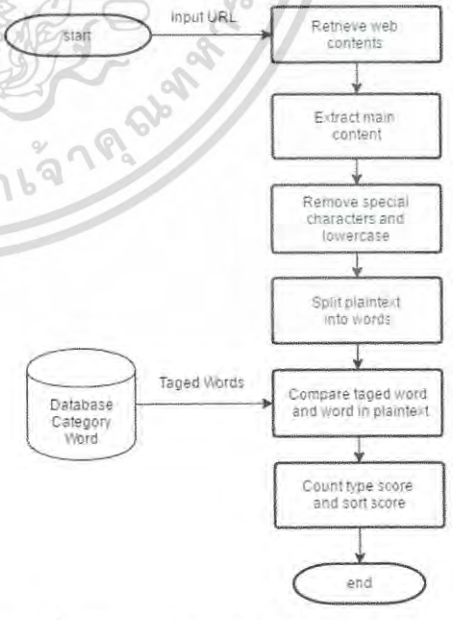


รูปที่ 4.6 Flow Chart การทำงานของ Tagging Type to Word

4.3.2 Finding Website Type from Tagged Word

เป็นขั้นตอนการหาประเภทของเว็บไซต์โดยใช้ข้อมูลของคำที่มีแท็กที่ได้บันทึกไว้จากฐานข้อมูล ประกอบด้วย 6 ขั้นตอน ดังนี้

1. ดึงข้อมูลของแต่ละเว็บจาก URL โดยใช้โปรโตคอล http แล้วนำข้อมูลที่ได้มาเก็บไว้บนระบบ
2. สกัดใจความหลักจากเว็บเพจโดยใช้วิธีการ Boilerpipe ในการลบส่วนที่ไม่เกี่ยวข้องกับเนื้อหาหลัก เพื่อสกัดเอาแค่ใจความสำคัญของข้อมูลในหน้า html ทำให้ได้ผลลัพธ์ออกมาเป็นข้อมูลที่เป็นตัวอักษร
3. แทนที่อักษรพิเศษ ,เว้นวรรคใหญ่ ,เว้นบรรทัด และรหัสอักษรด้วยเว้นวรรคเล็ก และแปลงตัวอักษรภาษาอังกฤษทั้งหมดให้เป็นตัวพิมพ์เล็ก
4. ใช้ไลบรารี่ PyICU ในการแยกประโยคภาษาไทยให้ออกมาเป็นคำที่มีความหมายโดยมีรากฐานคำศัพท์มาจาก Regular Expression ส่วนประโยคภาษาอังกฤษจะแยกเป็นคำไว้อยู่แล้ว
5. นับจำนวนคำที่มี Tag ประเภทของเว็บไซต์ภายในเว็บเพจโดยดึงคำที่ติด Tag มาจากฐานข้อมูล
6. เรียงลำดับคะแนนว่าเว็บเพจนี้ประกอบด้วยคำในประเภทไหนมากที่สุดได้ประเภทของเว็บ



รูปที่ 4.7 Flow Chart การทำงานของ Finding Website Type from Tagged Word

5. ผลการทดลองและสรุปผล

5.1 ผลการทดลอง

ข้อมูลที่ใช้ในการทดลองคือบทความรีวิวสินค้าและบริการบนเว็บไซต์มีชื่อเสียงในประเทศไทยที่ถูกเขียนขึ้นตั้งแต่วันที่ 1 ม.ค. 2558 ถึง 30 มี.ค. 2559 โดยจะแยกข้อมูลออกเป็น 2 ชุดด้วยกัน ข้อมูลชุดแรกสำหรับหาชื่อสินค้า จะมีเว็บสินค้าและบริการปะปนกันจำนวน 960 เว็บเพจ และข้อมูลชุดที่สองสำหรับหาประเภทสินค้าจะมีการแยกหมวดหมู่เว็บเพจไว้ทั้งหมด 4 หมวดหมู่ จำนวน 861 เว็บเพจ แยกออกเป็นสินค้า 2 หมวดหมู่และบริการ 2 หมวดหมู่ หมวดหมู่สินค้า ประกอบด้วย 1. สินค้าในหมวดหมู่เทคโนโลยี คือ โน๊ตบุ๊ค ,โทรศัพท์มือถือ ,กล้องถ่ายรูป ,แก็ดเจ็ต 2. สินค้าในหมวดหมู่เครื่องสำอางค์ หมวดหมู่บริการ ประกอบด้วย 1. ร้านอาหารและคาเฟ่ 2. ที่พัก คือ โรงแรมขนาดเล็ก (Hostel) ,โรงแรมขนาดใหญ่ (Hotel) ,รีสอร์ท ซึ่งสินค้าและบริการที่เลือกมาในหมวดหมู่ คือสินค้าและบริการที่ผู้บริโภคมักจะดูจากเว็บไซต์วิจารณ์หรือรีวิวก่อนทำการตัดสินใจซื้อหรือเข้ารับบริการ

5.1.1 การหาความถูกต้องของการสกัดชื่อและแบรนด์สินค้า

ใช้ข้อมูลชุดแรกในการทดสอบ โดยจะแยกข้อมูลเพื่อนำมาใช้สอนวิธีคิดการค้นหาชื่อสินค้าให้กับโปรแกรม 63% คิดเป็นจำนวน 602 เว็บเพจ สำหรับการทดสอบความถูกต้อง 37% คิดเป็นจำนวน 358 เว็บเพจ โปรแกรมจะเรียนรู้จากค่าความถูกต้องของรูปแบบประโยคจากเนื้อหาหลักของเว็บไซต์วิจารณ์สินค้า ค่าความถูกต้องคือค่าที่ได้จากการเปรียบเทียบชื่อสินค้าที่แท้จริงกับชื่อสินค้าที่ได้จากการคำณวน และอัตราส่วนความถูกต้องของชื่อสินค้าจะถูกนำมาใช้ในการพัฒนาและใช้ในการทดสอบวัดผล

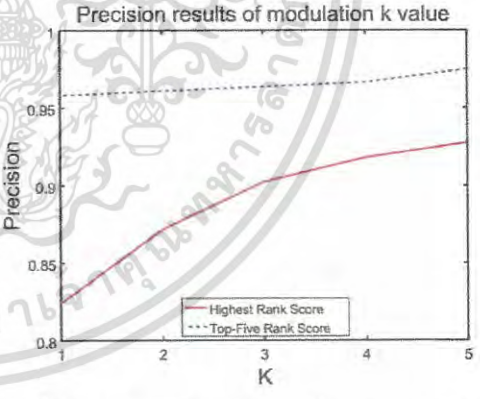
ผลการทดลองแสดงในตาราง 5.1 คือผลการทดลองการหาความถูกต้องของการสกัดชื่อและแบรนด์สินค้า ด้วยวิธีการเรียนรู้รูปแบบประโยคแบบขั้นบันได (Staircase Pattern Learning) และวิธีการจับคู่คำในรูปแบบขั้นบันได (Staircase pattern matching)

ตารางที่ 5.1 ผลการทดสอบการสกัดชื่อและแบรนด์สินค้า

Precision Type	Highest Rank Score	Top-Five Rank Score
Number of webpages	358	358
Number of correctly extracted webpages	332	349
Number of incorrectly extracted webpages	26	9
Precision (%)	92.73%	97.48%

ในรูปที่ 5.1 เป็นการแสดงค่าความถูกต้องที่เปลี่ยนแปลงไปหลังจากเปลี่ยนค่า K

K คือค่าคงที่ สำหรับกำหนดความยาวของคำที่ Pattern list หนึ่งอันสามารถมีได้ ในกระบวนการของการเรียนรู้รูปแบบประโยคแบบขั้นบันได ถ้า Pattern list มีจำนวนคำมาก จะยิ่งทำให้เกิดความแตกต่างระหว่าง Pattern อื่นมากขึ้น ซึ่งความแตกต่างเป็นสิ่งสำคัญที่ทำให้รู้ว่าคำที่โปรแกรมหามาได้นั้น ใช่ชื่อสินค้าหรือไม่ ถ้า K มีค่ามากจะทำให้ผลลัพธ์ที่ได้ออกมาดีขึ้น แต่ก็ใช้เวลาในการประมวลผลมากขึ้นตาม จากการทดลองทีมผู้วิจัยจึงพบว่า ค่า K ที่เหมาะสมที่สุดคือ 5 นั่นเอง



รูปที่ 5.1 ค่าความถูกต้องจากการเปลี่ยนแปลงของค่า K

5.1.2 การหาความถูกต้องของการสกัดประเภทสินค้า

ทีมผู้วิจัยใช้ข้อมูลชุดสองสำหรับการทดสอบ โดยจะแยกข้อมูลเพื่อนำมาใช้สอนวิธีจำแนกประเภทให้กับโปรแกรม 52% คิดเป็นจำนวน 450 เว็บเพจ และสำหรับการทดสอบความถูกต้อง 48% คิดเป็นจำนวน 411 เว็บเพจ โปรแกรมจะเรียนรู้จากคำที่ถูกใช้ในเว็บแต่ละประเภท เมื่อทดสอบแล้วจะได้ค่าความถูกต้องสำหรับวัดผล โดยค่าความถูกต้องมาจาก จำนวนเว็บไซต์ที่โปรแกรม

วิเคราะห์ออกมาได้ถูกต้องต่อจำนวนเว็บไซต์ที่วิเคราะห์ทั้งหมด ผลการทดลองดังกล่าวแสดงในตาราง 5.2

ตารางที่ 5.2 : ผลการทดสอบการสกัดชื่อและแบรนด์สินค้า

Precision Type	Type Equal
Number of webpages	411
Number of correctly extracted webpages	358
Number of incorrectly extracted webpages	53
Precision (%)	87.10%

5.2 สรุปผลการวิจัยและดำเนินงาน

จากการทดลองวิธีการที่ใช้หาชื่อและแบรนด์สินค้าจากเนื้อหาหลักบนเว็บไซต์วิจารณ์สินค้าที่เขียนในรูปแบบภาษาภาษาไทยและภาษาอังกฤษด้วยวิธีการเรียนรู้รูปแบบประโยคแบบขั้นบันได (Staircase Pattern Learning) และวิธีการจับคู่คำในรูปแบบขั้นบันได (Staircase pattern matching) สามารถหาชื่อและแบรนด์สินค้าได้อย่างแม่นยำ แต่มีข้อจำกัดในการคำนวน คือ 1.) เว็บไซต์วิจารณ์สินค้าจะต้องมีชื่อและแบรนด์สินค้าอยู่ในส่วนของเนื้อหาหลักของเว็บ 2.) ชื่อและแบรนด์สินค้าจะต้องเป็นภาษาอังกฤษและเนื้อหาหลักจะต้องเป็นภาษาไทยเท่านั้น

ส่วนการทดลองแบ่งประเภทสินค้า จะมี 2 ขั้นตอนคือ การเรียนรู้ประเภทของสินค้ากับคำศัพท์จากเว็บเพจที่กล่าวถึงสินค้า (Tagging Type to Word) และวิธีการหาประเภทสินค้าจากคำศัพท์ในฐานข้อมูล (Finding Website Type from Tagged Word) ก็ให้ผลลัพธ์ในการแบ่งประเภทที่ค่อนข้างแม่นยำ โดยวิธีการนี้จะมีประสิทธิภาพเพิ่มมากขึ้นตามจำนวนเว็บเพจที่นำมาสอนโปรแกรม

เอกสารอ้างอิง

[1] W. Choochaiwattana, "An algorithm of product information extraction from web pages: a document object model Analysis approach," in Proc. ICICM 2012, 2012, pp.103-107.

[2] B. Mehta and M. Narvekar, "DOM tree based approach for web content extraction," in Proc. ICCICT, 2015, pp. 1-6.