บทที่ 3

วิธีดำเนินการวิจัย

การทำวิจัยนี้ผู้วิจัยได้ศึกษาแนวคิด ทฤษฎี หลักการ ตลอดจนเอกสารและงานวิจัยต่างๆ ดังรายละเอียดที่กล่าวไว้ในบทที่ 2 ซึ่งในบทนี้ได้นำเอาความรู้ที่ได้มาประยุกต์ในการพัฒนาระบบแนะนำข้อมูลท่องเที่ยวในประเทศไทย โดยการปรับปรุงอัลกอริทึมเนมแมทชิ่ง นาอีฟเบย์และออนโทโลจีเชิงเวลา ประกอบด้วยเนื้อหา ดังนี้

1. อุปกรณ์ที่ใช้ในการดำเนินงานวิจัย
2. วิธีดำเนินงานวิจัย

อุปกรณ์ที่ใช้ในการดำเนินงานวิจัย

อุปกรณ์ที่ใช้ในการดำเนินงานวิจัย ประกอบด้วยเครื่องคอมพิวเตอร์จำนวน 1 เครื่อง ซึ่งมีส่วนประกอบที่สำคัญ คือ

1. อุปกรณ์ด้านฮาร์ดแวร์ ได้แก่

1.1 เครื่องคอมพิวเตอร์ (Notebook) – Intel® Core™i7-2670QM Processor 2.20 GHz หน่วยความจำ 8 GB.

1.2 ฮาร์ดดิสก์ขนาด 500 GB.

2. อุปกรณ์ด้านซอฟต์แวร์ ได้แก่

2.1 โปรแกรม Microsoft Windows 7 เป็นระบบปฏิบัติการ

2.2 โปรแกรม Eclipse 3.7.2 ใช้สำหรับการพัฒนาอัลกอริทึมทั้งหมดด้วยภาษา Java ประกอบด้วยอัลกอริทึมต่อไปนี้

2.2.1 อัลกอริทึม Name Variation Matching และอัลกอริทึม ISG

2.2.2 อัลกอริทึม Web Crawler

2.2.3 อัลกอริทึม LSI

2.2.4 อัลกอริทึมนาอีฟเบย์

2.2.5 ปรับปรุงอัลกอริทึมนาอีฟเบย์โดยใช้ค่าขอบเขต

2.2.6 ปฏิทินจันทรคติที่ใช้ในออนโทโลจีเชิงเวลา

และพัฒนาระบบที่ทำงานในลักษณะของ Web Application ด้วยภาษา JSP

2.3 โปรแกรม Java Developer Kit เวอร์ชั่น 7 เป็นตัวแปรภาษา

2.4 โปรแกรม Apache Tomcat 7.0 เป็นเว็บเซิร์ฟเวอร์

2.5 โปรแกรม Google Chrome เป็นเว็บบราว์เซอร์

2.6 โปรแกรม Protégé 3.5 สร้างออนโทโลจีท่องเที่ยวและเก็บข้อมูลท่องเที่ยวในประเทศไทย โดยใช้ภาษา SPARQL ในการสืบค้นข้อมูลท่องเที่ยวที่ถูกเก็บไว้ในออนโทโลจีท่องเที่ยว โดยใช้ Jena API ทำหน้าที่เป็น OWL Parser สำหรับดึงข้อมูลในออนโทโลจี

2.7 โปรแกรม Jess Inference Engine 7.1p2 ใช้เป็นกลไกการอนุมานตามกฎการแนะนำข้อมูลท่องเที่ยวในประเทศไทยที่ถูกเขียนด้วยภาษา SWRL และใช้ Protégé API 3.5 ในการเชื่อมต่อกับ Jess Inference Engine

2.8 โปรแกรม MySQL 5.0 ใช้สำหรับเก็บข้อมูลเว็บไซต์ท่องเที่ยวจากการดึงข้อมูลในหน้าเว็บไซต์ด้วยอัลกอริทึม Web Crawler

วิธีดำเนินการวิจัย

ผู้วิจัยได้ลำดับการดำเนินงานวิจัยตามขั้นตอนต่อไปนี้

1. นำหลักการและงานวิจัยที่เกี่ยวข้องที่ได้ศึกษาในบทที่ 2 มาทำการวิเคราะห์และออกแบบสถาปัตยกรรมของระบบแนะนำข้อมูลท่องเที่ยวในประเทศไทย

2. ด้านข้อมูล

2.1 การออกแบบความสัมพันธ์ของข้อมูลผลลัพธ์จากการสืบค้นข้อมูลท่องเที่ยวในเว็บไซต์กูเกิ้ลด้วยอัลกอริทึม Web Crawler และการจัดหมวดหมู่เว็บไซต์ท่องเที่ยว

2.2 การออกแบบและพัฒนาออนโทโลจีท่องเที่ยว รวมถึงกฎที่ใช้สำหรับการแนะนำข้อมูลท่องเที่ยวในประเทศไทยด้วยภาษา SWRL

2.3 การออกแบบและพัฒนาออนโทโลจีเชิงเวลา

3. ด้านอัลกอริทึม

3.1 การประยุกต์ใช้และพัฒนาอัลกอริทึมเนมแมทชิ่งสำหรับกรองข้อมูล

3.1.1 Name Variation Matching และ อัลกอริทึม ISG

3.2 พัฒนาอัลกอริทึม Web Crawler ในการดึงข้อมูลในหน้าเว็บไซต์

3.3 การวัดประสิทธิภาพอัลกอริทึม ISG และ Name Variation ที่ใช้ในการกรองข้อมูลของระบบแนะนำข้อมูลท่องเที่ยวในประเทศไทย

3.4 การจัดหมวดหมู่เว็บไซต์ท่องเที่ยวด้วยอัลกอริทึม LSI อัลกอริทึมนาอีฟเบย์และอัลกอริทึมนาอีฟเบย์ที่ปรับปรุงโดยใช้ค่าขอบเขต

3.5 การวัดประสิทธิภาพอัลกอริทึมจัดหมวดหมู่เว็บไซต์ท่องเที่ยวในข้อที่ 3.4

3.6 พัฒนาอัลกอริทึมสำหรับคำนวณหาข้อมูลท่องเที่ยวให้ตรงตามความต้องการของผู้ใช้ด้วยอัลกอริทึม K-NN

3.7 พัฒนาอัลกอริทึมสำหรับคำนวณปฏิทินจันทรคติเพื่อแนะนำข้อมูลท่องเที่ยวตามช่วงเวลาและบริเวณใกล้เคียง

4. การออกแบบและพัฒนาระบบแนะนำข้อมูลท่องเที่ยวในประเทศไทย

4.1 ระบบค้นหาข้อมูลท่องเที่ยวโดยใช้คำค้นหาหรือประโยค

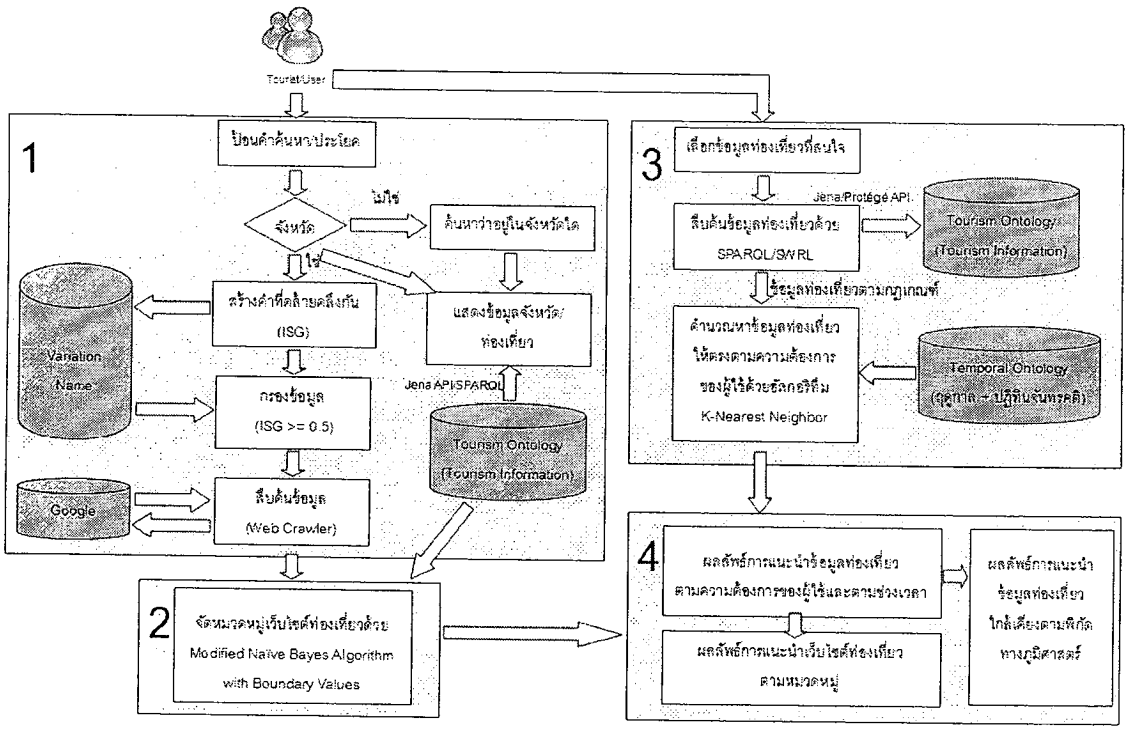
4.2 ระบบแนะนำข้อมูลท่องเที่ยวโดยผู้ใช้เลือกข้อมูลท่องเที่ยวตามความต้องการของผู้ใช้

5. การทดสอบและใช้งานระบบแนะนำข้อมูลท่องเที่ยวในประเทศไทย

6. สรุปผล อภิปรายผล และข้อเสนอแนะ รวมทั้งจัดทำรูปเล่มรายงาน

1. สถาปัตยกรรมของระบบแนะนำข้อมูลท่องเที่ยวในประเทศไทย

ในการทำวิจัยครั้งนี้ผู้วิจัยได้ออกแบบสถาปัตยกรรมของระบบแนะนำข้อมูลท่องเที่ยวในประเทศไทย พร้อมอธิบายขั้นตอนการทำงานของระบบ ดังต่อไปนี้



ภาพ 4 แสดงสถาปัตยกรรมของระบบแนะนำข้อมูลท่องเที่ยวในประเทศไทย

จากภาพ 4 แสดงสถาปัตยกรรมของระบบแนะนำข้อมูลท่องเที่ยวในประเทศไทย โดยแบ่งเป็น 4 ส่วนดังต่อไปนี้

1. การสืบค้นข้อมูลท่องเที่ยวและกรองข้อมูลเว็บไซต์ท่องเที่ยว ซึ่งมีขั้นตอนการทำงานดังต่อไปนี้

1.1 ผู้ใช้ป้อนชื่อจังหวัดหรือสถานที่ท่องเที่ยวหรือประโยคเพื่อสืบค้นข้อมูลท่องเที่ยว

1.2 ระบบจะตรวจสอบข้อมูลที่รับมาเป็นชื่อจังหวัดหรือไม่ ถ้าไม่ใช่ระบบจะค้นหาเพื่อให้รู้ว่าเป็นชื่อจังหวัดอะไร

1.3 แสดงข้อมูลพื้นฐานของจังหวัดนั้นๆ หรือข้อมูลท่องเที่ยวตามที่ผู้ใช้ป้อน ซึ่งใช้ Jena API ในการเชื่อมต่อกับออนโทโลจีท่องเที่ยวที่ถูกเก็บในรูปแบบไฟล์ OWL ด้วยภาษา SPARQL

1.4 นำชื่อจังหวัดมาหาความคล้ายคลึงด้วยอัลกอริทึม ISG ซึ่งจะช่วยให้ชื่อที่มีการเขียนผิดหรือสะกดผิดและปรากฏอยู่ในข้อมูลท่องเที่ยวสามารถถูกสืบค้นพบในอินเทอร์เน็ตได้ เช่น ถ้าค้นหาคำว่า พิษณุโลก อัลกอริทึมจะนำคำต่างๆ ที่คล้ายคลึงกับชื่อจังหวัด พิษณุโลก มาคำนวณความคล้ายคลึงและใช้ค่า ISG ที่มีค่ามากกว่าหรือเท่ากับ 0.5 ในการคัดกรอง

1.5 นำชื่อที่มีความคล้ายคลึงจากตาราง 9 ไปสืบค้นในเว็บไซต์กูเกิ้ลด้วยอัลกอริทึม Web Crawler และทำการจัดเก็บข้อมูลเว็บไซต์ลงในฐานข้อมูลเพื่อจัดเตรียมไว้ใช้ในสมการ 13 ต่อไป

2. การจัดหมวดหมู่เว็บไซต์ ซึ่งส่วนนี้จะนำข้อมูลเว็บไซต์ที่ได้จากการทำงานในข้อที่ 1 มาทำการจัดหมวดหมู่เว็บไซต์ท่องเที่ยวด้วยอัลกอริทึมนาอีฟเบย์ที่ปรับปรุงโดยใช้ค่าขอบเขต (บทที่ 2) แล้วแสดงผลของการจัดหมวดหมู่เว็บไซต์ท่องเที่ยว (ดังหัวข้อที่ 4.1)

3. ระบบแนะนำข้อมูลท่องเที่ยวในประเทศไทย ซึ่งส่วนนี้จะมีขั้นตอนการทำงานดังต่อไปนี้

3.1 ผู้ใช้ป้อนและเลือกประเภทการท่องเที่ยว ประเภทที่พัก ประเภทอาหารที่ชอบ เพศ อายุ วันที่เดินทาง จำนวนวันท่องเที่ยว ซึ่งอาจป้อนข้อมูลท่องเที่ยวคนเดียวหรือมากกว่า 1 คนก็ได้

3.2 นำข้อมูลท่องเที่ยวที่ผู้ใช้เลือกไปเทียบกับกฎสำหรับการท่องเที่ยวซึ่งเก็บอยู่ในรูปแบบภาษา SWRL และนำข้อมูลไปสืบค้นข้อมูลท่องเที่ยวด้วยภาษา SPARQL

3.3 คำนวณหาข้อมูลท่องเที่ยวให้ตรงกับความต้องการของผู้ใช้ด้วยอัลกอริทึม K-NN

3.4 แนะนำข้อมูลท่องเที่ยวตามช่วงเวลาหรือฤดูกาลที่มีในประเทศไทยด้วยการประยุกต์ใช้ออนโทโลจีเชิงเวลา

4. การแสดงผลลัพธ์จากระบบแนะนำข้อมูลท่องเที่ยวในประเทศไทย ซึ่งแบ่งการทำงานเป็น 2 ส่วนคือ

4.1 จากการสืบค้นโดยใช้คำค้นหาหรือใช้ประโยค

4.2 ผลลัพธ์จากการเลือกข้อมูลท่องเที่ยวตามความต้องการของผู้ใช้ ผลลัพธ์ที่ได้จะประกอบไปด้วย

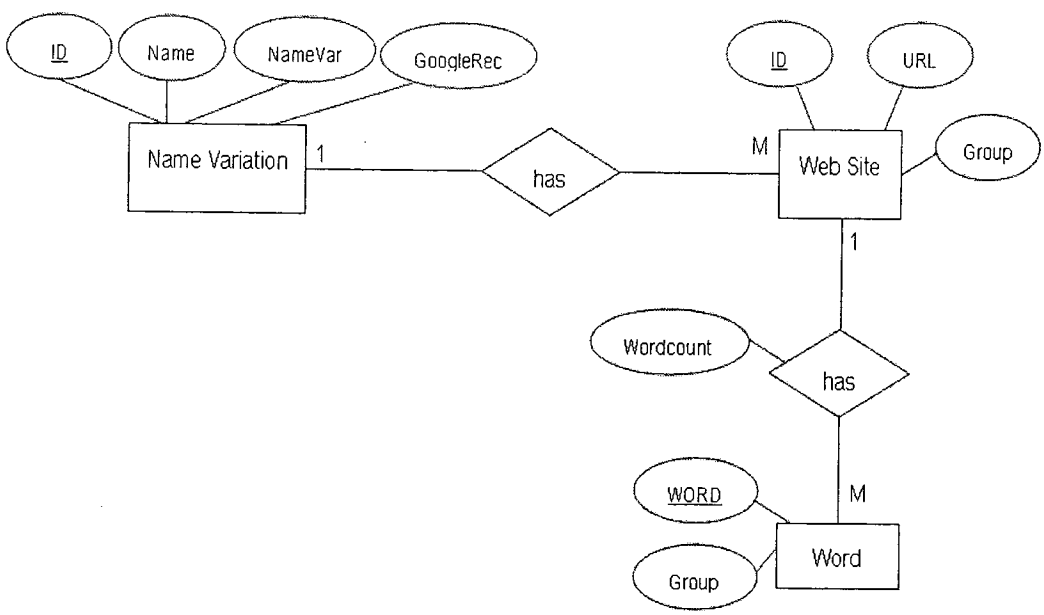
4.2.1 ผลลัพธ์การแนะนำเว็บไซต์ท่องเที่ยวตามการจัดหมวดหมู่ด้วยอัลกอริทึมนาอีฟเบย์ที่ปรับปรุงโดยใช้ค่าขอบเขต

4.2.2 ผลลัพธ์การแนะนำข้อมูลท่องเที่ยวตามความต้องการของผู้ใช้และตามช่วงเวลา

4.2.3 ผลลัพธ์การแนะนำข้อมูลท่องเที่ยวใกล้เคียงตามพิกัดทางภูมิศาสตร์

2. ด้านข้อมูล

2.1 การออกแบบความสัมพันธ์ของข้อมูลผลลัพธ์จากการสืบค้นข้อมูลท่องเที่ยวในเว็บไซต์กูเกิ้ลด้วยอัลกอริทึม Web Crawler และการจัดหมวดหมู่เว็บไซต์ท่องเที่ยว

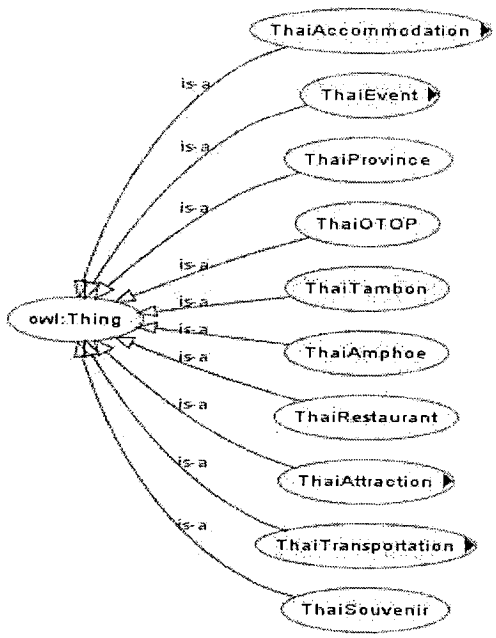


ภาพ 5 แสดงความสัมพันธ์ของข้อมูลผลลัพธ์จากการสืบค้นข้อมูลท่องเที่ยวในเว็บไซต์กูเกิ้ล

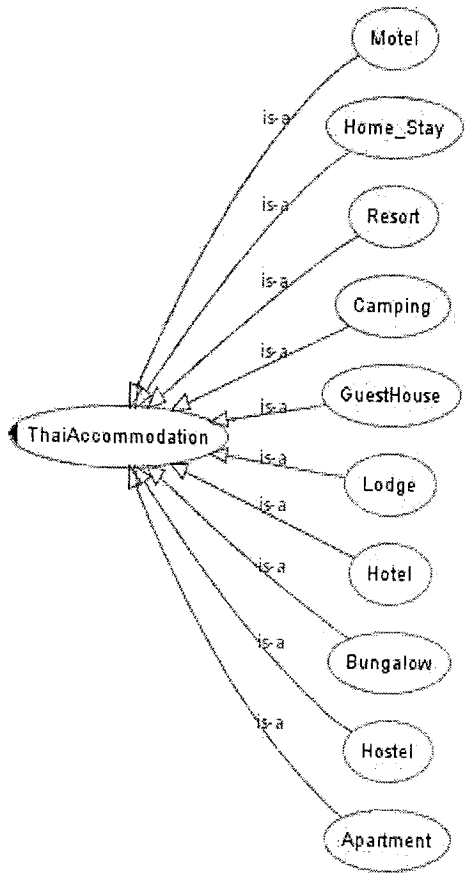
จากภาพ 5 แสดงความสัมพันธ์ของข้อมูลผลลัพธ์จากการสืบค้นข้อมูลท่องเที่ยวในเว็บไซต์กูเกิ้ลด้วยอัลกอริทึม Web Crawler และการจัดหมวดหมู่เว็บไซต์ท่องเที่ยวด้วยอัลกอริทึม LSI อัลกอริทึมนาอีฟเบย์และอัลกอริทึมนาอีฟเบย์ที่ปรับปรุงโดยใช้ค่าขอบเขต โดยอธิบายความสัมพันธ์ได้ดังนี้คือ ชื่อ 1 ชื่อ (Name Variation) มีหลายเว็บไซต์ ซึ่งในแต่ละเว็บไซต์จะเป็นของชื่อนั้นๆ และเว็บไซต์ 1 เว็บไซต์จะมีความถี่ของคำที่ปรากฏในเว็บนั้นหลายคำ ซึ่งแต่ละคำจะเป็นของเว็บไซต์นั้นๆ โดยความสัมพันธ์ในจะเกิดจากการทำงานในส่วนของการเก็บข้อมูลเว็บไซต์เพื่อที่จะใช้ในการวิเคราะห์เว็บไซต์เพื่อจัดหมวดหมู่เว็บไซต์ท่องเที่ยว ดังรายละเอียดที่จะกล่าวในหัวข้อการจัดหมวดหมู่เว็บไซต์ท่องเที่ยว

2.2 การออกแบบและพัฒนาออนโทโลจีท่องเที่ยว

งานวิจัยนี้ผู้วิจัยได้ออกแบบและพัฒนาออนโทโลจีท่องเที่ยวที่ใช้สำหรับเก็บข้อมูลท่องเที่ยวในประเทศไทย โดยใช้โปรแกรม Protégé 3.5 แบ่งเป็น 10 คลาสคือ คลาสจังหวัด (Thai Province) คลาสอำเภอ (Thai Amphoe) คลาสตำบล (Thai Tambon) คลาสเทศกาล (Thai Event) คลาสสถานที่ท่องเที่ยว (Thai Attraction) คลาสที่พัก (Thai Accommodation) คลาสการเดินทาง (Thai Transportation) คลาสร้านอาหาร (Thai Restaurant) คลาสร้านขายของฝาก (Thai Souvenir) และคลาสร้านขายสินค้าหนึ่งผลิตภัณฑ์หนึ่งตำบล (Thai OTOP) ดังภาพ 6 โดยในแต่ละคลาสแยกรายละเอียดได้ดังภาพ 7, 8, 9 และ 10 (นฤพนธ์ พนาวงศ์ และจักรกฤษณ์ เสน่ห์, 2553ก, 2553ข)

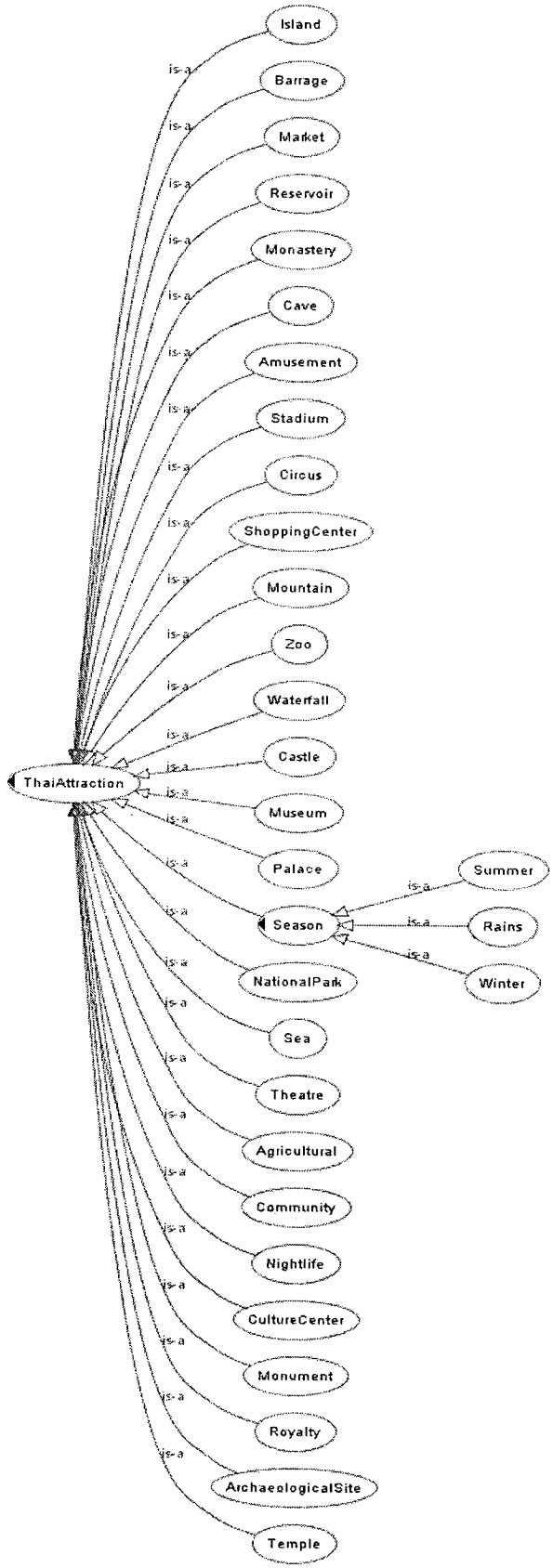


ภาพ 6 แสดงภาพรวมของออนโทโลจีท่องเที่ยว



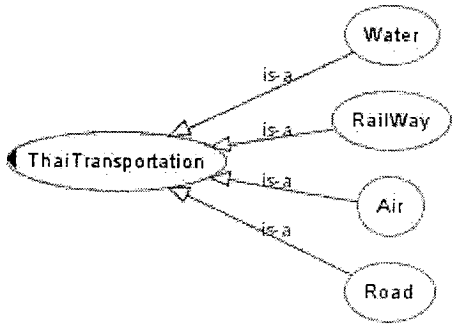
ภาพ 7 แสดงคลาสที่พัก

จากภาพ 7 แสดงรายละเอียดของคลาสที่พัก (ThaiAccommodation) หรือประเภทหรือชนิดที่พักต่างๆ ที่มีอยู่ในประเทศไทยที่นักท่องเที่ยวสามารถสืบค้นได้ ได้แก่ โมเทล (Motel) โรงแรม (Hotel) รีสอร์ต (Resort) ตั้งแคมป์ (Camping) บ้านเช่า/ที่พักชั่วคราว (Guest House) หอพัก (Hostel) อพาร์ทเม้นต์ (Apartment) โฮมสเตย์ (Home_Stay) กระท่อม (Lodge) และบังกะโล (Bungalow)



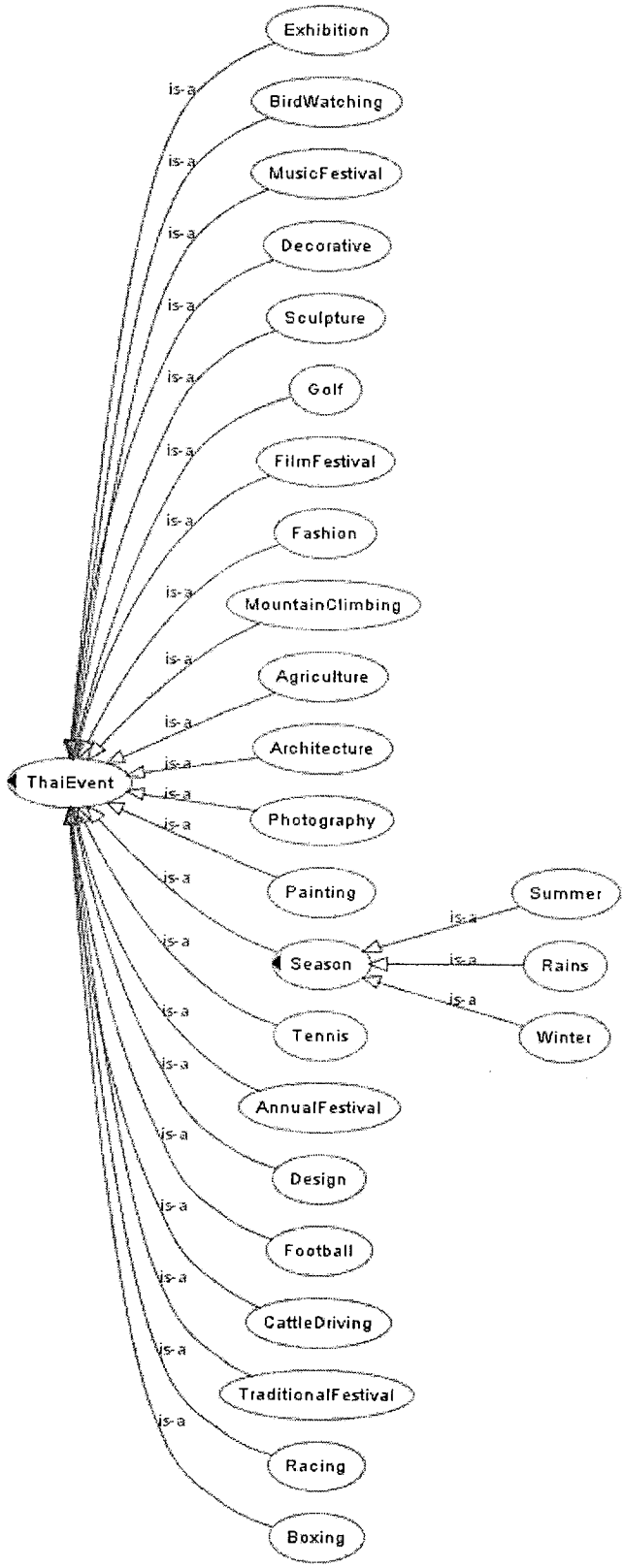
ภาพ 8 แสดงคลาสสถานที่ท่องเที่ยว

จากภาพ 8 แสดงรายละเอียดของคลาสสถานที่ท่องเที่ยว (ThaiAttraction) ที่มีอยู่ในประเทศไทยซึ่งประกอบด้วย ภูเขา (Mountain) น้ำตก (Waterfall) พิพิธภัณฑ์ (Museum) ถ้ำ (Cave) สวนสาธารณะ (National Park) โรงละครสัตว์ (Circus) ห้างสรรพสินค้า (Shopping Center) เขื่อน (Barrage) โรงหนัง (Theatre) อ่างเก็บน้ำ (Reservoir) สถานบันเทิง (Amusement) สวนสัตว์ (Zoo) ปราสาท (Castle) สถานบันเทิงยามราตรี (Nightlife) โบราณสถาน (Archaeological Site) เกาะ (Island) สนามกีฬา (Stadium) ทะเล (Sea) ชุมชน (Community) ทางราชการ (Royalty) วัด (Temple) ตลาด (Market) พระราชวัง (Palace) ศูนย์วัฒนธรรม (Culture Center) อนุสาวรีย์ (Monument) และสำนักสงฆ์ (Monastery) โดยเพิ่มออนโทโลจีเชิงเวลา ชื่อ ฤดูกาล (Season) ที่ประกอบด้วย ฤดูร้อน (Summer) ฤดูฝน (Rains) และฤดูหนาว (Winter)



ภาพ 9 แสดงคลาสการเดินทาง

จากภาพ 9 แสดงรายละเอียดของคลาสการเดินทาง (ThaiTransportation) ที่มีอยู่ในประเทศไทยซึ่งประกอบด้วย ทางน้ำ (Water) ทางรถไฟ (RailWay) ทางอากาศ (Air) และทางถนน (Road)



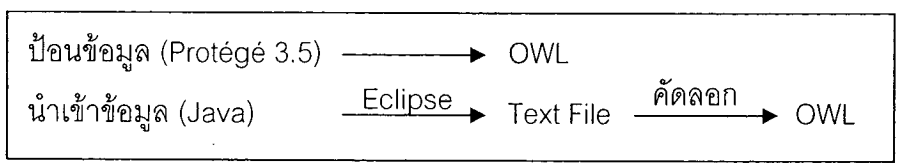
ภาพ 10 แสดงคลาสเทศกาล

จากภาพ 10 แสดงของคลาสเทศกาล (ThaiEvent) หรือกิจกรรมต่าง ๆ ที่สามารถเข้าร่วมได้ในประเทศไทย ซึ่งประกอบด้วยตกแต่ง (Decorative) สถาปัตยกรรม (Architecture) แข่งรถ (Racing) การออกแบบ (Design) แฟชั่น (Fashion) ถ่ายภาพ (Photography) เทศกาลประจำปี (Annual Festival) เทศกาลดนตรี (Music Festival) งานประเพณี (Traditional Festival) นิทรรศการ (Exhibition) เกษตรกรรม (Agriculture) ศิลปะ (Painting) กอฟล์ (Golf) เทศกาลหนัง (Film Festival) ฟุตบอล (Football) ต่อยมวย (Boxing) ประติมากรรม (Sculpture) เทนนิส (Tennis) ดูนก (Birdwatching) วิ่งควาย (Cattledriving) และปีนเขา (Mountain Climbing) โดยเพิ่มออนโทโลจีเชิงเวลา ชื่อ ฤดูกาล (Season) ที่ประกอบด้วย ฤดูร้อน (Summer) ฤดูฝน (Rains) และฤดูหนาว (Winter)

สำหรับการป้อนข้อมูลท่องเที่ยวเพื่อสร้างออนโทโลจีท่องเที่ยวนั้นสามารถทำได้ 2 วิธี ดังต่อไปนี้ (ภาพ 11)

1. การป้อนข้อมูลด้วยโปรแกรม Protégé 3.5 (ภาพ 12) จะได้ไฟล์ OWL ซึ่งวิธีนี้จะใช้เวลานาน จึงได้พัฒนาโปรแกรมในข้อที่ 2 เพื่อลดเวลาในการป้อนข้อมูล

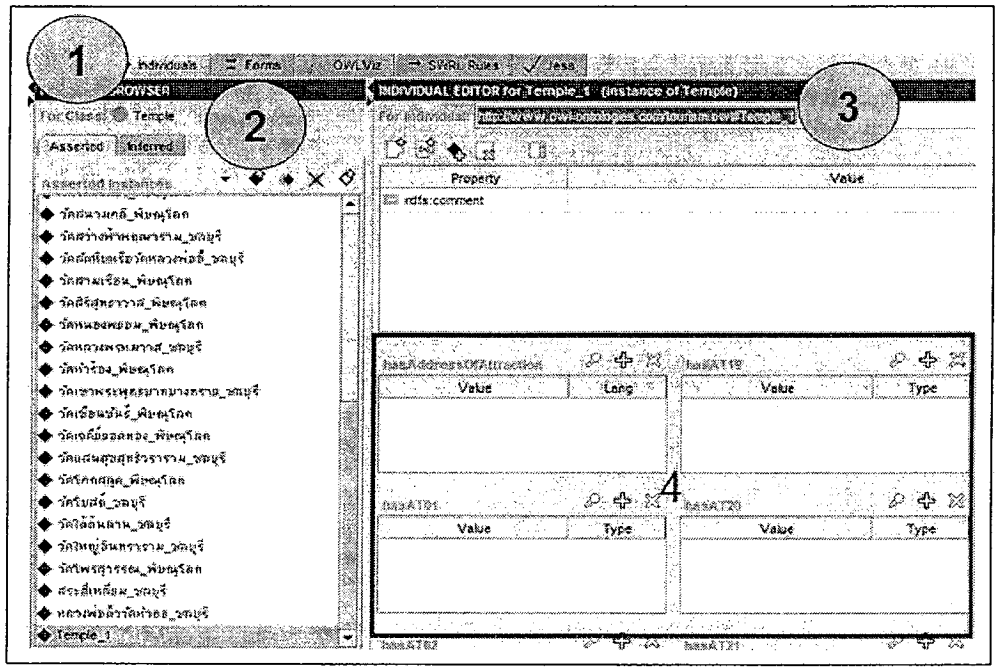
2. นำเข้าข้อมูลท่องเที่ยวโดยโปรแกรม Java รันผ่าน Eclipse ได้เป็น Text File จากนั้นคัดลอกลงไฟล์ OWL



ภาพ 11 แสดงวิธีการป้อนข้อมูลท่องเที่ยว

จากภาพ 12 แสดงตัวอย่างการป้อนข้อมูลท่องเที่ยวผ่านทางโปรแกรม Protégé 3.5 ซึ่งมีขั้นตอนการป้อนข้อมูลดังต่อไปนี้

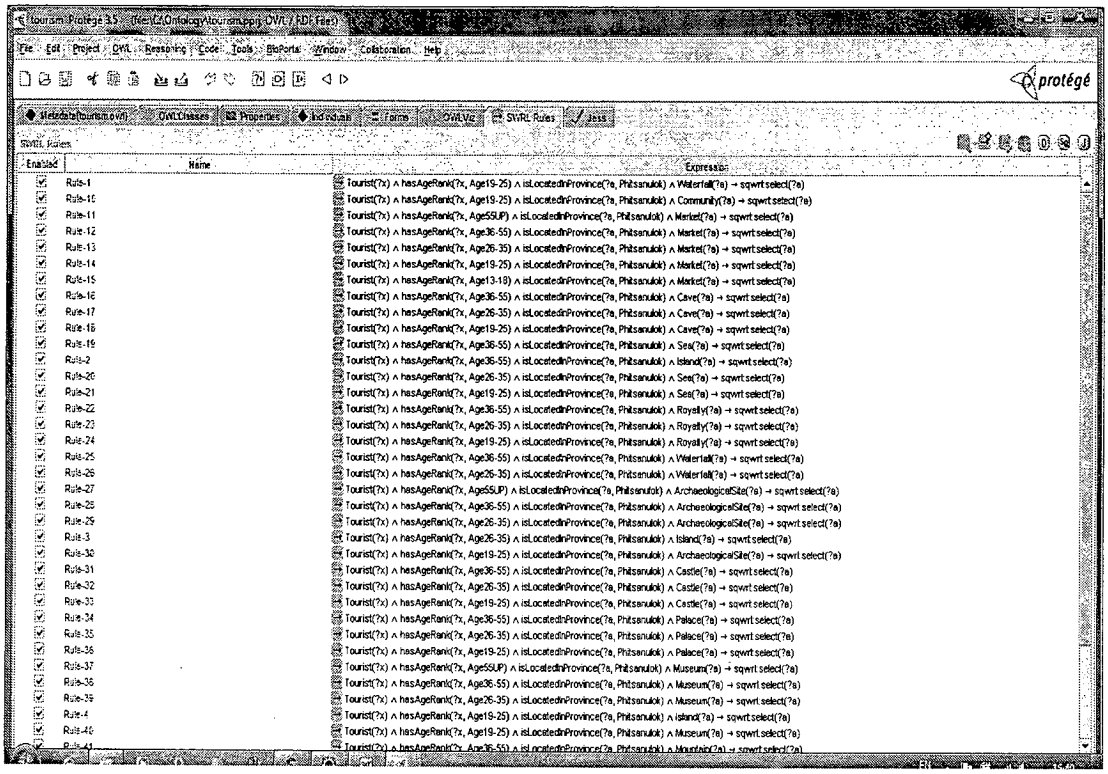
1. คลิก Individuals (หมายเลข 1) แล้วเลือกคลาสที่ต้องการป้อนข้อมูล
2. กดปุ่ม Create Instance (หมายเลข 2)
3. ตั้งชื่อข้อมูลหลังเครื่องหมาย # (หมายเลข 3)
4. ป้อนข้อมูลต่างๆ บริเวณหมายเลข 4



ภาพ 12 แสดงตัวอย่างหน้าจอการป้อนข้อมูลท่องเที่ยวผ่านทางโปรแกรม Protégé 3.5

โดยงานวิจัยนี้ได้นำเข้าข้อมูลพื้นฐานทั้งหมด 77 จังหวัด ประกอบด้วย ชื่อจังหวัด ชื่อเรียกเดิมหรือชื่อเรียกอื่น คำขวัญประจำจังหวัด ดอกไม้ประจำจังหวัด สัญลักษณ์ประจำจังหวัด ต้นไม้ประจำจังหวัด ละติจูด ลองติจูด และเว็บไซต์ประจำจังหวัด (ภาพ 12) สถานที่ท่องเที่ยว 77 จังหวัด จำนวน 3,850 รายการ โรงแรม 77 จังหวัดจำนวน 2,570 รายการ ร้านอาหาร 77 จังหวัดจำนวน 4,257 รายการ ร้านขายของฝาก 77 จังหวัดจำนวน 200 รายการ ร้านสินค้าหนึ่งตำบลหนึ่งผลิตภัณฑ์จำนวน 8,556 รายการและเทศกาล 77 จังหวัด จำนวน 77 รายการ

จากแนวความคิดเกี่ยวกับการท่องเที่ยวที่ได้อธิบายไว้แล้วในบทที่ 2 ทำให้สามารถสร้างกฎต่างๆ ที่ใช้ในการแนะนำข้อมูลท่องเที่ยว ซึ่งเขียนกฎด้วยภาษา SWRL พร้อมทดสอบกฎแต่ละกฎว่าสามารถทำงานได้อย่างถูกต้องด้วย Jess Inference Engine ที่ถูกติดตั้งในโปรแกรม Protégé 3.5 ดังแสดงในภาพ 13



ภาพ 13 แสดงกฎในภาษา SWRL ที่ใช้ในระบบแนะนำข้อมูลท่องเที่ยวในประเทศไทย

จากภาพ 13 แสดงกฎที่ใช้ในระบบแนะนำข้อมูลท่องเที่ยวในประเทศไทยของจังหวัดพิษณุโลก ซึ่งผู้วิจัยได้สร้างกฎไว้ทั้งหมด 93 กฎ โดยมี 5 ตัวแปร คือ นักท่องเที่ยว อายุ เพศ สถานที่ท่องเที่ยวและจังหวัดที่ต้องการไปเที่ยว ซึ่งภาษา SWRL นั้นจะถูกฝังไว้ในโปรแกรมที่พัฒนาด้วยภาษา JSP โดยใช้ Protégé API 3.5 ในการเชื่อมต่อไปยัง Jess Inference Engine กับออนโทโลจีท่องเที่ยวเพื่อให้ได้ข้อมูลท่องเที่ยวตามที่ผู้ใช้ต้องการ ดังแสดงรายละเอียดตามตาราง 8

ตาราง 8 แสดงกฎที่ใช้ในระบบแนะนำข้อมูลท่องเที่ยวในประเทศไทย

กฎข้อที่	กฎ	คำอธิบาย
1	Tourist(?x) ^ hasAgeRank(?x,Age36-55) ^ isLocatedInProvince(?a, "Province") ^ Island(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 36-55 ปีสามารถไปเที่ยวเกาะที่มีอยู่ในจังหวัดนั้นได้
2	Tourist(?x) ^ hasAgeRank(?x,Age26-35) ^ isLocatedInProvince(?a, "Province") ^	นักท่องเที่ยวที่มีอายุระหว่าง 26-35 ปีสามารถไปเที่ยวเกาะที่มีอยู่

ตาราง 8 (ต่อ)

กฎข้อที่	กฎ	คำอธิบาย
	Island(?a) -> sqwrl:select(?a)	ในจังหวัดนั้นได้
3	Tourist(?x) ^ hasAgeRank(?x,Age19-25) ^ isLocatedInProvince(?a, "Province") ^ Island(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 19-25 ปีสามารถไปเที่ยวเกาะที่มีอยู่ในจังหวัดนั้นได้
4	Tourist(?x) ^ hasAgeRank(?x,Age36-55) ^ isLocatedInProvince(?a, "Province") ^ Barrage(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 36-55 ปีสามารถไปเที่ยวเขื่อนที่มีอยู่ในจังหวัดนั้นได้
5	Tourist(?x) ^ hasAgeRank(?x,Age26-35) ^ isLocatedInProvince(?a, "Province") ^ Barrage(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 26-35 ปีสามารถไปเที่ยวเขื่อนที่มีอยู่ในจังหวัดนั้นได้
6	Tourist(?x) ^ hasAgeRank(?x,Age19-25) ^ isLocatedInProvince(?a, "Province") ^ Barrage(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 19-25 ปีสามารถไปเที่ยวเขื่อนที่มีอยู่ในจังหวัดนั้นได้
7	Tourist(?x) ^ hasAgeRank(?x,Age36-55) ^ isLocatedInProvince(?a, "Province") ^ Community(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 36-55 ปีสามารถไปเที่ยวชุมชนที่มีอยู่ในจังหวัดนั้นได้
8	Tourist(?x) ^ hasAgeRank(?x,Age26-35) ^ isLocatedInProvince(?a, "Province") ^ Community(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 26-35 ปีสามารถไปเที่ยวชุมชนที่มีอยู่ในจังหวัดนั้นได้
9	Tourist(?x) ^ hasAgeRank(?x,Age19-25) ^ isLocatedInProvince(?a, "Province") ^ Community(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 19-25 ปีสามารถไปเที่ยวชุมชนที่มีอยู่ในจังหวัดนั้นได้
10	Tourist(?x) ^ hasAgeRank(?x,Age55UP) ^ isLocatedInProvince(?a, "Province") ^ Market(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุตั้งแต่ 55 ปีขึ้นไปสามารถไปเที่ยวตลาดที่มีอยู่ในจังหวัดนั้นได้

ตาราง 8 (ต่อ)

กฎข้อที่	กฎ	คำอธิบาย
11	Tourist(?x) ^ hasAgeRank(?x,Age36-55) ^ isLocatedInProvince(?a, "Province") ^ Market(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 36-55 ปีสามารถไปเที่ยวตลาดที่มีอยู่ในจังหวัดนั้นได้
12	Tourist(?x) ^ hasAgeRank(?x,Age26-35) ^ isLocatedInProvince(?a, "Province") ^ Market(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 26-35 ปีสามารถไปเที่ยวตลาดที่มีอยู่ในจังหวัดนั้นได้
13	Tourist(?x) ^ hasAgeRank(?x,Age19-25) ^ isLocatedInProvince(?a, "Province") ^ Market(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 19-25 ปีสามารถไปเที่ยวตลาดที่มีอยู่ในจังหวัดนั้นได้
14	Tourist(?x) ^ hasAgeRank(?x,Age13-18) ^ isLocatedInProvince(?a, "Province") ^ Market(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 13-18 ปีสามารถไปเที่ยวตลาดที่มีอยู่ในจังหวัดนั้นได้
15	Tourist(?x) ^ hasAgeRank(?x,Age36-55) ^ isLocatedInProvince(?a, "Province") ^ Cave(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 36-55 ปีสามารถไปเที่ยวถ้ำที่มีอยู่ในจังหวัดนั้นได้
16	Tourist(?x) ^ hasAgeRank(?x,Age26-35) ^ isLocatedInProvince(?a, "Province") ^ Cave(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 26-35 ปีสามารถไปเที่ยวถ้ำที่มีอยู่ในจังหวัดนั้นได้
17	Tourist(?x) ^ hasAgeRank(?x,Age19-25) ^ isLocatedInProvince(?a, "Province") ^ Cave(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 19-25 ปีสามารถไปเที่ยวถ้ำที่มีอยู่ในจังหวัดนั้นได้
18	Tourist(?x) ^ hasAgeRank(?x,Age36-55) ^ isLocatedInProvince(?a, "Province") ^ Sea(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 36-55 ปีสามารถไปเที่ยวทะเลที่มีอยู่ในจังหวัดนั้นได้
19	Tourist(?x) ^ hasAgeRank(?x,Age26-35) ^ isLocatedInProvince(?a, "Province") ^ Sea(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 26-35 ปีสามารถไปเที่ยวทะเลที่มีอยู่ในจังหวัดนั้นได้

ตาราง 8 (ต่อ)

กฎข้อที่	กฎ	คำอธิบาย
20	Tourist(?x) ^ hasAgeRank(?x,Age19-25) ^ isLocatedInProvince(?a, "Province") ^ Sea(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 19-25 ปีสามารถไปเที่ยวทะเลที่มีอยู่ในจังหวัดนั้นได้
21	Tourist(?x) ^ hasAgeRank(?x,Age36-55) ^ isLocatedInProvince(?a, "Province") ^ Royalty(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 36-55 ปีสามารถไปเที่ยวสถานที่ทางราชการที่มีอยู่ในจังหวัดนั้นได้
22	Tourist(?x) ^ hasAgeRank(?x,Age26-35) ^ isLocatedInProvince(?a, "Province") ^ Royalty(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 26-35 ปีสามารถไปเที่ยวสถานที่ทางราชการที่มีอยู่ในจังหวัดนั้นได้
23	Tourist(?x) ^ hasAgeRank(?x,Age19-25) ^ isLocatedInProvince(?a, "Province") ^ Royalty(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 19-25 ปีสามารถไปเที่ยวสถานที่ทางราชการที่มีอยู่ในจังหวัดนั้นได้
24	Tourist(?x) ^ hasAgeRank(?x,Age36-55) ^ isLocatedInProvince(?a, "Province") ^ Waterfall(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 36-55 ปีสามารถไปเที่ยวน้ำตกที่มีอยู่ในจังหวัดนั้นได้
25	Tourist(?x) ^ hasAgeRank(?x,Age26-35) ^ isLocatedInProvince(?a, "Province") ^ Waterfall(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 26-35 ปีสามารถไปเที่ยวน้ำตกที่มีอยู่ในจังหวัดนั้นได้
26	Tourist(?x) ^ hasAgeRank(?x,Age19-25) ^ isLocatedInProvince(?a, "Province") ^ Waterfall(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 19-25 ปีสามารถไปเที่ยวน้ำตกที่มีอยู่ในจังหวัดนั้นได้
27	Tourist(?x) ^ hasAgeRank(?x,Age55UP) ^ isLocatedInProvince(?a, "Province") ^ ArchaeologicalSite(?a)-> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุตั้งแต่ 55 ปีขึ้นไปสามารถไปเที่ยวโบราณสถานที่มีอยู่ในจังหวัดนั้นได้
28	Tourist(?x) ^ hasAgeRank(?x,Age36-55) ^ isLocatedInProvince(?a, "Province") ^	นักท่องเที่ยวที่มีอายุระหว่าง 36-55 ปีสามารถไปเที่ยวโบราณ

ตาราง 8 (ต่อ)

กฎข้อที่	กฎ	คำอธิบาย
	ArchaeologicalSite(?a) -> sqwrl:select(?a)	สถานที่มีอยู่ในจังหวัดนั้นได้
29	Tourist(?x) ^ hasAgeRank(?x,Age26-35) ^ isLocatedInProvince(?a, "Province") ^ ArchaeologicalSite(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 26-35 ปีสามารถไปเที่ยวโบราณสถานที่มีอยู่ในจังหวัดนั้นได้
30	Tourist(?x) ^ hasAgeRank(?x,Age19-25) ^ isLocatedInProvince(?a, "Province") ^ ArchaeologicalSite(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 19-25 ปีสามารถไปเที่ยวโบราณสถานที่มีอยู่ในจังหวัดนั้นได้
31	Tourist(?x) ^ hasAgeRank(?x,Age36-55) ^ isLocatedInProvince(?a, "Province") ^ Castle(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 36-55 ปีสามารถไปเที่ยวปราสาทที่มีอยู่ในจังหวัดนั้นได้
32	Tourist(?x) ^ hasAgeRank(?x,Age26-35) ^ isLocatedInProvince(?a, "Province") ^ Castle(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 26-35 ปีสามารถไปเที่ยวปราสาทที่มีอยู่ในจังหวัดนั้นได้
33	Tourist(?x) ^ hasAgeRank(?x,Age19-25) ^ isLocatedInProvince(?a, "Province") ^ Castle(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 19-25 ปีสามารถไปเที่ยวปราสาทที่มีอยู่ในจังหวัดนั้นได้
34	Tourist(?x) ^ hasAgeRank(?x,Age36-55) ^ isLocatedInProvince(?a, "Province") ^ Palace(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 36-55 ปีสามารถไปเที่ยวพระราชวังที่มีอยู่ในจังหวัดนั้นได้
35	Tourist(?x) ^ hasAgeRank(?x,Age26-35) ^ isLocatedInProvince(?a, "Province") ^ Palace(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 26-35 ปีสามารถไปเที่ยวพระราชวังที่มีอยู่ในจังหวัดนั้นได้

ตาราง 8 (ต่อ)

กฎข้อที่	กฎ	คำอธิบาย
36	Tourist(?x) ^ hasAgeRank(?x,Age19-25) ^ isLocatedInProvince(?a, "Province") ^ Palace(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 19-25 ปีสามารถไปเที่ยวพระราชวังที่มีอยู่ในจังหวัดนั้นได้
37	Tourist(?x) ^ hasAgeRank(?x,Age55UP) ^ isLocatedInProvince(?a, "Province") ^ Museum(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุตั้งแต่ 55 ปีขึ้นไปสามารถไปเที่ยวพิพิธภัณฑ์ที่มีอยู่ในจังหวัดนั้นได้
38	Tourist(?x) ^ hasAgeRank(?x,Age36-55) ^ isLocatedInProvince(?a, "Province") ^ Museum(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 36-55 ปีสามารถไปเที่ยวพิพิธภัณฑ์ที่มีอยู่ในจังหวัดนั้นได้
39	Tourist(?x) ^ hasAgeRank(?x,Age26-35) ^ isLocatedInProvince(?a, "Province") ^ Museum(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 26-35 ปีสามารถไปเที่ยวพิพิธภัณฑ์ที่มีอยู่ในจังหวัดนั้นได้
40	Tourist(?x) ^ hasAgeRank(?x,Age19-25) ^ isLocatedInProvince(?a, "Province") ^ Museum(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 19-25 ปีสามารถไปเที่ยวพิพิธภัณฑ์ที่มีอยู่ในจังหวัดนั้นได้
41	Tourist(?x) ^ hasAgeRank(?x,Age36-55) ^ isLocatedInProvince(?a, "Province") ^ Mountain(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 36-55 ปีสามารถไปเที่ยวภูเขาที่มีอยู่ในจังหวัดนั้นได้
42	Tourist(?x) ^ hasAgeRank(?x,Age26-35) ^ isLocatedInProvince(?a, "Province") ^ Mountain(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 26-35 ปีสามารถไปเที่ยวภูเขาที่มีอยู่ในจังหวัดนั้นได้
43	Tourist(?x) ^ hasAgeRank(?x,Age19-25) ^ isLocatedInProvince(?a, "Province") ^ Mountain(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 19-25 ปีสามารถไปเที่ยวภูเขาที่มีอยู่ในจังหวัดนั้นได้
44	Tourist(?x) ^ hasAgeRank(?x,Age36-55) ^ isLocatedInProvince(?a, "Province") ^	นักท่องเที่ยวที่มีอายุระหว่าง 36-55 ปีสามารถไปเที่ยวโรงละคร

ตาราง 8 (ต่อ)

กฎข้อที่	กฎ	คำอธิบาย
	Circus(?a) -> sqwrl:select(?a)	สัตว์ที่มีอยู่ในจังหวัดนั้นได้
45	Tourist(?x) ^ hasAgeRank(?x,Age26-35) ^ isLocatedInProvince(?a, "Province") ^ Circus(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 26-35 ปีสามารถไปเที่ยวโรงละครสัตว์ที่มีอยู่ในจังหวัดนั้นได้
46	Tourist(?x) ^ hasAgeRank(?x,Age19-25) ^ isLocatedInProvince(?a, "Province") ^ Circus(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 19-25 ปีสามารถไปเที่ยวโรงละครสัตว์ที่มีอยู่ในจังหวัดนั้นได้
47	Tourist(?x) ^ hasAgeRank(?x,Age0-12) ^ isLocatedInProvince(?a, "Province") ^ Circus(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 0-12 ปีสามารถไปเที่ยวโรงละครสัตว์ที่มีอยู่ในจังหวัดนั้นได้
48	Tourist(?x) ^ hasAgeRank(?x,Age36-55) ^ isLocatedInProvince(?a, "Province") ^ Theatre(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 36-55 ปีสามารถไปเที่ยวโรงหนังที่มีอยู่ในจังหวัดนั้นได้
49	Tourist(?x) ^ hasAgeRank(?x,Age26-35) ^ isLocatedInProvince(?a, "Province") ^ Theatre(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 26-35 ปีสามารถไปเที่ยวโรงหนังที่มีอยู่ในจังหวัดนั้นได้
50	Tourist(?x) ^ hasAgeRank(?x,Age19-25) ^ isLocatedInProvince(?a, "Province") ^ Theatre(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 19-25 ปีสามารถไปเที่ยวโรงหนังที่มีอยู่ในจังหวัดนั้นได้
51	Tourist(?x) ^ hasAgeRank(?x,Age0-12) ^ isLocatedInProvince(?a, "Province") ^ Theatre(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 0-12 ปีสามารถไปเที่ยวโรงหนังที่มีอยู่ในจังหวัดนั้นได้
52	Tourist(?x) ^ hasAgeRank(?x,Age55UP) ^ isLocatedInProvince(?a, "Province") ^ Temple(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุตั้งแต่ 55 ปีขึ้นไปสามารถไปเที่ยววัดที่มีอยู่ในจังหวัดนั้นได้
53	Tourist(?x) ^ hasAgeRank(?x,Age36-55)	นักท่องเที่ยวที่มีอายุระหว่าง 36-

ตาราง 8 (ต่อ)

กฎข้อที่	กฎ	คำอธิบาย
53	^ isLocatedInProvince(?a, "Province") ^ Temple(?a) -> sqwrl:select(?a)	55 ปีสามารถไปเที่ยววัดที่มีอยู่ในจังหวัดนั้นได้
54	Tourist(?x) ^ hasAgeRank(?x,Age26-35) ^ isLocatedInProvince(?a, "Province") ^ Temple(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 26-35 ปีสามารถไปเที่ยววัดที่มีอยู่ในจังหวัดนั้นได้
55	Tourist(?x) ^ hasAgeRank(?x,Age19-25) ^ isLocatedInProvince(?a, "Province") ^ Temple(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 19-25 ปีสามารถไปเที่ยววัดที่มีอยู่ในจังหวัดนั้นได้
56	Tourist(?x) ^ hasAgeRank(?x,Age55UP) ^ isLocatedInProvince(?a, "Province") ^ CultureCenter(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุตั้งแต่ 55 ปีขึ้นไปสามารถไปเที่ยวศูนย์วัฒนธรรมที่มีอยู่ในจังหวัดนั้นได้
57	Tourist(?x) ^ hasAgeRank(?x,Age36-55) ^ isLocatedInProvince(?a, "Province") ^ CultureCenter(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 36-55 ปีสามารถไปเที่ยวศูนย์วัฒนธรรมที่มีอยู่ในจังหวัดนั้นได้
58	Tourist(?x) ^ hasAgeRank(?x,Age26-35) ^ isLocatedInProvince(?a, "Province") ^ CultureCenter(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 26-35 ปีสามารถไปเที่ยวศูนย์วัฒนธรรมที่มีอยู่ในจังหวัดนั้นได้
59	Tourist(?x) ^ hasAgeRank(?x,Age19-25) ^ isLocatedInProvince(?a, "Province") ^ CultureCenter(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 19-25 ปีสามารถไปเที่ยวศูนย์วัฒนธรรมที่มีอยู่ในจังหวัดนั้นได้
60	Tourist(?x) ^ hasAgeRank(?x,Age36-55) ^ isLocatedInProvince(?a, "Province") ^ Amusement(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 36-55 ปีสามารถไปเที่ยวสถานบันเทิงที่มีอยู่ในจังหวัดนั้นได้
61	Tourist(?x) ^ hasAgeRank(?x,Age26-35) ^ isLocatedInProvince(?a, "Province") ^ Amusement(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 26-35 ปีสามารถไปเที่ยวสถานบันเทิงที่มีอยู่ในจังหวัดนั้นได้

ตาราง 8 (ต่อ)

กฎข้อที่	กฎ	คำอธิบาย
62	Tourist(?x) ^ hasAgeRank(?x,Age19-25) ^ isLocatedInProvince(?a, "Province") ^ Amusement(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 19-25 ปีสามารถไปเที่ยวสถานบันเทิงที่มีอยู่ในจังหวัดนั้นได้
63	Tourist(?x) ^ hasAgeRank(?x,Age13-18) ^ isLocatedInProvince(?a, "Province") ^ Amusement(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 13-18 ปีสามารถไปเที่ยวสถานบันเทิงที่มีอยู่ในจังหวัดนั้นได้
64	Tourist(?x) ^ hasAgeRank(?x,Age0-12) ^ isLocatedInProvince(?a, "Province") ^ Amusement(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 0-12 ปีสามารถไปเที่ยวสถานบันเทิงที่มีอยู่ในจังหวัดนั้นได้
65	Tourist(?x) ^ hasAgeRank(?x,Age36-55) ^ hasGender(?x,Male) ^ isLocatedInProvince(?a, "Province") ^ Nightlife(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 36-55 ปีและเป็นเพศชายสามารถไปเที่ยวสถานบันเทิงยามราตรีที่มีอยู่ในจังหวัดนั้นได้
66	Tourist(?x) ^ hasAgeRank(?x,Age26-35) ^ hasGender(?x,Male) ^ isLocatedInProvince(?a, "Province") ^ Nightlife(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 26-35 ปีและเป็นเพศชายสามารถไปเที่ยวสถานบันเทิงยามราตรีที่มีอยู่ในจังหวัดนั้นได้
67	Tourist(?x) ^ hasAgeRank(?x,Age19-25) ^ hasGender(?x,Male) ^ isLocatedInProvince(?a, "Province") ^ Nightlife(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 19-25 ปีและเป็นเพศชายสามารถไปเที่ยวสถานบันเทิงยามราตรีที่มีอยู่ในจังหวัดนั้นได้
68	Tourist(?x) ^ hasAgeRank(?x,Age36-55) ^ isLocatedInProvince(?a, "Province") ^ Stadium(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 36-55 ปีสามารถไปเที่ยวสนามกีฬาที่มีอยู่ในจังหวัดนั้นได้
69	Tourist(?x) ^ hasAgeRank(?x,Age26-35) ^ isLocatedInProvince(?a, "Province") ^	นักท่องเที่ยวที่มีอายุระหว่าง 26-35 ปีสามารถไปเที่ยวสนามกีฬา

ตาราง 8 (ต่อ)

กฎข้อที่	กฎ	คำอธิบาย
	Stadium(?a) -> sqwrl:select(?a)	ที่มีอยู่ในจังหวัดนั้นได้
70	Tourist(?x) ^ hasAgeRank(?x,Age19-25) ^ isLocatedInProvince(?a, "Province") ^ Stadium(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 19-25 ปีสามารถไปเที่ยวสนามกีฬาที่มีอยู่ในจังหวัดนั้นได้
71	Tourist(?x) ^ hasAgeRank(?x,Age36-55) ^ isLocatedInProvince(?a, "Province") ^ Zoo(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 36-55 ปีสามารถไปเที่ยวสวนสัตว์ที่มีอยู่ในจังหวัดนั้นได้
72	Tourist(?x) ^ hasAgeRank(?x,Age26-35) ^ isLocatedInProvince(?a, "Province") ^ Zoo(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 26-35 ปีสามารถไปเที่ยวสวนสัตว์ที่มีอยู่ในจังหวัดนั้นได้
73	Tourist(?x) ^ hasAgeRank(?x,Age19-25) ^ isLocatedInProvince(?a, "Province") ^ Zoo(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 19-25 ปีสามารถไปเที่ยวสวนสัตว์ที่มีอยู่ในจังหวัดนั้นได้
74	Tourist(?x) ^ hasAgeRank(?x,Age0-12) ^ isLocatedInProvince(?a, "Province") ^ Zoo(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 0-12 ปีสามารถไปเที่ยวสวนสัตว์ที่มีอยู่ในจังหวัดนั้นได้
75	Tourist(?x) ^ hasAgeRank(?x,Age36-55) ^ isLocatedInProvince(?a, "Province") ^ NationalPark(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 36-55 ปีสามารถไปเที่ยวสวนสาธารณะที่มีอยู่ในจังหวัดนั้นได้
76	Tourist(?x) ^ hasAgeRank(?x,Age26-35) ^ isLocatedInProvince(?a, "Province") ^ NationalPark(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 26-35 ปีสามารถไปเที่ยวสวนสาธารณะที่มีอยู่ในจังหวัดนั้นได้

ตาราง 8 (ต่อ)

กฎข้อที่	กฎ	คำอธิบาย
77	Tourist(?x) ^ hasAgeRank(?x,Age19-25) ^ isLocatedInProvince(?a, "Province") ^ NationalPark(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 19-25 ปีสามารถไปเที่ยวสวนสาธารณะที่มีอยู่ในจังหวัดนั้นได้
78	Tourist(?x) ^ hasAgeRank(?x,Age55UP) ^ isLocatedInProvince(?a, "Province") ^ Monastery(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุตั้งแต่ 55 ปีขึ้นไปสามารถไปเที่ยวสำนักสงฆ์ที่มีอยู่ในจังหวัดนั้นได้
79	Tourist(?x) ^ hasAgeRank(?x,Age36-55) ^ isLocatedInProvince(?a, "Province") ^ Monastery(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 36-55 ปีสามารถไปเที่ยวสำนักสงฆ์ที่มีอยู่ในจังหวัดนั้นได้
80	Tourist(?x) ^ hasAgeRank(?x,Age26-35) ^ isLocatedInProvince(?a, "Province") ^ Monastery(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 26-35 ปีสามารถไปเที่ยวสำนักสงฆ์ที่มีอยู่ในจังหวัดนั้นได้
81	Tourist(?x) ^ hasAgeRank(?x,Age19-25) ^ isLocatedInProvince(?a, "Province") ^ Monastery(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 19-25 ปีสามารถไปเที่ยวสำนักสงฆ์ที่มีอยู่ในจังหวัดนั้นได้
82	Tourist(?x) ^ hasAgeRank(?x,Age55UP) ^ isLocatedInProvince(?a, "Province") ^ ShoppingCenter(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุตั้งแต่ 55 ปีขึ้นไปสามารถไปเที่ยวห้างสรรพสินค้าที่มีอยู่ในจังหวัดนั้นได้
83	Tourist(?x) ^ hasAgeRank(?x,Age36-55) ^ isLocatedInProvince(?a, "Province") ^ ShoppingCenter(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 36-55 ปีสามารถไปเที่ยวห้างสรรพสินค้าที่มีอยู่ในจังหวัดนั้นได้
84	Tourist(?x) ^ hasAgeRank(?x,Age26-35) ^ isLocatedInProvince(?a, "Province") ^ ShoppingCenter(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 26-35 ปีสามารถไปเที่ยวห้างสรรพสินค้าที่มีอยู่ในจังหวัดนั้นได้

ตาราง 8 (ต่อ)

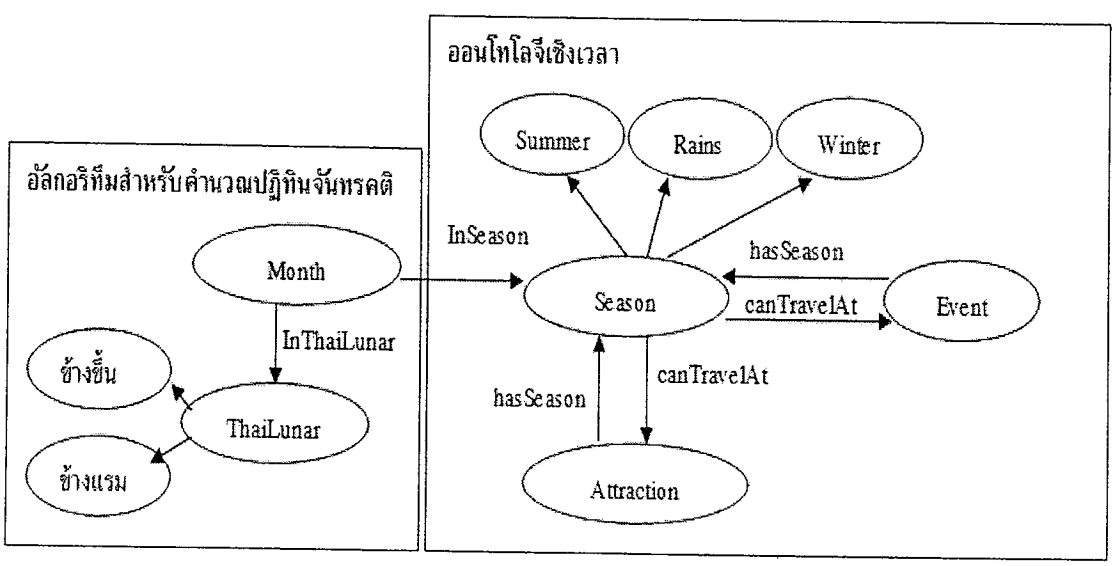
กฎข้อที่	กฎ	คำอธิบาย
85	Tourist(?x) ^ hasAgeRank(?x,Age19-25) ^ isLocatedInProvince(?a, "Province") ^ ShoppingCenter(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 19-25 ปีสามารถไปเที่ยวห้างสรรพสินค้าที่มีอยู่ในจังหวัดนั้นได้
86	Tourist(?x) ^ hasAgeRank(?x,Age13-18) ^ isLocatedInProvince(?a, "Province") ^ ShoppingCenter(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 13-18 ปีสามารถไปเที่ยวห้างสรรพสินค้าที่มีอยู่ในจังหวัดนั้นได้
87	Tourist(?x) ^ hasAgeRank(?x,Age0-12) ^ isLocatedInProvince(?a, "Province") ^ ShoppingCenter(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 0-12 ปีสามารถไปเที่ยวห้างสรรพสินค้าที่มีอยู่ในจังหวัดนั้นได้
88	Tourist(?x) ^ hasAgeRank(?x,Age36-55) ^ isLocatedInProvince(?a, "Province") ^ Monument(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 36-55 ปีสามารถไปเที่ยวอนุสาวรีย์ที่มีอยู่ในจังหวัดนั้นได้
89	Tourist(?x) ^ hasAgeRank(?x,Age26-35) ^ isLocatedInProvince(?a, "Province") ^ Monument(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 26-35 ปีสามารถไปเที่ยวอนุสาวรีย์ที่มีอยู่ในจังหวัดนั้นได้
90	Tourist(?x) ^ hasAgeRank(?x,Age19-25) ^ isLocatedInProvince(?a, "Province") ^ Monument(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 19-25 ปีสามารถไปเที่ยวอนุสาวรีย์ที่มีอยู่ในจังหวัดนั้นได้
91	Tourist(?x) ^ hasAgeRank(?x,Age36-55) ^ isLocatedInProvince(?a, "Province") ^ Reservoir(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 36-55 ปีสามารถไปเที่ยวอ่างเก็บน้ำที่มีอยู่ในจังหวัดนั้นได้
92	Tourist(?x) ^ hasAgeRank(?x,Age26-35) ^ isLocatedInProvince(?a, "Province") ^	นักท่องเที่ยวที่มีอายุระหว่าง 26-35 ปีสามารถไปเที่ยวอ่างเก็บน้ำ

ตาราง 8 (ต่อ)

กฎข้อที่	กฎ	คำอธิบาย
	Reservoir(?a) -> sqwrl:select(?a)	ที่มีอยู่ในจังหวัดนั้นได้
93	Tourist(?x) ^ hasAgeRank(?x,Age19-25) ^ isLocatedInProvince(?a, "Province") ^ Reservoir(?a) -> sqwrl:select(?a)	นักท่องเที่ยวที่มีอายุระหว่าง 19-25 ปีสามารถไปเที่ยวอ่างเก็บน้ำที่มีอยู่ในจังหวัดนั้นได้

2.3 การออกแบบและพัฒนาออนโทโลจีเชิงเวลา

ข้อมูลท่องเที่ยวตามฤดูกาลหรือช่วงเวลาและวันสำคัญตามประเพณีของไทยนั้นเป็นสิ่งที่นักท่องเที่ยวควรจะทราบ เนื่องจากจะทำให้ไม่พลาดเทศกาลสำคัญของจังหวัดนั้นๆ และท่องเที่ยวได้ด้วยความสุข สนุกสนาน ดังนั้นผู้วิจัยจึงใช้ฤดูกาลที่มีในประเทศไทย คือ ฤดูร้อน ฤดูฝน และฤดูหนาว ดังได้กล่าวรายละเอียดไว้แล้วในหัวข้อที่ 3 หลักการออนโทโลจีเชิงเวลา บทที่ 2 รวมถึงการพัฒนาอัลกอริทึมสำหรับคำนวณปฏิทินจันทรคติ ซึ่งจะอยู่ในหัวข้อการแนะนำข้อมูลท่องเที่ยวตามช่วงเวลาและบริเวณใกล้เคียงเพื่อแนะนำวันสำคัญต่างๆ ที่ในแต่ละปีจะไม่ตรงนั้น ดังแสดงความสัมพันธ์ในภาพ 14



ภาพ 14 แสดงออนโทโลจีเชิงเวลา

จากภาพ 14 แสดงการออกแบบออนโทโลจีเชิงเวลา ซึ่งสถานที่ท่องเที่ยวหรือเทศกาลที่ผู้ใช้สามารถท่องเที่ยวได้ตามฤดูกาลที่มีอยู่ในประเทศไทย (ฤดูร้อน ฤดูฝนและฤดูหนาว) ตามวันที่ผู้ใช้ได้ระบุ (เดือน) โดยวันที่ดังกล่าวนั้นจะเชื่อมโยงไปยังปฏิทินจันทรคติ (Thai Lunar) ที่จะใช้บอกวันข้างขึ้นหรือข้างแรมและวันพระ โดยผู้ใช้จะต้องระบุวันที่เดินทางท่องเที่ยว จำนวนวันท่องเที่ยว ซึ่งออนโทโลจีเชิงเวลานี้จะใช้กับคลาสสถานที่ท่องเที่ยวและเทศกาล

3. **ด้านอัลกอริทึม**

3.1 **การประยุกต์ใช้และพัฒนาอัลกอริทึมเนมแมทชิ่งสำหรับกรองข้อมูล**

งานวิจัยนี้ได้สร้างอัลกอริทึม Name Variation Matching เพื่อสร้างชื่อที่สามารถเขียนได้หลากหลายหรือคล้ายคลึงกันของชื่อจังหวัดในประเทศไทยและจัดเก็บลงฐานข้อมูล ซึ่งการทำงานของ Name Variation Matching มีดังต่อไปนี้

1. ตรวจสอบว่าเป็นภาษาไทยหรือภาษาอังกฤษ

2. กรณีเป็นภาษาไทยสร้างชื่อจากความคล้ายคลึงของพยัญชนะและตัวสะกด แต่หากเป็นภาษาอังกฤษสร้างชื่อจากความคล้ายคลึงของตัวอักษร (Snae, et al., 2007; Snae and Brueckner, 2009) และใช้อัลกอริทึม Name Variation Matching (ภาพ 15) ทำงาน ผลลัพธ์ที่ได้จะแสดงดังตาราง 9

```
if (varname.charAt(i) == 'ฃ') {
   namevar = namevar.substring(0,namevar.indexOf(varname.charAt(i))) + 'ข' +
   Save2MySql.update(con, "insert tb_genvar (keyword,variation,level) values ('" + varname.trim()
}
else if (varname.charAt(i) == 'ค') {
   temp = namevar.substring(0,namevar.indexOf(varname.charAt(i)));
   namevar = temp + 'ศ' + varname.substring(i+1);
   Save2MySql.update(con, "insert tb_genvar (keyword,variation,level) values ('" + varname.trim()
   namevar = temp + 'ฆ' + varname.substring(i+1);
   Save2MySql.update(con, "insert tb_genvar (keyword,variation,level) values ('" + varname.trim()
}
else if (varname.charAt(i) == 'ศ') {
   temp = namevar.substring(0,namevar.indexOf(varname.charAt(i)));
```

ภาพ 15 แสดงตัวอย่างอัลกอริทึม Name Variation Matching

ตาราง 9 แสดงตัวอย่างการทำงานของอัลกอริทึม Name Variation Matching

ลำดับ	คำค้นหา	ตำแหน่งตัวอักษร	ตัวอักษรที่เปลี่ยน	เปลี่ยนเป็น	คำที่ได้
1.	พิษณุโลก	1	พ	ภ	ภิษณุโลก
2.	พิษณุโลก	2	ษ	ศ	พิศณุโลก
3.	ภิษณุโลก	2	ษ	ศ	ภิศณุโลก
4.	พิษณุโลก	3	ณ	น	พิษนุโลก
5.	ภิษณุโลก	3	ณ	น	ภิษนุโลก
6.	พิศณุโลก	3	ณ	น	พิศนุโลก
7.	ภิศณุโลก	3	ณ	น	ภิศนุโลก

จากนั้นใช้อัลกอริทึม ISG ดังตัวอย่างอัลกอริทึมในภาพ 16 เพื่อหาค่าความคล้ายคลึงของชื่อจังหวัดที่มีการสะกดผิดหรือเขียนผิดซึ่งปรากฏอยู่ในเว็บไซต์ท่องเที่ยวต่างๆ บนอินเทอร์เน็ต โดยจะช่วยให้ชื่อที่มีการเขียนผิดหรือสะกดผิดและปรากฏอยู่ในข้อมูลท่องเที่ยวสามารถถูกสืบค้นพบในอินเทอร์เน็ตได้ เช่น ถ้าเราจะค้นหาคำว่า พิษณุโลก อัลกอริทึมจะนำคำต่างๆ ที่ได้จากอัลกอริทึม Name Variation Matching มาทำการคัดกรองโดยใช้ค่า ISG ที่มีความคล้ายคลึงมากกว่าหรือเท่ากับ 0.5 ขึ้นไปในการไปสืบค้นข้อมูลท่องเที่ยวในเว็บไซต์กูเกิ้ล โดยใช้อัลกอริทึม Web Crawler ผลลัพธ์ที่จากการใช้อัลกอริทึม ISG แสดงในตาราง 10 ซึ่งอัลกอริทึม ISG มีหลักการทำงานดังต่อไปนี้

1. รับชื่อ 2 ชื่อและคำนวณหาความยาวของชื่อทั้งสอง โดยยึดชื่อที่มีความยาวสูงสุดเป็นหลัก

2. กำหนดค่าเริ่มต้นของตัวชี้ตำแหน่งอักษร ค่าความเหมือนและค่าความแตกต่าง ให้มีค่าเป็น ศูนย์

3. ตรวจสอบตัวชี้ตำแหน่งอักษรกับความยาวสูงสุดของชื่อ หากพบว่าตัวชี้ตำแหน่งมากกว่าความยาวสูงสุดของชื่อแล้วให้ไปทำงานข้อที่ 8

4. ถ้าตัวอักษรของชื่อทั้งสองเหมือนกันให้เพิ่มค่าความเหมือนในตำแหน่งใดๆ

5. ถ้าตัวอักษรของชื่อทั้งสองไม่เหมือนกันให้เพิ่มค่าความแตกต่าง

6. เพิ่มค่าตัวชี้ตำแหน่งอักษร

7. กลับไปยังข้อที่ 3

8. คำนวณหาค่า ISG จากสมการ 2 (บทที่ 2)

ตาราง 10 แสดงตัวอย่างการคำนวณชื่อจังหวัดพิษณุโลกด้วยอัลกอริทึม ISG

ลำดับ	ชื่อที่คล้ายคลึงกับ จังหวัดพิษณุโลก	จำนวนตัวอักษรที่เหมือนกัน (I)	จำนวนตัวอักษรที่แตกต่างกัน (D)	ค่า ISG I/(I+D)	จำนวนรายการที่พบ (กูเกิ้ล, กันยายน 2554)
1.	พิษณุโลก	8	0	1.0000	5,470,000
2.	พิษะณุโลก	8	1	0.8889	110,000
3.	พิดษะณุโลก	8	2	0.8000	76,200
4.	พิดสะณุโลก	7	3	0.7000	2,290,000
5.	พิจสะณุโลก	7	3	0.7000	2,200,000
6.	พิษนุโลก	7	1	0.8750	931,000
7.	พิตสะณุโลก	7	3	0.7000	106,000
8.	พิจสะนุโลก	6	4	0.6000	10,100,000
9.	พิสสะนุโลก	6	4	0.6000	1,230,000
10.	พิดโลก	5	3	0.6250	5,770,000
11.	ภีฏสะนุโลก	4	6	0.4000	3,000,000
12.	ภีดศะนุโลก	4	6	0.4000	197,000
13.	พีดสะนุโรก	4	6	0.4000	15,300,000
14.	พีดศะนุโรก	4	6	0.4000	1,200,000
15.	พีดโรก	3	5	0.3750	66,700,000

ที่มา: นฤพนธ์ พนาวงศ์ และจักรกฤษณ์ เสน่ห์ นมะหุต, 2556ก

จากตาราง 10 แสดงตัวอย่างการคำนวณชื่อจังหวัดพิษณุโลกที่มีการเขียนหลากหลายแบบตามหลักการออกเสียงในภาษาไทยและภาษาอังกฤษ พร้อมแสดงตัวอย่างวิธีการคำนวณค่า ISG ระหว่างชื่อ 2 ชื่อในภาษาไทย โดยการเปรียบเทียบจะใช้วิธีการเปรียบเทียบจากชื่อที่มีจำนวนตัวอักษรมากที่สุด ตัวอย่างเช่น พิษณุโลก กับ พีดสะนุโรก มีตัวอักษรที่เหมือนกัน 4 ตัว คือ พ สระอุ โ ก และมีตัวอักษรต่างกัน 6 ตัว คือ สระอี ด ส สระอะ น ร สำหรับลำดับที่ 11 - 15 จะไม่ถูกนำไปใช้สืบค้นข้อมูลในเว็บไซต์กูเกิ้ลเนื่องจากค่า ISG น้อยกว่า 0.5

```
j = 0;
for (i = 0; i < len1; ++i) {
    if (compchar(var1.substring(i,i+1),var2.substring(j,j+1
        same++;
        ++j;
    }
    else {
        for (k = j+1; k < len2 && !compchar(var1.substring(
        if ( k < len2) {
            same++;
            diff += (k - j);
            j = k + 1;
        }
    }
}
```

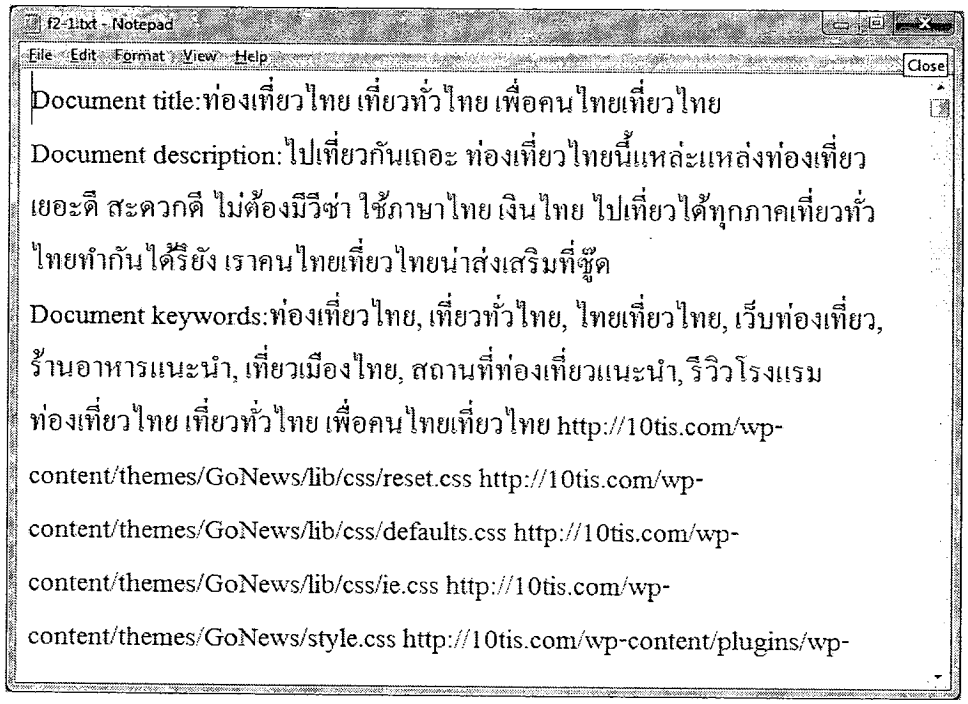
ภาพ 16 แสดงตัวอย่างอัลกอริทึม ISG

3.2 การพัฒนาอัลกอริทึม Web Crawler ในการดึงข้อมูลในหน้าเว็บไซต์

งานวิจัยนี้ผู้วิจัยพัฒนาอัลกอริทึม Web Crawler ในการเก็บข้อมูลเว็บไซต์ต่างๆ เช่น เว็บไซต์ http://10tis.com โดยใช้ Jericho API สำหรับดึงรายละเอียดจากเว็บไซต์พร้อมตัด HTML Tags ดังตัวอย่างโปรแกรมในภาพ 17 จากนั้นจัดเก็บเป็นไฟล์นามสกุล .txt ดังเนื้อหาในภาพ 18

```
String sourceUrlString = "http://10tis.com";
if (sourceUrlString.indexOf(':')==-1) sourceUrlString="file:"+sourceUrlString;
MicrosoftConditionalCommentTagTypes.register();
PHPTagTypes.register();
PHPTagTypes.PHP_SHORT.deregister();
MasonTagTypes.register();
Source source=new Source(new URL(sourceUrlString));
source.fullSequentialParse();
BufferedWriter fileout = new BufferedWriter(new FileWriter(filename));
fileout.write("Document title:");
String title=getTitle(source);
```

ภาพ 17 แสดงตัวอย่างการดึงเนื้อหาจากเว็บ http://10tis.com

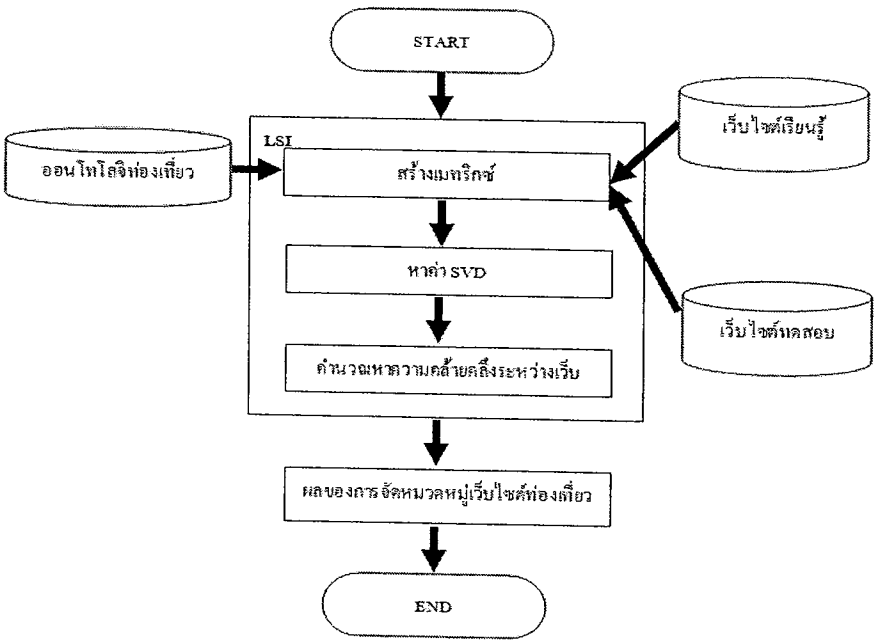


ภาพ 18 แสดงตัวอย่างเนื้อหาจากเว็บไซต์ http://10tis.com **ที่ถูกดึงมาจากการใช้** Jericho API

หลังจากการใช้อัลกอริทึม Web Crawler ในการเก็บข้อมูลท่องเที่ยวจากเว็บไซต์ต่างๆ แล้ว ผู้วิจัยได้นำเว็บไซต์เหล่านั้นมาทำการจัดหมวดหมู่เว็บไซต์ท่องเที่ยวด้วยอัลกอริทึม LSI อัลกอริทึมนาอีฟเบย์ และอัลกอริทึมนาอีฟเบย์ที่ปรับปรุงโดยใช้ค่าขอบเขต ดังรายละเอียดที่จะได้กล่าวต่อไป

3.3 การจัดหมวดหมู่เว็บไซต์ท่องเที่ยวด้วยอัลกอริทึม LSI

โดยงานวิจัยนี้ผู้วิจัยได้นำแนวคิด LSI (Thorleuchter and Poel, 2013) มาประยุกต์ใช้ในการวิเคราะห์เว็บไซต์เพื่อหาความคล้ายคลึงในการจัดหมวดหมู่เว็บไซต์ท่องเที่ยวประเทศไทย ซึ่งมีหลักการทำงาน ดังภาพ 19 เริ่มจากการสร้างเมทริกซ์ของคำที่ใช้ในการเรียนรู้และการทดสอบ จากนั้นแยกเมทริกซ์ของคำด้วยวิธีการ Single Value Decomposition (SVD) แล้วนำไปคำนวณหาค่าความคล้ายคลึงระหว่างเอกสารเว็บในแต่ละหมวดหมู่และรายงานผลของการจัดหมวดหมู่เว็บไซต์ท่องเที่ยว ดังรายละเอียดดังต่อไปนี้



ภาพ 19 แสดงการทำงานของการจัดหมวดหมู่เว็บไซต์ท่องเที่ยวด้วยอัลกอริทึม LSI

จากภาพ 19 แสดงการทำงานของอัลกอริทึม LSI โดยเริ่มจากการสร้างเมทริกซ์ของคำที่ใช้ในการเรียนรู้และการทดสอบ จากนั้นแยกเมทริกซ์ของคำด้วยวิธีการ SVD แล้วนำไปคำนวณหาค่าความคล้ายคลึงในแต่ละหมวดหมู่และรายงานผลของการจัดหมวดหมู่ ซึ่งแบ่งการทำงานเป็น 4 ขั้นตอนดังต่อไปนี้

1. **ข้อมูลสำหรับการเรียนรู้**

ในขั้นตอนนี้ผู้วิจัยได้แบ่งการทำงานเป็น 3 ขั้นตอนดังรายละเอียดต่อไปนี้

1.1 สร้างเมทริกซ์ของคำ

ในงานวิจัยนี้ได้ใช้ข้อมูลท่องเที่ยวจากเว็บไดเร็กทอรี่ truehits จำนวน 1,048 เว็บเป็นชุดข้อมูลการเรียนรู้แบ่งเป็น 6 หมวดหมู่ คือ สถานที่ท่องเที่ยว 233 เว็บไซต์ ที่พัก 200 เว็บไซต์ ร้านอาหาร 318 เว็บไซต์ ร้านขายของฝาก 54 เว็บไซต์ ร้านสินค้าหนึ่งตำบลหนึ่งผลิตภัณฑ์ 88 เว็บไซต์และเทศกาล 155 เว็บไซต์ มาตัด HTML Tags แล้วนำมาสร้างเมทริกซ์ของคำ ประกอบด้วยคำที่ใช้ในการค้นหาและความถี่ของคำที่ปรากฏในแต่ละเอกสาร ซึ่งคำเหล่านั้นได้นำมาจากออนโทโลจีท่องเที่ยวของนฤพนธ์ พนาวงศ์ และจักรกฤษณ์ เสน่ห์ (2553ก, 2553ข) โดยจะต้องสร้าง เมทริกซ์ของคำทุกหมวดหมู่ ในที่นี้ผู้วิจัยขอยกตัวอย่างเมทริกซ์ของคำที่ใช้ในการเรียนรู้เฉพาะหมวดสถานที่ท่องเที่ยวเท่านั้น ดังแสดงในภาพ 20

ตาราง 11 แสดงตัวอย่างค่าน้ำหนักของคำ

Word	เที่ยว	น้ำตก	เกาะ	ทะเล	วัด	ภูเขา	ตลาด
Weight	1.00	0.70	0.92	0.86	0.92	0.78	0.70

1.3 ปรับค่าน้ำหนัก

จากนั้นนำเมทริกซ์ของคำตามภาพ 20 ไปคูณกับค่าน้ำหนัก (ตาราง 11) ดังแสดงตัวอย่างการคำนวณและผลลัพธ์ของเมทริกซ์การเรียนรู้หมวดสถานที่ท่องเที่ยวในภาพ 21 ซึ่งจะได้เมทริกซ์ของคำของข้อมูลสำหรับการเรียนรู้ โดยจะนำไปใช้เปรียบเทียบกับเมทริกซ์ของคำที่ใช้ในการทดสอบ โดยวิธีการ SVD ในขั้นตอนที่ 3 และ 4 ตามลำดับ

M1 =

Term (m)	Web_1	Web_2	Web_3	Web_4	...	Web_{233}
เที่ยว (m_1)	431x1	79x1	152x1	179x1	...	222x1
	=431	=79	=152	=179		=222
น้ำตก (m_2)	15x0.7	19x0.7	26x0.7	17x0.7	...	24x0.7
	=10.05	=13.30	=18.20	=11.90		=16.80
ทะเล (m_3)	10x0.86	12 x0.86	17x0.86	18x0.86	...	46 x0.86
	=8.60	=10.32	=14.62	=15.48		=39.56
วัด (m_4)	8 x0.92	11 x0.92	68 x0.92	43 x0.92	...	11x0.92
	=7.36	=10.12	=62.56	=39.56		=10.12
:	:	:	:	:	:	:
ตลาด (m_{58})	13x0.7	7 x0.7	19 x0.7	24 x0.7	....	8 x0.7
	=9.10	=4.90	=13.30	=16.80		=5.60

ภาพ 21 แสดงตัวอย่างเมทริกซ์ของคำหมวดสถานที่ท่องเที่ยวหลังคูณค่าน้ำหนัก

2. ข้อมูลสำหรับการทดสอบ

ในการทดสอบแต่ละเว็บไซต์ต้องสร้างเมทริกซ์ของคำเช่นเดียวกับการเรียนรู้ โดยใช้ความถี่ของคำที่ปรากฏในเว็บไซต์ที่ใช้ในการทดสอบ ซึ่งคำว่า เที่ยว น้ำตก ทะเล วัด และ ตลาด ปรากฏอยู่ในเว็บไซต์ travel.edtguide.com เป็นจำนวน 140, 3, 5, 18 และ 8 คำ ตามลำดับ ดังแสดงตัวอย่างในภาพ 22 โดยงานวิจัยนี้ใช้ข้อมูลเว็บไซต์ท่องเที่ยวประเทศไทยจากผลของการ

	Term (m)	Web_1	Web_2	Web_3	Web_4	...	Web_{233}
	เที่ยว (m_1)	431	79	152	179	...	222
	น้ำตก (m_2)	15	19	26	17	...	24
	ทะเล (m_3)	10	12	17	18	...	46
M1 =	วัด (m_4)	8	11	68	43	...	11
	:	:	:	:	:	:	:
	ตลาด (m_{58})	13	7	19	24	....	8

ภาพ 20 แสดงตัวอย่างเมทริกซ์ของคำในหมวดสถานที่ท่องเที่ยว

จากภาพ 20 แสดงตัวอย่างเมทริกซ์ของคำที่สร้างมาจากความถี่ของคำที่ปรากฏในแต่ละเว็บไซต์ที่ใช้ในการเรียนรู้หมวดสถานที่ท่องเที่ยว ซึ่งจะเห็นว่าคำค้นหา "เที่ยว" (m_1) ปรากฏในเอกสาร Web_1 จำนวน 431 คำ Web_2 จำนวน 79 คำ Web_3 จำนวน 152 Web_4 จำนวน 179 คำ และ Web_{233} จำนวน 222 คำ ตามลำดับ ซึ่งจะเห็นได้ว่าคำว่า "เที่ยว" จะปรากฏทุกเว็บไซต์ของหมวดสถานที่ท่องเที่ยวจำนวน 233 เว็บไซต์ และคำค้นหา "น้ำตก" (m_2) ปรากฏในเอกสาร Web_1 จำนวน 15 คำ Web_2 จำนวน 19 คำ Web_3 จำนวน 26 คำ Web_4 จำนวน 17 คำ และ Web_{233} จำนวน 24 คำ เป็นต้น

1.2 หาค่าน้ำหนักของคำ

ในขั้นตอนนี้เป็นการปรับค่าน้ำหนักของคำที่ปรากฏในเอกสาร ซึ่งงานวิจัยนี้ผู้วิจัยใช้วิธีวิเคราะห์เนื้อหาเว็บไซต์ที่เป็นข้อความในส่วนของหัวเว็บเพจ ตัวเว็บเพจ โดยไม่พิจารณาลิงค์ที่เกี่ยวข้องทั้งนี้เนื่องจากต้องการความรวดเร็วในการประมวลผล (นฤพนธ์ พนาวงศ์ และจักรกฤษณ์ เสน่ห์ นมะหุต, 2557) จากนั้นนำมาหาความถี่ของคำที่ปรากฏในแต่ละเว็บไซต์ (โปรวิชั่น, 2552) จำนวน 50 เว็บไซต์ มาหาค่าน้ำหนักตามสมการ 9 และหากมีเว็บไซต์เพิ่มขึ้นค่าน้ำหนักของคำจะถูกปรับใหม่อย่างอัตโนมัติ ดังแสดงตัวอย่างในตาราง 11

$$\text{ค่าน้ำหนักของคำ} = \frac{\text{จำนวนเว็บไซต์ที่คำนั้นปรากฏในเว็บไซต์กลุ่มตัวอย่าง}}{\text{จำนวนเว็บไซต์ของกลุ่มตัวอย่าง}} \qquad (9)$$

สืบค้นในเว็บไซต์กูเกิ้ลที่ได้จากระบบจัดหมวดหมู่เว็บไซต์ท่องเที่ยวประเทศไทยด้วยอัลกอริทึมนาอีฟเบย์ จำนวน 475 เว็บไซต์ (นฤพนธ์ พนาวงศ์ และจักรกฤษณ์ เสน่ห์ นมะหุต, 2556ค)

	Term (m)	
	เที่ยว (m_1)	140
	น้ำตก (m_2)	3
	ทะเล (m_3)	5
M2 =	วัด (m_4)	18
	:	:
	ตลาด (m_{58})	8

ภาพ 22 แสดงตัวอย่างเมทริกซ์ของคำที่ใช้ในการทดสอบ (http://travel.edtguide.com)

ในการสร้างเมทริกซ์ของคำสำหรับข้อมูลการทดสอบนั้นให้ทำทีละเว็บไซต์จนครบจำนวนกลุ่มตัวอย่างที่นำมาทดสอบ 475 เว็บไซต์เฉพาะขั้นตอนที่ 1.1 เท่านั้นเนื่องจากมีเว็บไซต์เดียว จากนั้นค่อยนำไปเปรียบเทียบกับเมทริกซ์ของคำสำหรับข้อมูลการเรียนรู้ในขั้นตอนที่ 3 และ 4 ตามลำดับ

3. **แยกเมทริกซ์ของคำ-เอกสาร** (Term-Document Matrix)

ในขั้นตอนนี้เป็นการหาค่า SVD โดยนำเมทริกซ์ของคำที่ได้จากขั้นตอนที่ 1.3 ของการเรียนรู้ (M1) มาหาค่า SVD ด้วยสมการ 10 และเมทริกซ์ของคำสำหรับข้อมูลการทดสอบ (M2) มาหาค่า SVD ด้วยสมการ 11 (Eckart and Young, 1936) ดังต่อไปนี้

$$SVD(M1) = U\sum V^T \qquad (10)$$

เมื่อ U คือ เมทริกซ์เชิงตั้งฉากแทนความสัมพันธ์ของคำ-คำ

$\sum$ คือ เมทริกซ์ทแยงมุมแทนความสัมพันธ์ของคำ-เอกสาร

V^T คือ เมทริกซ์เชิงตั้งฉากแทนความสัมพันธ์ของเอกสาร-เอกสารและดำเนินการสับเปลี่ยนตำแหน่งตามหลักการ Transpose

$$SVD(M2) = M2^{T}U\sum^{-1} \quad (11)$$

เมื่อ $M2^{T}$ คือ เมทริกซ์ข้อมูลทดสอบแทนที่แสดงความสัมพันธ์ของคำ-เอกสารดำเนินการสับเปลี่ยนตำแหน่งตามหลักการ Transpose

U คือ เมทริกซ์เชิงตั้งฉากแทนความสัมพันธ์ของคำ-คำของข้อมูลการเรียนรู้

$\sum$ คือ เมทริกซ์ทแยงมุมแทนความสัมพันธ์ของคำ-เอกสารของข้อมูลการเรียนรู้

จากสมการ 10 และสมการ 11 สามารถคำนวณหาค่า SVD ของเมทริกซ์การเรียนรู้ (M1) และเมทริกซ์ที่ใช้ทดสอบ (M2) ในหมวดสถานที่ท่องเที่ยว ดังนี้

SVD(M1) = {0.2408,0.0250,0.0095,-0.0141,0.0126,-0.1058,-0.0467}

SVD(M2) = {0.7276,-0.2502,0.1717,-0.1339,0.0786,-0.1201,0.0576}

4. คำนวณหาค่าความคล้ายคลึง (Similarity Value)

ในขั้นตอนนี้จะนำค่า SVD ที่คำนวณได้ของเมทริกซ์ข้อมูลสำหรับการเรียนรู้และเมทริกซ์ข้อมูลสำหรับการทดสอบจากขั้นตอนที่ 3 มาหาค่าความคล้ายคลึงโดยใช้สูตรโคไซน์ (Salton and McGill, 1983) ดังสมการ 12 ตัวอย่างเช่น เมื่อนำค่าที่ได้จากขั้นตอนที่ 3 ของเว็บไซต์ http://travel.edtguide.com มาคำนวณหาค่าความคล้ายคลึงด้วยสมการ 12 จะได้ค่าความคล้ายคลึงในหมวดสถานที่ท่องเที่ยว (Att) ที่พัก (Acc) ร้านอาหาร (Res) ร้านขายของฝาก (Sou) ร้านสินค้าหนึ่งตำบลหนึ่งผลิตภัณฑ์ (OTOP) และเทศกาล (Even) เท่ากับ 0.90, 0.24, 0.89, -0.13, -0.21 และ 0.24 ตามลำดับ ซึ่งจะเห็นได้ว่าหมวดสถานที่ท่องเที่ยว (Att) มีค่าความคล้ายคลึงมากที่สุดเท่ากับ 0.90 ดังนั้นสรุปได้ว่าเว็บไซต์นี้อยู่ในหมวดสถานที่ท่องเที่ยว (Att)

$$Sim_{M1,M2} = \frac{\sum_{i=1}^{n} SVD(M1)*SVD(M2)}{\sqrt{\sum_{i=1}^{n} SVD(M1)^{2}}\sqrt{\sum_{i=1}^{n} SVD(M2)^{2}}} \quad (12)$$

เมื่อ $Sim_{M1,M2}$ คือ ค่าความคล้ายคลึงระหว่างเอกสารในแต่ละหมวดหมู่

M1 คือ เวกเตอร์ของเอกสารการเรียนรู้ในแต่ละหมวดหมู่

M2 คือ เวกเตอร์ของเอกสารทดสอบในแต่ละหมวดหมู่

n คือ จำนวนสมาชิกทั้งหมดของเวกเตอร์ M1 และ M2

จากนั้นหาค่าความคล้ายคลึงจนครบทั้งหมด 475 เว็บไซต์ ซึ่งผลลัพธ์ในการจัดหมวดหมู่เว็บไซต์ท่องเที่ยวด้วยอัลกอริทึม LSI จะแสดงในบทที่ 4

3.4 การจัดหมวดหมู่เว็บไซต์ท่องเที่ยวด้วยอัลกอริทึมนาอีฟเบย์

งานวิจัยนี้ได้ประยุกต์ใช้อัลกอริทึมนาอีฟเบย์ในการจัดหมวดหมู่เว็บไซต์ท่องเที่ยว ซึ่งใช้ข้อมูลการเรียนรู้ดังที่กล่าวไว้แล้วในหัวข้อการจัดหมวดหมู่ด้วยอัลกอริทึม LSI และใช้ผลการสืบค้นในเว็บไซต์กูเกิ้ลที่ได้จากระบบจัดหมวดหมู่เว็บไซต์ท่องเที่ยวประเทศไทยด้วยอัลกอริทึมนาอีฟเบย์ จำนวน 475 เว็บไซต์ (นฤพนธ์ พนาวงศ์ และจักรกฤษณ์ เสน่ห์ นมะหุต, 2556ค) เป็นชุดข้อมูลทดสอบ

แต่เนื่องจากผลของการคำนวณจากสมการที่ 7 ได้ค่าติดลบในทุกหมวดและจากความถี่ของคำค้นหาไปในทางกลุ่มที่มีผลการคำนวณติดลบมากที่สุด ดังนั้นผู้วิจัยจึงเลือกใช้ค่าน้อยที่สุดมาใช้ในการพิจารณาเว็บไซต์นั้นควรจัดอยู่ในหมวดหมู่ใด ดังสมการ 13

$$Cmap = \underset{c \in C}{\operatorname{argmin}} \left(P(c) \sum_{k=1}^{n} \log(P(t_k|c)) \right) \quad (13)$$

โดยมีรายละเอียดในการประยุกต์ใช้อัลกอริทึมนาอีฟเบย์สำหรับจัดหมวดหมู่เว็บไซต์ท่องเที่ยว ดังต่อไปนี้

1. นำข้อมูลเว็บไซต์ที่ใช้สำหรับการทดสอบมาจัดหมวดหมู่เว็บไซต์ท่องเที่ยวประเทศไทยด้วยอัลกอริทึมนาอีฟเบย์ (สมการ 13) จำนวน 6 หมวดหมู่ คือ สถานที่ท่องเที่ยว ที่พัก ร้านอาหาร ร้านขายของฝาก ร้านหนึ่งตำบลหนึ่งผลิตภัณฑ์และเทศกาล โดยใช้คำในออนโทโลจีท่องเที่ยวสำหรับจัดหมวดหมู่เว็บไซต์ท่องเที่ยวประเทศไทยของ นฤพนธ์ พนาวงศ์ และจักรกฤษณ์ เสน่ห์ (2553ก; 2553ข)

2. นำค่าผลคูณความน่าจะเป็นระหว่าง P(c) กับ $P(t_k|c)$ (P(NB)) ของแต่ละหมวดหมู่จากข้อที่ 1 มาเปรียบเทียบกันและเลือกค่าความน่าจะเป็นน้อยที่สุดมาใช้ในการจัดหมวดหมู่เว็บไซต์

3. ทำซ้ำในข้อที่ 1 จนกว่าจะครบทั้ง 475 เว็บไซต์

4. แสดงการจัดหมวดหมู่ของแต่ละเว็บไซต์

3.5 การจัดหมวดหมู่เว็บไซต์ท่องเที่ยวด้วยอัลกอริทึมนาอีฟเบย์ที่ปรับปรุงโดยใช้ค่าขอบเขต (MNB with Boundary Values)

จากสมการที่ 13 จะเห็นได้ว่าอัลกอริทึมนาอีฟเบย์ให้ P(NB) เพียงค่าเดียวในการจัดหมวดหมู่เว็บไซต์ว่าควรถูกจัดอยู่ในหมวดหมู่ใดเพียงหมวดเดียวเท่านั้น ซึ่งผู้วิจัยพบปัญหา คือ บางเว็บไซต์นำเสนอเนื้อหาที่หลากหลาย ประกอบด้วยหลายส่วน เช่น มีทั้งสถานที่ท่องเที่ยว ที่พัก และร้านอาหาร รวมอยู่ในเว็บไซต์เดียว ทำให้เมื่อใช้อัลกอริทึมนาอีฟเบย์จะทำให้เกิดความผิดพลาดในการจัดหมวดหมู่เว็บไซต์ได้ ดังนั้นงานวิจัยนี้ได้ปรับปรุงอัลกอริทึมนาอีฟเบย์โดยนำค่าขอบเขตมาใช้ในการจัดหมวดหมู่เว็บไซต์ท่องเที่ยวเพื่อให้ในแต่ละเว็บไซต์ถูกจัดอยู่หลายหมวดหมู่ เนื่องจากเนื้อหาในแต่ละเว็บไซต์มีเนื้อหาหลากหลายไม่เฉพาะเจาะจงเรื่องท่องเที่ยวเท่านั้น ยังมีเนื้อหาอื่นที่เป็นประโยชน์ต่อการท่องเที่ยว เช่น โรงแรม ที่พัก ร้านอาหาร เทศกาลต่าง ๆ รวมอยู่ในเว็บไซต์เดียว ซึ่งมีขั้นตอนการทำงานดังภาพ 23

1. นำข้อมูลท่องเที่ยวที่ได้จากเว็บไดเร็กทอรี่ truehits จำนวน 1,048 เว็บ แบ่งเป็น 6 หมวดหมู่ คือ สถานที่ท่องเที่ยว ที่พัก ร้านอาหาร ร้านขายของฝาก ร้านหนึ่งตำบลหนึ่งผลิตภัณฑ์และเทศกาล มาตัด HTML Tags

2. นำข้อมูลในข้อที่ 1 มาคำนวณหาค่า P(NB) ของแต่ละหมวดหมู่ด้วยอัลกอริทึมนาอีฟเบย์ (สมการ 13) โดยใช้คำในออนโทโลจีท่องเที่ยวในการจัดหมวดหมู่เว็บไซต์ท่องเที่ยว (นฤพนธ์ และจักรกฤษณ์, 2553ก; 2553ข)

3. นำค่า P(NB) ที่ได้จากการทำงานในข้อที่ 2 ของทุกเว็บไซต์การเรียนรู้มาคำนวณหาค่า P(NB) ต่ำสุดและ P(NB) สูงสุดในแต่ละหมวดหมู่ โดยใช้สมการที่ 14 และ 15 จากนั้นกำหนดค่า P(NB) ต่ำสุดและ P(NB) สูงสุดในแต่ละหมวดให้เป็นค่าขอบเขต (Threshold) เพื่อใช้ในการจัดหมวดหมู่เว็บไซต์ โดยกำหนดให้ CiMax เป็นค่าสูงสุดของแต่ละหมวดหมู่และ CiMin เป็นค่าต่ำสุดของแต่ละหมวดหมู่ (i) ดังแสดงในตาราง 12

$$CiMax = \underset{1 \le i \le 6}{\operatorname{argmax}}(Cmap) \tag{14}$$

$$CiMax = \underset{1 \le i \le 6}{\operatorname{argmin}}(Cmap) \tag{15}$$

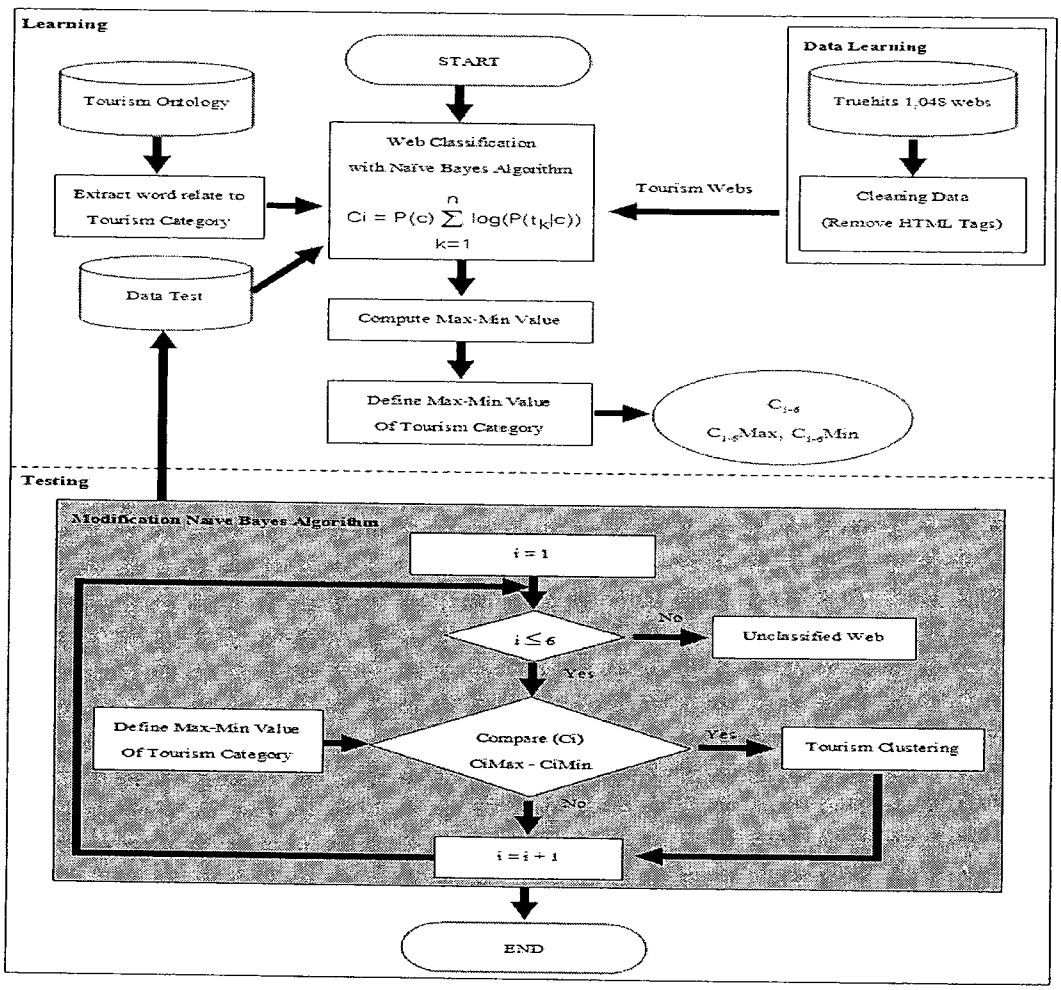
4. นำข้อมูลเว็บไซต์ที่ใช้สำหรับทดสอบมาคำนวณหาค่า P(NB) ของทุกหมวดหมู่

5. นำค่า P(NB) ที่ได้จากข้อที่ 4 ทุกหมวดหมู่ มาทำการเปรียบเทียบกับค่าขอบเขตในตาราง 12 โดยมีเงื่อนไขดังต่อไปนี้

5.1 ถ้าค่า P(NB) ที่ได้จากการคำนวณมีค่าอยู่ในช่วงของเขตเขตที่กำหนดไว้ในตาราง 12 แสดงว่าเว็บไซต์นี้สามารถจัดให้อยู่ในหมวดหมู่นั้นได้ แต่ถ้าค่า P(NB) ไม่อยู่ในช่วงของขอบเขตที่กำหนดไว้ก็ไม่สามารถจัดให้อยู่ในหมวดหมู่นั้นได้

5.2 ทำซ้ำในข้อที่ 5.1 จนครบทั้ง 475 เว็บไซต์

6. แสดงผลการจัดหมวดหมู่เว็บไซต์



ภาพ 23 แสดงการทำงานของอัลกอริทึมนาอีฟเบย์ที่ปรับปรุงโดยใช้ค่าขอบเขต

ตาราง 12 แสดงค่า P(NB) ขอบเขตต่ำสุดและสูงสุดของแต่ละหมวดหมู่ท่องเที่ยว

หมวดหมู่	ค่าขอบเขตต่ำสุด (Cmin)	ค่าขอบเขตสูงสุด (Cmax)
สถานที่ท่องเที่ยว (Att)	-20.06	-0.08
ที่พัก (Acc)	-4.35	-0.08
ร้านอาหาร (Res)	-14.97	-0.13
ร้านขายของที่ระลึก (Sou)	-0.47	-0.02
ร้านสินค้าหนึ่งตำบลหนึ่งผลิตภัณฑ์ (OTOP)	-1.17	-0.03
เทศกาล (Even)	-4.21	-0.05

จากตัวอย่างความถี่ของคำที่ถูกค้นพบในเว็บไซต์ travel.edtguide.com ในหมวดหมู่สถานที่ท่องเที่ยว (ตาราง 13) จากนั้นนำความถี่ของคำไปคำนวณตามสมการที่ 13 ของแต่ละหมวดหมู่ เช่น หมวดสถานที่ท่องเที่ยว (Att) โดยมีข้อมูลเว็บไซต์เรียนรู้ในหมวดสถานท่องเที่ยวเป็นจำนวน 233 เว็บไซต์จากข้อมูลเว็บไซต์เรียนรู้ทั้งหมดจำนวน 1,048 เว็บไซต์

= (233/1048) * (log(141/214) + log(4/214) + log(13/214) + log(4/214) + log(4/214) + log(7/214) + log(11/214) + log(6/214) + log(16/214) + log(4/214) + log(9/214) + log(3/214) + log(4/214))

= -4.12

โดยงานวิจัยนี้ผู้วิจัยได้ใช้การวิเคราะห์เว็บไซต์เพื่อจัดหมวดหมู่เว็บไซต์ท่องเที่ยว ใช้การวิเคราะห์เนื้อหาเว็บไซต์ ประกอบด้วย หัวเว็บเพจ ตัวเว็บ โดยไม่พิจารณาลิงค์ที่เกี่ยวข้อง ทั้งนี้เพื่อความรวดเร็วในการประมวลผล

ตาราง 13 แสดงความถี่ของคำที่พบในเว็บไซต์ travel.edtguide.com **(หมวดสถานที่ท่องเที่ยว)**

คำ	ความถี่ของคำที่พบในเว็บไซต์ travel.edtguide.com
เที่ยว	140
น้ำตก	3
พิพิธภัณฑ์	12
อุทยาน	3
สรรพสินค้า	3
สวนสัตว์	6
เกาะ	10
ทะเล	5
วัด	15
วิหาร	3
ตลาด	8
วัง	2
วัฒนธรรม	3
ผลรวมความถี่ของคำทั้งหมด	213

จากผลลัพธ์ที่ได้จากการจัดหมวดหมู่เว็บไซต์ท่องเที่ยวด้วยอัลกอริทึมนาอีฟเบย์ที่ปรับปรุงโดยใช้ค่าขอบเขต (บทที่ 4 ตารางที่ 25 - 27) ผู้วิจัยได้ทำการทดลองเพิ่มเติม ดังต่อไปนี้ 1) ใช้การคำนวณค่าของผลต่างระยะทางระหว่าง P(NB) กับค่าขอบเขตต่ำสุดของแต่ละหมวดหมู่ท่องเที่ยว 2) ผลต่างระยะทางระหว่าง P(NB) กับค่าจุดศูนย์กลางของแต่ละหมวดหมู่ท่องเที่ยว ดังรายละเอียดต่อไปนี้

3.6 การจัดหมวดหมู่เว็บไซต์ท่องเที่ยวด้วยการคำนวณค่าของผลต่างระยะทางระหว่าง P(NB) **กับค่าขอบเขตต่ำสุดของแต่ละหมวดหมู่ท่องเที่ยว** (MNB with Distance)

สำหรับการจัดหมวดหมู่เว็บไซต์ท่องเที่ยวนี้เป็นวิธีการคำนวณหาค่าของผลต่างระยะทางระหว่าง P(NB) กับค่าขอบเขตต่ำสุด (Cmin) ของแต่ละหมวดหมู่ท่องเที่ยวในตาราง 12 ซึ่งได้เรียกวิธีนี้ว่า MNB with Distance จากนั้นนำค่าของผลต่างระยะทางระหว่าง P(NB) กับค่า

ขอบเขตต่ำสุดของแต่ละหมวดหมู่ท่องเที่ยว (Diff(P(NB),Cmin)) มาใช้ในการจัดหมวดหมู่เว็บไซต์ท่องเที่ยว โดยผู้วิจัยนำเว็บไซต์เรียนรู้จำนวน 1,048 เว็บไซต์มาหาค่า Diff(P(NB),Cmin) และได้ทำการทดลองหาค่าขอบเขตต่ำสุดที่ดีที่สุดของแต่ละหมวดหมู่เพื่อใช้สำหรับเป็นค่าเขตแดนของระยะทางที่ดีที่สุด (Threshold) ซึ่งมีขั้นตอนดังต่อไปนี้

1. เริ่มจากการสุ่มค่า Diff(P(NB),Cmin) ที่น้อยที่สุดของแต่ละหมวดหมู่มาทดสอบการจัดหมวดหมู่เว็บไซต์ท่องเที่ยว

2. ทำการปรับค่า Diff(P(NB),Cmin) เพื่อหาค่าประสิทธิภาพของการจัดหมวดหมู่ที่ดีที่สุด

3. นำค่า Diff(P(NB,Cmin) ที่ได้ค่าประสิทธิภาพที่ดีที่สุดมาทำการตั้งค่าเขตแดนของระยะทางที่ดีที่สุดในแต่ละหมวดหมู่ (ตาราง 14)

ตาราง 14 แสดงค่าเขตแดนของระยะทางที่ดีที่สุดในแต่ละหมวดหมู่ท่องเที่ยว

หมวดหมู่	ค่าเขตแดนของระยะทางที่ดีที่สุด
สถานที่ท่องเที่ยว (Att)	-8.52
ที่พัก (Acc)	-1.25
ร้านอาหาร (Res)	-0.27
ร้านขายของที่ระลึก (Sou)	-0.40
ร้านสินค้าหนึ่งตำบลหนึ่งผลิตภัณฑ์ (OTOP)	-0.70
เทศกาล (Even)	-1.44

สำหรับการจัดหมวดหมู่ด้วยวิธีนี้ทำได้ด้วยการคำนวณหาค่า Diff(P(NB), Cmin) ในแต่ละหมวดหมู่ โดยหากค่า Diff(P(NB,Cmin) ที่คำนวณได้นั้นมีค่าน้อยกว่าหรือเท่ากับค่าเขตแดนของระยะทางที่ดีที่สุด (ตาราง 14) แสดงว่าเว็บไซต์นั้นสามารถจัดให้อยู่ในหมวดหมู่นั้นได้ แต่ถ้าค่า Diff(P(NB), Cmin) ที่คำนวณได้มีค่ามากกว่าค่าเขตแดนของระยะทางที่ดีที่สุดแสดงว่าเว็บไซต์นั้นไม่สามารถจัดให้อยู่ในหมวดหมู่นั้นได้ ตัวอย่างเช่น หมวดหมู่สถานที่ท่องเที่ยว (Att) ของเว็บไซต์ travel.edtguide.com มีค่า P(NB) เท่ากับ -4.12 เมื่อทำการคำนวณหาค่า Diff(-4.12,-20.06) จะได้เท่ากับ -15.94 จากนั้นนำค่า -15.94 มาเปรียบเทียบกับค่าเขตแดนของระยะทางที่ดีที่สุดในหมวดหมู่สถานที่ท่องเที่ยว (ตาราง 14) พบว่าค่า -15.94 มีค่าน้อยกว่าค่าเขตแดนของระยะทางที่ดีที่สุดในหมวดหมู่สถานที่ท่องเที่ยว (-8.52) ดังนั้นจึงสรุปได้ว่าเว็บไซต์นี้

สามารถถูกจัดให้อยู่ในหมวดหมู่สถานที่ท่องเที่ยวได้ สำหรับการจัดหมวดหมู่ที่พัก ร้านอาหาร ร้านขายของฝาก ร้านสินค้าหนึ่งตำบลหนึ่งผลิตภัณฑ์และเทศกาลก็ทำเช่นเดียวกัน ดังแสดงผลลัพธ์ในการจัดหมวดหมู่เว็บไซต์ท่องเที่ยวด้วยวิธี MNB with Distance ในบทที่ 4

3.7 **การจัดหมวดหมู่เว็บไซต์ท่องเที่ยวด้วยการคำนวณค่าของผลต่างระยะทางระหว่าง** P(NB) **กับค่าจุดศูนย์กลางของแต่ละหมวดหมู่ท่องเที่ยว** (MNB with Nearest Centroid)

สำหรับการจัดหมวดหมู่เว็บไซต์ท่องเที่ยวนี้เป็นวิธีการคำนวณหาค่าของผลต่างระยะทางระหว่าง P(NB) กับค่าจุดศูนย์กลาง (Centroid) ของแต่ละหมวดหมู่ท่องเที่ยว (Diff(P(NB),Centroid)) ซึ่งได้เรียกวิธีนี้ว่า MNB with Nearest Centroid โดยผู้วิจัยใช้ค่า P(NB) ของแต่ละหมวดหมู่จากเว็บไซต์เรียนรู้จำนวน 1,048 เว็บไซต์และทำการคัดเลือกเฉพาะค่า P(NB) ที่มีการกระจายแบบหนาแน่นมาใช้ในการหาค่าจุดศูนย์กลางที่ดีที่สุดของแต่ละหมวดหมู่ท่องเที่ยว ในงานวิจัยนี้จะเห็นได้ว่าค่า P(NB) นี้มีการกระจายข้อมูลแบบ Linear Regression (ดูเพิ่มเติมที่ภาคผนวก ก) ดังนั้นการหาค่าจุดศูนย์กลางของแต่ละหมวดหมู่ที่มีการกระจายแบบ Linear Regression สามารถทำได้ด้วยการคำนวณหาค่าเฉลี่ยของ P(NB) ในแต่ละหมวดหมู่ (Jordan, et al., 2006) หรือใช้อัลกอริทึม K-Mean (Khan and Mohamudally, 2010) โดยค่าจุดศูนย์กลางสามารถคำนวณได้จากสมการ 16 และผลลัพธ์ของค่าจุดศูนย์กลางในแต่ละหมวดหมู่ แสดงในตาราง 15

$$\text{จุดศูนย์กลาง} = \frac{\text{ผลรวมของค่าความน่าจะเป็นของแต่ละหมวดหมู่}}{\text{จำนวนเว็บไซต์ทั้งหมวดของแต่ละหมวดหมู่}} \qquad (16)$$

ตาราง 15 แสดงจุดศูนย์กลางของแต่ละหมวดหมู่ท่องเที่ยว

หมวดหมู่	จุดศูนย์กลาง
สถานที่ท่องเที่ยว (Att)	-1.77
ที่พัก (Acc)	-0.61
ร้านอาหาร (Res)	-5.46
ร้านขายของที่ระลึก (Sou)	-0.13
ร้านสินค้าหนึ่งตำบลหนึ่งผลิตภัณฑ์ (OTOP)	-0.19
เทศกาล (Even)	-0.35

หลังจากที่ได้ค่าจุดศูนย์กลางของแต่ละหมวดหมู่ท่องเที่ยวจากนั้นนำค่าจุดศูนย์กลางมาคำนวณหาค่าขอบเขตของรัศมีที่ดีที่สุดจากจุดศูนย์กลางของแต่ละหมวดหมู่ (r) สำหรับการหาค่ารัศมีที่ดีที่สุดของแต่ละหมวดหมู่นั้นทำได้โดย (ดูเพิ่มเติมที่ ภาคผนวก ก)

1. เริ่มจากการสุ่มค่ารัศมีที่น้อยที่สุดของแต่ละหมวดหมู่ โดยใช้ค่า Diff(P(NB), Centroid) มาทดสอบการจัดหมวดหมู่เว็บไซต์ท่องเที่ยว

2. ทำการปรับค่ารัศมีเพื่อหาค่าประสิทธิภาพของการจัดหมวดหมู่ที่ดีที่สุด

3. นำค่ารัศมีที่ได้ค่าประสิทธิภาพดีที่สุดมาทำการตั้งค่าเขตแดนของขอบเขตรัศมีที่ดีที่สุดในแต่ละหมวดหมู่ (ตาราง 16)

ตาราง 16 แสดงค่าขอบเขตของรัศมีที่ดีที่สุดในแต่ละหมวดหมู่ท่องเที่ยว

หมวดหมู่	ค่าขอบเขตของรัศมีที่ดีที่สุด
สถานที่ท่องเที่ยว (Att)	18.33
ที่พัก (Acc)	3.80
ร้านอาหาร (Res)	9.52
ร้านขายของที่ระลึก (Sou)	0.22
ร้านสินค้าหนึ่งตำบลหนึ่งผลิตภัณฑ์ (OTOP)	0.30
เทศกาล (Even)	2.69

สำหรับการจัดหมวดหมู่ด้วยวิธีนี้ทำได้ด้วยการคำนวณหาค่า Diff(P(NB), Centroid) ในแต่ละหมวดหมู่ โดยหากค่า Diff(P(NB, Centroid) ที่คำนวณได้นั้นมีค่าที่อยู่ในขอบเขตของรัศมีที่ดีที่สุด (ตาราง 16) แสดงว่าเว็บไซต์นั้นสามารถจัดให้อยู่ในหมวดหมู่นั้นได้ แต่ถ้าค่า Diff(P(NB),Centroid) ที่คำนวณได้มีค่าไม่อยู่ในขอบเขตของรัศมีที่ดีที่สุดแสดงว่าเว็บไซต์นั้นไม่สามารถจัดให้อยู่ในหมวดหมู่นั้นได้ ตัวอย่างเช่น หมวดหมู่สถานที่ท่องเที่ยว (Att) ของเว็บไซต์ travel.edtguide.com มีค่า P(NB) เท่ากับ -4.12 เมื่อทำการคำนวณหาค่า Diff(-4.12,-1.77) จะได้เท่ากับ 2.35 จากนั้นนำค่า 2.35 มาเปรียบเทียบว่าอยู่ในขอบเขตของรัศมีที่ดีที่สุดในหมวดหมู่สถานที่ท่องเที่ยวหรือไม่ (ตาราง 16) ซึ่งพบว่าค่าผลต่างนี้ (2.35) อยู่ในขอบเขตของรัศมีที่ดีที่สุดของหมวดหมู่สถานที่ท่องเที่ยว (มีค่าน้อยกว่า18.33) ดังนั้นจึงสรุปได้ว่าเว็บไซต์นี้สามารถถูกจัดให้อยู่ในหมวดหมู่สถานที่ท่องเที่ยวได้ สำหรับการจัดหมวดหมู่ที่พัก ร้านอาหาร ร้านขายของฝาก ร้าน

สินค้าหนึ่งตำบลหนึ่งผลิตภัณฑ์และเทศกาลก็ทำเช่นเดียวกัน ดังแสดงผลลัพธ์ในการจัดหมวดหมู่เว็บไซต์ท่องเที่ยวด้วยวิธี MNB with Nearest Centroid ในบทที่ 4

3.8 การคำนวณหาข้อมูลท่องเที่ยวให้ตรงตามความต้องการของผู้ใช้ด้วยอัลกอริทึม K-NN

เนื่องจากผู้วิจัยต้องการให้ได้ข้อมูลท่องเที่ยวที่ตรงกับความต้องการของผู้ใช้มากที่สุด ดังนั้นเมื่อผู้ใช้ต้องการข้อมูลท่องเที่ยวจะต้องป้อนและเลือกประเภทสถานที่ท่องเที่ยวที่ต้องการ ประเภทที่พัก ประเภทอาหารที่ชอบ สิ่งอำนวยความสะดวกที่ต้องการและกิจกรรมที่สนใจดังหน้าจอที่จะกล่าวในหัวข้อถัดไป จากนั้นระบบจะคำนวณหาความคล้ายคลึงของข้อมูลท่องเที่ยวกับข้อมูลท่องเที่ยวที่ผู้ใช้ต้องการด้วยอัลกอริทึม K-NN ซึ่ง ณ ที่นี้ขอยกตัวอย่างการคำนวณหาความคล้ายคลึงของการเลือกประเภทร้านอาหารที่ผู้ใช้ต้องการ ดังแสดงในตาราง 17 โดยตัวอย่างชุดข้อมูลประเภทร้านอาหารได้มาจากข้อมูลร้านอาหารที่ถูกเก็บไว้ในออนโทโลจีท่องเที่ยวตามที่ได้ออกแบบไว้แล้ว จากนั้นข้อมูลร้านอาหารจะถูกนำมาเปรียบเทียบเฉพาะจังหวัดที่ผู้ใช้ต้องการไปเท่านั้น เช่น ร้านส้มตำพันล้าน เป็นประเภท ยำ&ตำ ร้านก๋วยเตี๋ยวเรืออนุบาล เป็นประเภท ก๋วยเตี๋ยว KFC เป็นประเภท จานด่วน เป็นต้น และเพื่อให้เห็นถึงการทำงานของอัลกอริทึม K-NN ผู้วิจัยจึงได้ยกตัวอย่างความต้องการของคน 1 คนที่ชอบอาหารประเภทบาร์บีคิว เมื่อนำมาเขียนเป็นตารางความต้องการทานอาหารจะได้ดังตาราง 18

ตาราง 17 แสดงตัวอย่างชุดข้อมูลประเภทร้านอาหาร

ประเภทร้านอาหาร	บาร์บีคิว	ก๋วยเตี๋ยว	สเต็ก	จานด่วน	ชาบู	เบเกอรี่	ยำ&ตำ	ต้มซำ	สุกี้	ทะเล	ชื่อร้านอาหาร
ชุดที่ 1	0	0	0	0	0	0	1	0	0	0	ร้านส้มตำพันล้าน
ชุดที่ 2	0	1	0	0	0	0	0	0	0	0	ก๋วยเตี๋ยวเรืออนุบาล
ชุดที่ 3	0	0	0	1	0	0	0	0	0	0	KFC
ชุดที่ 4	0	0	0	0	0	0	0	0	1	1	ป.ปลาซีฟู้ด

ตาราง 18 แสดงความต้องการทานอาหาร

ประเภทร้านอาหาร	บาร์บีคิว	ก๋วยเตี๋ยว	แฮมเบอร์เกอร์	พิซซ่า	ยำ&ตำ	ต้มยำ	ผัด	ทะเล	ชื่อร้านอาหาร
	1	0	0	0	0	0	0	0	0	????

จากนั้นคำนวณหาระยะทางแฮมมิ่ง ของชุดข้อมูล 1 ถึง 4 ด้วยสมการ 8 ดังตาราง 19 จะเห็นได้ว่าจากการคำนวณหาระยะทางแฮมมิ่งของร้านอาหารทั้งหมด ระบบจะแนะนำร้านอาหารที่มีค่าระยะทางน้อยที่สุด คือ ร้านส้มตำพันล้าน ก๋วยเตี๋ยวเรืออนุบาล และ KFC (ใกล้เคียงที่สุด) สำหรับการแนะนำสถานที่ท่องเที่ยวและที่พัก ใช้หลักการคำนวณด้วยอัลกอริทึมนี้เช่นเดียวกัน

จากตาราง 19 จะเห็นได้ว่าไม่พบร้านอาหารประเภทบาร์บีคิวอยู่ในข้อมูลร้านอาหารทั้งหมด ทำให้ผลลัพธ์ที่ได้ไม่ถูกต้อง ดังนั้นเพื่อความถูกต้องในการแนะนำข้อมูลท่องเที่ยว ผู้วิจัยจึงได้ปรับปรับปรุงอัลกอริทึม K-NN ที่ใช้ระยะทางแฮมมิ่ง จากสมการที่ 8 เป็นสมการที่ 17 ดังรายละเอียดต่อไปนี้

$$D_H = \sum_{i=1}^{n} \left(x_i = y_i \right) \qquad (17)$$

โดย D_H คือ ผลรวมของชุดข้อมูลท่องเที่ยว x ที่เท่ากับชุดข้อมูลท่องเที่ยวที่ผู้ใช้เลือก y

ตาราง 19 แสดงการคำนวณหาระยะทางแฮมมิ่งของร้านอาหาร

	บาร์บีคิว	ก๋วยเตี๋ยว	สเต็ก	หมูหมัน	พิซซ่า	ยูเอมเกา	ต้มยำ	ต้มจืด	สุกี้	ตามสั่ง	ระยะทางแฮมมิ่ง
ร้านส้มตำพันล้าน	0	0	0	0	0	0	1	0	0	0	
ต้องการทานอาหาร	1	0	0	0	0	0	0	0	0	0	
	1	0	0	0	0	0	1	0	0	0	2
ก๋วยเตี๋ยวเรืออนุบาล	0	1	0	0	0	0	0	0	0	0	
ต้องการทานอาหาร	1	0	0	0	0	0	0	0	0	0	
	1	1	0	0	0	0	0	0	0	0	2
KFC	0	0	0	1	0	0	0	0	0	0	
ต้องการทานอาหาร	1	0	0	0	0	0	0	0	0	0	
	1	0	0	1	0	0	0	0	0	0	2
ป.ปลาซีฟู้ด	0	0	0	0	0	0	0	0	1	1	
ต้องการทานอาหาร	1	0	0	0	0	0	0	0	0	0	
	1	0	0	0	0	0	0	0	1	1	3

ตาราง 20 แสดงการคำนวณหาระยะทางแฮมมิ่งของร้านอาหารหลังปรับปรุง K-NN

	บาร์บีคิว	ก๋วยเตี๋ยว	สเต็ก	หมูหมัน	พิซซ่า	ยูเอมเกา	ต้มยำ	ต้มจืด	สุกี้	ตามสั่ง	ระยะทางแฮมมิ่ง
ร้านส้มตำพันล้าน	0	0	0	0	0	0	1	0	0	0	
ต้องการทานอาหาร	1	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	0	0

ตาราง 20 (ต่อ)

	บาร์บีคิว	ก๋วยเตี๋ยว	สเต็ก	หมูกระทะ	พิซซ่า	เบเกอรี่	ยำ&ตำ	ติ่มซำ	ปู้	ชาบู	ระยะทางแฮมมิ่ง
ก๋วยเตี๋ยวเรืออนุบาล	0	1	0	0	0	0	0	0	0	0	
ต้องการทานอาหาร	1	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	0	0
KFC	0	0	0	1	0	0	0	0	0	0	
ต้องการทานอาหาร	1	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	0	0
ป.ปลาซีฟู้ด	0	0	0	0	0	0	0	0	1	1	
ต้องการทานอาหาร	1	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	0	0

จากตาราง 20 จะเห็นได้ว่าจากการคำนวณหาระยะทางแฮมมิ่งของร้านอาหารทั้งหมด มีค่าเป็นศูนย์ ทั้งนี้เนื่องจากไม่พบร้านอาหารประเภทบาร์บีคิวในระบบ ซึ่งระบบจะแนะนำร้านอาหารทั้งหมด แต่แสดงข้อความเป็น "ใกล้เคียงน้อยที่สุด"

3.9 การแนะนำข้อมูลท่องเที่ยวตามช่วงเวลาและบริเวณใกล้เคียง

จากการออกแบบและพัฒนาออนโทโลจีเชิงเวลาดังที่ได้กล่าวมาแล้วนั้น ทำให้ผู้ใช้ไม่พลาดเทศกาลหรือประเพณีที่สำคัญๆ ตามช่วงเวลาที่มีในจังหวัดนั้นๆ อีกทั้งยังทำให้การท่องเที่ยวได้อย่างมีความสุข เหมาะสมกับฤดูกาลท่องเที่ยว ดังแสดงการทำงานของอัลกอริทึมเพื่อหาวันสำคัญต่างๆ ตามปฏิทินจันทรคติ (ภาพ 24) และการเรียกใช้ฟังก์ชั่นด้วยภาษา JSP ในการคำนวณหาฤดูกาลและปฏิทินจันทรคติ (ภาพ 25) ดังรายละเอียดต่อไปนี้

1. รับข้อมูลวันที่จากผู้ใช้ที่จะไปท่องเที่ยว
2. คำนวณหาจำนวนวันจากวันออกพรรษาของปีจากวันที่ในข้อที่ 1 ดังนี้ จากสมการ 1 (หัวข้อปฏิทินสุริยคติและปฏิทินจันทรคติ)
3. คำนวณหาวันทางจันทรคติ ตามเงื่อนไขดังต่อไปนี้
   3.1 ถ้าจำนวนวันที่คำนวณได้จากข้อที่ 1 เท่ากับ 354 วันเรียกกว่า ปกติมาส
   3.2 ถ้าจำนวนวันที่คำนวณได้จากข้อที่ 1 เท่ากับ 355 วันเรียกกว่า ปีอธิกวาร (เดือน 7 จะมี 30 วัน)
   3.3 ถ้าจำนวนวันที่คำนวณได้จากข้อที่ 1 เท่ากับ 384 วันเรียกกว่า ปีอธิกมาส (เดือน 8 มี 2 ครั้ง)
4. หาวันที่ออกพรรษาของปีจากวันที่ในข้อที่ 1 แล้วเปรียบเทียบเงื่อนไขดังต่อไปนี้
   4.1 ถ้าวันที่จากข้อที่ 1 เท่ากับวันออกพรรษา (ขึ้น 15 ค่ำ เดือน 11) ข้ามไปข้อที่ 5
   4.2 ถ้าวันที่จากข้อที่ 1 น้อยกว่าวันออกพรรษา ลดจำนวนวันลงทีละ 1 และลดเดือนลงทีละ 1 กรณีที่จำนวนวันของเดือนนั้นๆ ครบ
   4.3 ถ้าวันที่จากข้อที่ 1 มากกว่าวันออกพรรษา เพิ่มจำนวนวันทีละ 1 และเพิ่มเดือนทีละ 1 กรณีที่จำนวนวันของเดือนนั้นๆ ครบ
   4.4 ทำซ้ำข้อที่ 4.2 หรือ 4.3 ไปจนกว่าวันที่จะเท่ากับวันที่ในข้อที่ 1
5. แสดงวันที่ตามปฏิทินจันทรคติ

ภาพ 24 แสดงอัลกอริทึมหาวันสำคัญตามปฏิทินจันทรคติ

สำหรับอัลกอริทึมนี้สามารถใช้คำนวณวันสำคัญตามเทศกาลท่องเที่ยวหรือประเพณีที่มีในประเทศไทยใน 1 ปี ดังตัวอย่างการคำนวณหาปฏิทินจันทรคติ

1. ผู้ใช้ป้อนวันที่ 15 ต.ค. 2557 สำหรับเดินทางท่องเที่ยว
2. ตรวจสอบปี โดยใช้สมการที่ 1 จะได้ว่าปี 2557 นี้เป็นปีปกติมาส
3. นำปีไปทำการหาวันออกพรรษาของปี 2557 ซึ่งจะตรงกับวันที่ 8 ต.ค.
4. เพิ่มจำนวนวันจนถึงวันที่ผู้ใช้ป้อนเข้ามา และแสดงผลลัพธ์ แรม 7 ค่ำ เดือน 11

จากนั้นนำวันทางจันทรคติ แรม 7 ค่ำ เดือน 11 ไปทำการตรวจสอบกับออนโทโลจีท่องเที่ยว แล้วแสดงผลการแนะนำข้อมูลท่องเที่ยวตามช่วงเวลาที่มาตรงกับเทศกาลท่องเที่ยวหรือประเพณีอะไรบ้าง อีกทั้งยังใช้วันที่ท่องเที่ยวที่ผู้ใช้ป้อนไปคำนวณหาฤดูกาลท่องเที่ยวอีกด้วย ดังฟังก์ชั่นการทำงานในภาพ 25

```
String thaiprovince = "พิษณุโลก",String dateInput = "05/11/2557";
// ฟังก์ชั่นหาฤดูกาล ส่งชื่อจังหวัดและวันที่
String season = Function.findSeason(thaiprovince,dateInput);
// ฟังก์ชั่นหาวันพระและวันสำคัญตามปฏิทินจันทรคติ ส่งวัน เดือน ปี เป็นตัวเลข
Thailunar thailunar = new Thailunar(day,mon,year);
datelunar = thailunar.getThailunar();
System.out.println(dateInput + " (" + datelunar + ") " + thailunar.getBuddhaDay() +
((thailunar.isBuddha())?" <b>วันนี้เป็นวันพระ</b> แนะนำให้ไปทำบุญตามวัดต่างๆ":"") + "<br>");
```

ภาพ 25 แสดงตัวอย่างโปรแกรมคำนวณฤดูกาลและวันสำคัญตามปฏิทินจันทรคติ

แต่อย่างไรก็ตามเมื่อนักท่องเที่ยวไปเที่ยวชมตามสถานที่ต่างๆ มักจะมีข้อสงสัยเกี่ยวกับข้อมูลท่องเที่ยวใกล้เคียง กล่าวคือ มีสถานที่ท่องเที่ยวใดบ้าง ที่พัก ร้านอาหาร ร้านขายของฝาก และร้านสินค้าหนึ่งตำบลหนึ่งผลิตภัณฑ์ใดบ้างที่อยู่บริเวณใกล้เคียงกับสถานที่ท่องเที่ยวนั้นๆ โดยงานวิจัยนี้ผู้วิจัยใช้พิกัดทางภูมิศาสตร์ (ละติจูด, ลองติจูด) ที่ได้เก็บไว้แล้วในออนโทโลจีท่องเที่ยว จากนั้นมาคำนวณหาระยะทางแล้วแสดงผลการแนะนำข้อมูลท่องเที่ยวบริเวณใกล้เคียงเรียงตามระยะทางจากน้อยไปหามาก (จากสถานที่ท่องเที่ยวไปยังสถานที่ใกล้เคียง) ในที่นี้ผู้วิจัยขอยกตัวอย่างโปรแกรมการคำนวณระยะทางจากวัดพระศรีรัตนมหาธาตุวรมหาวิหารไปยังวัดนางพญา จังหวัดพิษณุโลก โดยใช้ฟังก์ชั่น calDistance ดังภาพ 26

```
// ตำแหน่งพิกัดทางภูมิศาสตร์วัดพระศรีรัตนมหาธาตุวรมหาวิหาร
double la1 = 16.823661, long1 = 100.262242;

// ตำแหน่งพิกัดทางภูมิศาสตร์วัดนางพญา
double la2 = 16.822333, long2 = 100.262038;
distance = Function.calDistance(la1, long1, la2, long2, 'K');
out.print("ระยะทาง = " + distance + " กิโลเมตร"); // 0.15 กิโลเมตร
```

ภาพ 26 แสดงตัวอย่างโปรแกรมคำนวณระยะทางโดยใช้พิกัดทางภูมิศาสตร์

4. การออกแบบและพัฒนาระบบแนะนำข้อมูลท่องเที่ยวในประเทศไทย

จากแนวคิดของ Siricharoen (2008) ที่อธิบายเรื่อง e-Tourism ว่าจะต้องประกอบไปด้วย เที่ยวที่ไหน พักที่ไหน ไปเมื่อไหร่ ทำกิจกรรมอะไร เดินทางอย่างไร รวมถึง บุญเลิศ จิตตั้งวัฒนา (2555) ที่อธิบายถึงปัจจัยที่ใช้ในการตัดสินใจท่องเที่ยวของนักท่องเที่ยว คือ จะไปท่องเที่ยวที่ไหน ไปเมื่อไหร่ ทางใด ไปกับใครบ้าง ใช้เวลาเท่าไหร่ เที่ยวชมอะไร สถานที่ประเภทไหน ไปกี่วัน มาประยุกต์ใช้ในการออกแบบหน้าจอ ซึ่งแบ่งเป็น 2 ส่วนคือ จากการสืบค้นโดยใช้คำค้นหาหรือประโยค (ภาพ 27) และเลือกข้อมูลท่องเที่ยวตามความต้องการของผู้ใช้ (ภาพ 28) ซึ่งจะถูกป้อนโดยผู้ใช้คนใดคนหนึ่งในกลุ่มของสมาชิกที่ไปท่องเที่ยวด้วยกันหรือจะป้อนครบทุกคนก็ได้และผู้ป้อนสามารถเลือกระดับความสำคัญในการไปท่องเที่ยวตามต้องการ เพื่อแนะนำข้อมูลท่องเที่ยวตามความต้องการของผู้ใช้ ซึ่งมีตัวเลือกแบ่งเป็นข้อย่อยต่าง ๆ ดังนี้

1. ประเภทที่ท่องเที่ยว แบ่งเป็น ภูเขา น้ำตก อ่างเก็บน้ำ พิพิธภัณฑ์ ถ้ำ สวนสาธารณะ ละครสัตว์ ห้างสรรพสินค้า เขื่อน ภาพยนตร์ อ่างเก็บน้ำ สถานบันเทิง สวนสัตว์ ปราสาท โบราณสถาน เกาะ สนามกีฬา ทะเล ชุมชน ราชการ วัด ตลาด วัง ศูนย์วัฒนธรรม อนุสาวรีย์ สำนักสงฆ์ (นฤพนธ์ พนาวงศ์ และจักรกฤษณ์ เสน่ห์, 2553ก, 2553ข)

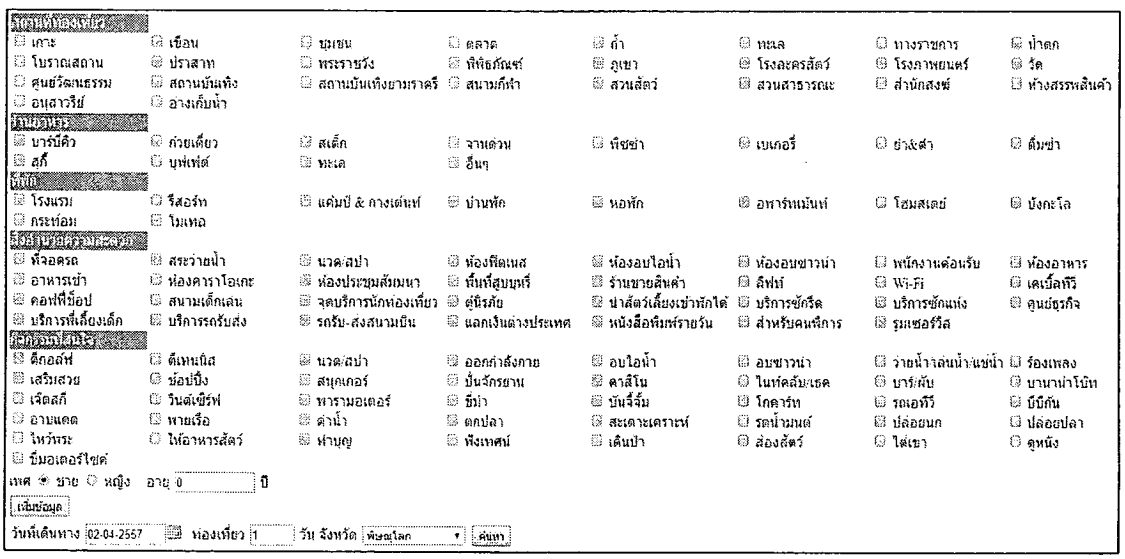
2. อาหารที่ชอบ แบ่งเป็น บาร์บีคิว ก๋วยเตี๋ยว สเต็ก จานด่วน พิซซ่า เบเกอรี่ ยำ&ตำ สุกี้ บุฟเฟต์ ทะเลและอื่นๆ (นฤพนธ์ พนาวงศ์ และจักรกฤษณ์ เสน่ห์ นมะหุต, 2555)

3. ประเภทที่พัก แบ่งเป็น โรงแรม รีสอร์ท แค้มป์ บ้านพัก หอพัก อพาร์ทเม้นต์ โฮมสเตย์ บังกะโล กระท่อม โมเทล (นฤพนธ์ พนาวงศ์ และจักรกฤษณ์ เสน่ห์, 2553ก, 2553ข) สิ่งอำนวยความสะดวกต่างๆ คือ ที่จอดรถ สระว่ายน้ำ นวดหรือสปา ห้องฟิตเนต ห้องอบไอน้ำ ห้องอบซาวน่า ห้องอาหาร ห้องประชุมสัมมนา พนักงานต้อนรับ ห้องคาราโอเกะ อาหารเช้า พื้นที่สูบบุหรี่ ร้านขายสินค้า ลิฟต์ อินเตอร์เน็ตไร้สาย เคเบิ้ลทีวี คอฟฟี่ช้อป สนามเด็กเล่น จุดบริการนักท่องเที่ยว ตู้นิรภัย นำสัตว์เลี้ยงเข้าพักได้ บริการซักรีด ศูนย์ธุรกิจ บริการพี่เลี้ยงเด็ก บริการรถรับส่ง รถรับ-ส่งสนามบิน แลกเงินตราต่างประเทศ หนังสือพิมพ์รายวัน มีบริการสำหรับคนพิการ รูมเซอร์วิสและกิจกรรมที่สนใจ คือ ตีกอล์ฟ ตีเทนนิส นวดหรือสปา ออกกำลังกาย อบไอน้ำ อบซาวน่า ว่ายน้ำ ร้องเพลง เสริมสวย ช้อปปิ้ง สนุกเกอร์ ปั่นจักรยาน คาสิโน ไนต์คลับ บาร์หรือผับ บานาน่าโบ๊ท เจ๊ตสกี วินด์เซิร์ฟ พารามอเตอร์ ขี่ม้า บันจี้จั้ม โกคาร์ท รถเอทีวี บีบีกัน ไหว้พระ ให้อาหารสัตว์ ทำบุญ ฟังเทศน์ เดินป่า ส่องสัตว์ ไต่เขา ดูหนังและขี่มอเตอร์ไซค์ (รุจิรา ธรรมสมบัติ และทองพูล หีบไธสง, 2552; Chaves, et al., 2012)

4. เพศ แบ่งเป็น ชาย และ หญิง
5. อายุ
6. วันที่เดินทาง
7. จำนวนวันท่องเที่ยว
8. จังหวัดที่ต้องการไปท่องเที่ยว

ภาพ 27 แสดงตัวอย่างการออกแบบหน้าจอการสืบค้นโดยใช้คำค้นหาหรือประโยค

จากภาพ 27 เมื่อผู้ใช้ป้อนคำค้นหาหรือประโยคเพื่อสืบค้นข้อมูลท่องเที่ยว จากนั้นกดปุ่ม Enter หรือ กดปุ่มค้นหา ระบบจะแสดงผลการสืบค้นตามที่ผู้ใช้ป้อน ดังรายละเอียดที่จะกล่าวต่อไปในบทที่ 4



ภาพ 28 แสดงตัวอย่างการออกแบบหน้าจอเลือกข้อมูลท่องเที่ยว ตามความต้องการของผู้ใช้

จากภาพ 28 เมื่อผู้ใช้เลือกข้อมูลท่องเที่ยวที่สนใจ ประกอบด้วย สถานที่ท่องเที่ยว ร้านอาหาร ที่พัก สิ่งอำนวยความสะดวก กิจกรรมที่สนใจ เพศ อายุ วันที่เดินทาง จำนวนวันท่องเที่ยวและจังหวัดที่ต้องการไป โดยระบบได้ออกแบบให้ผู้ใช้สามารถเลือกข้อมูลต่างๆ ได้มากกว่า 1 คน ด้วยการกดปุ่มเพิ่มข้อมูล จากนั้นกดปุ่มค้นหา เพื่อให้ระบบแนะนำข้อมูลท่องเที่ยวตามความต้องการของผู้ใช้ ดังรายละเอียดที่จะกล่าวต่อไปในบทที่ 4

โดยงานวิจัยนี้ใช้ภาษา SPARQL (ภาพ 29 ส่วนที่ 1) ในการสืบค้นข้อมูลท่องเที่ยว ซึ่งผู้ใช้จะป้อนชื่อจังหวัดหรือสถานที่ท่องเที่ยวหรือประโยคสอบถามที่ต้องการ ดังได้กล่าวไว้แล้วในภาพ 27 จากนั้นใช้ Jena API ในการเชื่อมต่อกับออนโทโลจีท่องเที่ยวที่ถูกเก็บในรูปแบบไฟล์ OWL ซึ่งมีรูปแบบการใช้งานคำสั่ง SPARQL 2 รูปแบบ คือ กรณีค้นพบข้อมูลสถานที่ท่องเที่ยวที่เดียวและกรณีค้นพบข้อมูลสถานที่ท่องเที่ยวมากกว่า 1 รายการ ดังรายละเอียดโปรแกรมในภาพ 29 โดยระบบจะสืบค้นคำค้นหาที่ผู้ใช้ต้องการว่ามีจำนวนกี่รายการ หากไม่พบหรือไม่มีรายการตามคำค้นหานั้น ระบบจะแสดงข้อความ "ไม่พบคำค้นหาที่ผู้ใช้ต้องการ" หากพบ 1 รายการระบบจะแสดงข้อมูลตามที่ผู้ใช้ต้องการทันทีและหากพบรายการมากกว่า 1 รายการ ระบบจะแสดงตัวเลือกเพื่อให้ผู้ใช้เลือกข้อมูลตามต้องการ (ภาพ 29 ส่วนที่ 2 การแสดงผล)

```
String inputFileName="C:/ontology/tourism.owl", lng = "th";
Model model = ModelFactory.createDefaultModel();
InputStream in = FileManager.get().open(inputFileName); model.read(in," ");
String sparql = prefix + "SELECT (COUNT(*) AS ?count) " +
        "WHERE {?a myont:hasNameOfAttraction ?count; myont:hasNameOfAttraction
        ?name; " +
         ". FILTER langMatches( lang(?name), '" +  lng + "') " +
        ". FILTER (regex(str(?name),'" + keyword + "', \"i\")}";
query = QueryFactory.create(sparql);
qe = QueryExecutionFactory.create(query,model);   results = qe.execSelect();
if (results.hasNext()) {
        QuerySolution row = results.next();
        int rowcount = Integer.parseInt(row.getLiteral("count").getValue().toString());
        if (rowcount == 0) out.print("ไม่พบคำค้นหา " + keyword);
        if (rowcount == 1) dispInfo();
        else dispSelectInfo();
} else out.print("ไม่พบคำค้นหา " + keyword);
```

1. SPARQL

2. การแสดงผล

ภาพ 29 แสดงตัวอย่างการใช้คำสั่ง SPARQL เพื่อสืบค้นข้อมูลสถานที่ท่องเที่ยว

การแนะนำข้อมูลท่องเที่ยวด้วยภาษา SWRL โดยจะต้องโหลดออนโทโลจีท่องเที่ยวและกำหนดข้อมูลนักท่องเที่ยวดังภาพ 30 ที่แสดงตัวอย่างการสืบค้นข้อมูลท่องเที่ยวน้ำตกในจังหวัดพิษณุโลก โดยมีตัวแปรเป็นนักท่องเที่ยวที่มีอายุระหว่าง 19 ถึง 25 ปี จากนั้นทำการประมวลผลและแสดงผลการแนะนำข้อมูลท่องเที่ยวดังภาพ 31

```
String uri ="file:///C:/Ontology/tourism.owl"; // Load Ontology - โหลดออนโทโลจีท่องเที่ยว
OWLModel owlModel = ProtegeOWL.createJenaOWLModelFromURI(uri);
// Get Tourist Class - ดึงโครงสร้างคลาสนักท่องเที่ยว
OWLNamedClass touristClass = owlModel.getOWLNamedClass("Tourist");
OWLObjectProperty ageProperty = owlModel.getOWLObjectProperty("hasAgeRank");
ageProperty.setDomain(touristClass);
OWLObjectProperty genderProperty = owlModel.getOWLObjectProperty("hasGender");
genderProperty.setDomain(touristClass);
// Add Individual - สร้างข้อมูลนักท่องเที่ยว ประกอบไปด้วย อายุ และ เพศ
OWLIndividual individual1 = touristClass.createOWLIndividual("Naruepon.p");
individual1.setPropertyValue(ageProperty, "Age19-25");
individual1.setPropertyValue(genderProperty, "Male");
```

ภาพ 30 แสดงคำสั่งเพื่อโหลดออนโทโลจีท่องเที่ยวและกำหนดข้อมูลนักท่องเที่ยว

```
// Connect Jess Inference Engine – เชื่อมต่อ Jess Inference Engine
SQWRLQueryEngine queryEngine=P3SQWRLQueryEngineFactory.create("Jess",owlModel);
// SQWRL Command – คำสั่ง SQWRL
queryEngine.createSQWRLQuery("Query-1","Tourist(?x) ^ hasAgeRank(?x,Age19-25)  ^
            isLocatedInProvince(?a,'Phitsanulok') ^ Waterfall(?a) -> sqwrl:select(?a)");
Result result = queryEngine.runSQWRLQuery("Query-1");
while (result.hasNext()) { // Display Result - แสดงผลการแนะนำข้อมูลท่องเที่ยวน้ำตก
        IndividualValue aValue = result.getObjectValue("?a");
        System.out.println(aValue.toString()); result.next();
}
```

ภาพ 31 แสดงการใช้ Protégé API ประมวลผลภาษา SWRL ด้วย Jess Inference Engine