การสกัดตารางและรายการบนเว็บเป็นอาร์ดีเอฟ

นายจุลเทพ นันทขว้าง

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรดุษฎีบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2563

EXTRACTION OF TABLES AND LISTS ON THE WEB TO RDF

Mr. Julthep Nandakwang

A Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy (Computer Engineering) in Computer Engineering

Department of Computer Engineering
FACULTY OF ENGINEERING
Chulalongkorn University

Academic Year 2020

หัวข้อวิทยานิพนธ์	การสกัดตารางและรายการบนเว็บเป็นอาร์ดีเอฟ
โดย	นายจุลเทพ นันทขว้าง
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	ศาสตราจารย์ ดร.ประภาส จงสถิตย์วัฒนา

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรดุษฎีบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(ศาสตราจารย์ ดร.สุพจน์ เตชวรสินสกุล)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(รองศาสตราจารย์ ดร.อัศนีย์ ก่อตระกูล)

..... อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(ศาสตราจารย์ ดร.ประภาส จงสถิตย์วัฒนา)

..... กรรมการ
(ศาสตราจารย์ ดร.บุญเสริม กิจศิริกุล)

..... กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.สุกรี สินธุภิญโญ)

..... กรรมการ
(อาจารย์ ดร.ดวงดาว วิชาดากุล)

..... กรรมการภายนอกมหาวิทยาลัย
(รองศาสตราจารย์ ดร.อัศนีย์ ก่อตระกูล)

จุลเทพ นันทขว้าง : การสกัดตารางและรายการบนเว็บเป็นอาร์ดีเอฟ. (EXTRACTION OF TABLES AND LISTS ON THE WEB TO RDF) อ.ที่ปรึกษาหลัก : ศ. ดร.ประภาส จงสถิตย์วัฒนา

ทุกวันนี้ ลิงก์เดต้าได้เติบโตเพิ่มขึ้นอย่างรวดเร็วตามการเติบโตของเว็บ นอกเหนือจากข้อมูลใหม่ที่สร้างขึ้นในรูปแบบซีแมนติกโดยเฉพาะ ส่วนหนึ่งมาจากการแปลงข้อมูลโครงสร้างที่มีอยู่ให้อยู่ในรูปแบบของข้อมูลเปิดระดับห้าดาว อย่างไรก็ตามยังคงมีข้อมูลจำนวนมากในรูปแบบโครงสร้างและกึ่งโครงสร้าง ตัวอย่างเช่นตารางและรายการซึ่งเป็นรูปแบบหลักที่มนุษย์ใช้อ่าน ยังรอการแปลงอยู่ งานวิจัยนี้กล่าวถึงงานวิจัยต่าง ๆ ที่เกี่ยวกับการแปลงตารางและรายการมาเป็นข้อมูลในรูปแบบต่าง ๆ เพื่อให้เครื่องสามารถอ่านได้ นอกจากนี้ยังเสนอวิธีการในการแปลงตารางและรายการเป็นรูปแบบ Resource Description Framework และยังคงเก็บโครงสร้างต้นฉบับที่จำเป็นไว้อย่างละเอียด ซึ่งทำให้สามารถที่จะสร้างข้อมูลโครงสร้างเดิมกลับมาได้ ระบบ TULIP ถูกสร้างขึ้นเพื่อเป็นเครื่องมือสำหรับการพัฒนาซีแมนติกเว็บ วิธีการที่เสนอมีความยืดหยุ่นมากกว่าเมื่อเทียบกับงานอื่น ๆ เดต้าโมเดลของ TULIP สามารถรองรับการเก็บข้อมูลต้นฉบับอย่างครบถ้วนและสามารถนำมาแสดงใหม่ในมุมมองที่แตกต่างไปจากเดิม เครื่องมือนี้สามารถใช้สร้างข้อมูลจำนวนมหาศาลสำหรับเครื่องคอมพิวเตอร์เพื่อให้ใช้งานได้กว้างมากขึ้นกว่าเดิม

สาขาวิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนิสิต .....

ปีการศึกษา 2563 ลายมือชื่อ อ.ที่ปรึกษาหลัก .....

5771461321 : MAJOR COMPUTER ENGINEERING

KEYWORD: Semantic Web, Linked Data, Knowledge Representation, Table, List, Data Extraction, Knowledge Extraction, Data Transformation

Julthep Nandakwang : EXTRACTION OF TABLES AND LISTS ON THE WEB TO RDF. Advisor: Prof. PRABHAS CHONGSTITVATANA, Ph.D.

Currently, Linked Data is increasing at a rapid rate as the growth of the Web. Aside from new information that has been created exclusively as Semantic Web-ready, part of them comes from the transformation of existing structural data to be in the form of five-star open data. However, there are still many legacy data in structured and semi-structured form, for example, tables and lists, which are the principal format for human-readable, waiting for transformation. This work discusses attempts in the research area to transform table and list data to make them machine-readable in various formats. Furthermore, the research proposes a method for transforming tables and lists into Resource Description Framework format while maintaining their essential configurations thoroughly. It is possible to recreate their original form back informatively. A system named TULIP has been developed which embodied this conversion method as a tool for the future development of the Semantic Web. The proposed method is more flexible compared to other works. The TULIP data model contains complete information of the source; hence it can be projected into different views. This tool can be used to create a tremendous amount of data for machines to be used at a broader scale.

Field of Study: Computer Engineering Student's Signature .....

Academic Year: 2020 Advisor's Signature .....

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จได้เนื่องจากได้รับความกรุณาจาก ศ. ดร. ประภาส จงสถิตย์วัฒนา รับเป็นที่ปรึกษาและให้คำแนะนำ จนทำให้วิทยานิพนธ์ฉบับนี้เสร็จสิ้นสมบูรณ์ อีกทั้งขอขอบพระคุณ รศ. ดร. อัศนีย์ ก่อตระกูล, ศ. ดร. บุญเสริม กิจศิริกุล, ผศ. ดร. สุกรี สินธุภิญโญ และ อ. ดร. ดวงดาว วิชาดากุล ซึ่งเป็นประธานและกรรมการสอบ ที่ได้เสียสละเวลา ให้คำชี้แนะและชี้ข้อบกพร่อง รวมถึงแนะนำแนวทางในการวิจัยเพื่อให้งานวิจัยนี้สำเร็จลุล่วงไปได้เป็นอย่างดี

อีกทั้งขอขอบคุณ คุณกนิษฐ เมืองกระจ่าง ที่ให้การสนับสนุน และเพื่อน ๆ ทุกคนที่ฝ่ายสารสนเทศ บริษัท ไทยโตชิบาอุตสาหกรรม จำกัด ที่ให้คำชี้แนะ และคอยเป็นธุระช่วยเหลือดูแลการงานในขณะที่ดำเนินงานวิจัย

สุดท้ายนี้ขอกราบขอบพระคุณ คุณพ่อทวี คุณแม่สุนิตย์ ที่ผลักดันให้เป็นนักวิทยาศาสตร์ และขอขอบคุณภรรยา จิรนุช ลูกสาว นลินา และลูกชาย จิรณัฐ ที่คอยเป็นห่วงเป็นใย ดูแล และให้กำลังใจมาโดยตลอด

จุลเทพ นันทขว้าง

สารบัญ

หน้า

สารบัญตาราง

หน้า

สารบัญภาพ

หน้า

บทที่ 1

บทนำ

ที่มาและความสำคัญของปัญหา

ปัจจุบันวิกิ (wiki) ถูกนำไปใช้อย่างกว้างขวางในฐานะแพลทฟอร์มหนึ่งของระบบจัดการความรู้ (knowledge management) และถูกนำไปใช้ในองค์กรธุรกิจทั้งเล็กและใหญ่รวมไปถึงองค์กรสาธารณะ เช่น วิกิพีเดีย (Wikipedia) แนวคิดหลักของวิกิพีเดียคือการรวบรวมความรู้ในแทบทุกโดเมนโดยให้ผู้ใช้ (ซึ่งเรียกว่า “วิกิพีเดียน”) แบ่งปันความรู้และช่วยกันตรวจทานเนื้อหาในระบบ ปัญหาสำคัญของวิกิพีเดียอยู่ที่วิธีการนำเสนอข้อมูลความรู้ในรูปแบบที่ผู้ใช้สามารถอ่านและแก้ไขได้โดยง่ายเป็นธรรมชาติแบบที่มนุษย์ใช้ในชีวิตประจำวัน แต่เป็นการยากสำหรับเครื่องคอมพิวเตอร์ที่จะสกัด วิเคราะห์ หรืออนุมานความรู้เหล่านั้นเพื่อนำมาประมาลผลสังเคราะห์เป็นบทสรุปหรือข้อเท็จจริงเพื่อเป็นองค์ความรู้ต่อไป ดังนั้นเนื้อหาในวิกิพีเดียจึงจำเป็นต้องถูกแปลงให้อยู่ในรูปแบบที่เครื่องคอมพิวเตอร์สามารถ “เข้าใจ” หรือจัดการได้โดยง่ายโดยการเพิ่มเติม “ความหมาย” ลงไปในเนื้อหาที่อยู่ในรูปแบบต่าง ๆ เช่น ข้อความ ตาราง รายการ รูปภาพ และเสียง เป็นต้น

งานวิจัยนี้มีวัตถุประสงค์เพื่อศึกษาถึงเทคนิควิธีการต่าง ๆ ที่เกี่ยวข้องกับการนำข้อมูลหลายส่วนจากวิกิพีเดียมาใช้ รวมถึงนำเสนอวิธีการสกัดข้อมูลในรูปแบบตารางและรายการที่อยู่ในวิกิพีเดียหรือระบบอื่น ๆ ที่ทำงานโดยซอฟท์แวร์ประเภทเดียวกันกับวิกิพีเดีย โดยมีเป้าหมายที่จะนำมาซึ่งข้อมูลลิงก์เดต้าที่ปัจจุบันนี้ยังคงขาดหายไปจากการสกัดข้อมูลความรู้จากวิกิพีเดียด้วยวิธีการอื่น ๆ ซึ่งในท้ายที่สุด จะเปลี่ยนแปลง “ความรู้ที่สร้างโดยมนุษย์สำหรับมนุษย์” มาเป็น “ความรู้ที่สร้างโดยมนุษย์สำหรับเครื่องจักร”

วัตถุประสงค์ของงานวิจัย

งานวิจัยนี้มีวัตถุประสงค์เพื่อศึกษาวิธีการในการสกัดข้อมูลจากตารางและรายการในรูปแบบต่าง ๆ และนำข้อมูลนั้นมาแปลงรูปให้อยู่ในรูปแบบลิงก์เดต้า โดยใช้เทคโนโลยีซีแมนติกเว็บ

มีข้อมูลมากมายในวิกิพีเดียที่อยู่ในรูปแบบตารางและรายการ เนื่องจากข้อมูลสำคัญบางประเภทไม่สามารถเขียนให้อยู่ในรูปแบบเรียงความได้ จึงได้ถูกนำเสนอในรูปแบบตารางและรายการ ยิ่งไปกว่านั้นบางบทความของวิกิพีเดียจะมีเฉพาะตารางหรือรายการเท่านั้น ตัวอย่างเช่น จะมีบทความอยู่ส่วนหนึ่งที่เรียกว่า stand-alone list[1] คือบทความที่มีชื่อเริ่มต้นด้วย “List of …” (หรือในบางกรณีจะเป็น “Index of …”, “Timeline of …”, “Outline of …” หรือ “Glossary of …” เป็นต้น) เห็นได้ชัดว่าเนื้อหาในบทความนั้นจะมีเฉพาะรายการเท่านั้น ซึ่งอาจจะอยู่ในรูปแบบรายการหรือตาราง หรือทั้งสองอย่างรวมกันในบทความเดียวกันก็ได้ ดังนั้นหากเราสามารถสกัดข้อมูลจากตารางและรายการในวิกิพีเดียให้อยู่ในรูปแบบลิงก์เดต้า จะทำให้เกิดข้อมูลเชิงซีแมนติกมากมายสำหรับแอปพลิเคชันทางด้านซีแมนติกเว็บต่าง ๆ และทำให้การใช้งานซีแมนติกเว็บเป็นไปอย่างมีประสิทธิภาพมากขึ้น

นอกจากนี้ ผู้วิจัยได้นำเสนอโมเดลการรีพรีเซนต์ตารางและรายการในรูปแบบลิงก์เดต้า รวมทั้งอัลกอริทึมเพื่อจะสกัดข้อมูลเหล่านั้นออกมาจากวิกิพีเดีย ในระบบที่ชื่อว่า TULIP ซึ่งประกอบไปด้วย TULIP RDF data model และ TULIP vocabulary สำหรับแปลงข้อมูลจาก DOM (Document Object Model) มาเป็นรูปแบบ RDF พร้อมทั้งยกตัวอย่างการใช้งานโดยสังเขป โดยมีเป้าหมายเพื่อสร้างเฟรมเวิร์คที่จะนำความรู้ที่ซ่อนอยู่ในวิกิพีเดียมาเพื่อทำให้คอมพิวเตอร์สามารถเข้าใจและประมวลผลในเชิงซีแมนติกได้อย่างสะดวกและถูกต้องมากยิ่งขึ้น นอกจากนั้นยังจะได้เดต้าเซ็ตขนาดใหญ่ที่สกัดได้มาจากตารางและรายการในวิกิพีเดียเพื่อนำไปประยุกต์ใช้ในแอปพลิเคชันด้านซีแมนติกเว็บต่าง ๆ

ขอบเขตงานวิจัย

1. ใช้เฉพาะข้อมูลวิกิพีเดียที่อยู่ในภาษาอังกฤษเท่านั้น เพื่อลดความหลากหลายของการนำเข้าข้อมูล
2. ระบบสามารถทำงานได้ตลอด 24 ชั่วโมง เฉพาะบนเซิร์ฟเวอร์ทดสอบที่สร้างมาเพื่องานวิจัยนี้โดยเฉพาะ
3. ผู้วิจัยเป็นผู้กำหนดขอบเขตให้ระบบ เพื่อสกัดเฉพาะข้อมูลที่สนใจทำการทดลองเท่านั้น

[1] Stand-alone lists. Wikipedia. Available from: https://en.wikipedia.org/wiki/Wikipedia:Stand-alone_lists

ขั้นตอนวิจัย

1. ศึกษาและทำความเข้าใจเกี่ยวกับเทคโนโลยีซีแมนติกเว็บและเทคโนโลยีที่เกี่ยวข้อง
2. ศึกษางานวิจัยที่เกี่ยวกับการนำข้อมูลจากซีแมนติกเว็บมาใช้งานในรูปแบบต่าง ๆ
3. ศึกษาเทคนิควิธีการต่าง ๆ ในการสืบค้นและนำเข้าข้อมูลจากเว็บและวิกิพีเดีย
4. ศึกษางานวิจัยและระบบซอฟท์แวร์ที่เกี่ยวข้องกับการรีพรีเซนต์ข้อมูลตารางและรายการ
5. ศึกษาและทำความเข้าใจเกี่ยวกับการพัฒนาซอฟท์แวร์ที่ทำงานบนเว็บแพลตฟอร์ม
6. ศึกษาและทำความเข้าใจเกี่ยวกับเครื่องมือพัฒนาทางด้านซีแมนติกเว็บ
7. ออกแบบและพัฒนาระบบสืบค้นและสกัดข้อมูลตารางและรายการจากวิกิพีเดีย และแปลงข้อมูลให้อยู่ในรูปแบบลิงก์เดต้า
8. ทดสอบและประเมินผลระบบที่พัฒนา
9. ตีพิมพ์ผลงานทางวิชาการ
10. สรุปผลงานวิจัย ข้อเสนอแนะ แนวทางวิจัยต่อเนื่อง และจัดทำวิทยานิพนธ์

ประโยชน์ที่คาดว่าจะได้รับจากงานวิจัย

1. ได้ศึกษาแนวคิดของการสืบค้น สกัด และแปลงข้อมูลกึ่งโครงสร้าง มาเป็นข้อมูลเชิงโครงสร้าง
2. ได้ศึกษาแนวคิดและออกแบบการรีพรีเซนต์ข้อมูลตารางและรายการในรูปแบบลิงก์เดต้า
3. ได้เครื่องมือต้นแบบที่ใช้ในการสืบค้น สกัด และแปลงข้อมูลตารางและรายการจากวิกิพีเดียมาเป็นข้อเท็จจริงในรูปแบบลิงก์เดต้า
4. ได้เดต้าเซ็ตของข้อเท็จจริงจำนวนหนึ่งที่สามารถใช้งานได้จริงเพื่อสนับสนุนแอปพลิเคชันทางด้านซีแมนติกเว็บในระบบงานต่าง ๆ

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

เว็บได้มีการพัฒนาผ่านมาหลายยุค เนื้อหาข้อมูลต่าง ๆ ในเว็บก็มีการเปลี่ยนแปลงทั้งรูปแบบและวิธีการมาโดยตลอด การค้นหาด้วยคีย์เวิร์ดมีปัญหาหลายอย่าง ในอนาคตเราจะมีข้อมูลมากมายมหาศาล ทำให้การค้นหาแบบเดิมจะไม่เพียงพอ การแปลงข้อมูลที่มีมากมายอยู่แล้วให้เป็นรูปแบบซึ่งทำให้การค้นหาอย่างเฉลียวฉลาดได้ เว็บในอนาคตนี้เรียกว่าซีแมนติกเว็บ

เนื้อหาในบทนี้จะแนะนำซีแมนติกเว็บโดยสังเขป และชนิดของข้อมูลพื้นฐานในซีแมนติกเว็บที่เรียกว่า RDF เนื่องจากมีรูปแบบต่าง ๆ มากมายของข้อมูลซึ่งอยู่ในเว็บ จึงมีงานวิจัยจำนวนมากที่จะแปลงเนื้อหาเหล่านี้ให้เป็นรูปแบบที่ค้นหาได้โดยใช้เทคโนโลยีซีแมนติกเว็บ เราจะกล่าวถึงข้อกำหนดการเปิดเผยข้อมูลในลิงก์เดต้า และการทำให้เนื้อหาในเว็บทั่วไปให้เป็นข้อมูลที่เครื่องสามารถเข้าใจได้ (machine understandable data)

ข้อมูลเชิงโครงสร้าง, กึ่งโครงสร้าง และไร้โครงสร้าง

หากมองในแง่ของความสะดวกในการนำข้อมูลมาประมวลผลโดยเครื่องคอมพิวเตอร์ ข้อมูลสามารถแบ่งออกได้เป็น 3 ประเภทคือ

1. ข้อมูลเชิงโครงสร้าง (structured data) เป็นข้อมูลที่มีโครงสร้างชัดเจนตายตัวอยู่แล้ว เช่น ข้อมูลที่อยู่ในฐานข้อมูลเชิงสัมพันธ์ (relational database) ข้อมูลประเภทนี้สามารถนำมาประมวลผลต่อได้โดยตรง

2. ข้อมูลกึ่งโครงสร้าง (semi-structured data) เป็นข้อมูลที่ไม่สามารถมองหาโครงสร้างได้อย่างชัดเจนนัก เช่น ตาราง รายการ แผนภูมิ เป็นต้น ถึงแม้ว่าข้อมูลเหล่านี้มนุษย์จะสามารถมองเห็น “โครงสร้าง” ได้ (ไม่ว่าจะดูซับซ้อนเพียงใด อย่างเช่น ข้อมูลในรูปแบบอินโฟกราฟฟิค) และมนุษย์สามารถเข้าใจได้โดยง่าย แต่เนื่องจากข้อมูลประเภทนี้มีความไม่แน่นอนและความกำกวมทั้งในแง่ของโครงสร้างและความหมาย จึงจำเป็นต้องได้รับ

การจัดการโดยกระบวนวิธีต่าง ๆ เพื่อแปลงข้อมูลก่อน จึงสามารถนำมาประมวลผลต่อไปได้

3. ข้อมูลไร้โครงสร้าง (unstructured data) เป็นข้อมูลที่ไม่สามารถมองหาโครงสร้างได้อย่างโจ่งแจ้ง เช่น ตัวหนังสือในรูปแบบเรียงความ รูปภาพ เสียง วิดิโอ ฯลฯ ต้องนำมาประมวลผลก่อนโดยวิธีการที่เฉพาะเจาะจง เช่น การประมวลภาษาธรรมชาติ (natural language processing หรือ NLP) และวิธีการอื่น ๆ เพื่อแปลงข้อมูลนั้นให้อยู่ในรูปแบบที่เครื่องคอมพิวเตอร์สามารถนำไปใช้งานได้ต่อไป ข้อมูลประเภทนี้จะมีความไม่แน่นอนและความกำกวมสูงที่สุด

เวิร์ลไวด์เว็บ

เวิร์ลไวด์เว็บ (World Wide Web[1] หรือ WWW) คือการเชื่อมโยงรีซอร์สต่าง ๆ บนอินเตอร์เน็ตเข้าด้วยกัน โดยในเบื้องต้นใช้แนวคิด hypertext[2] ในการเชื่อมโยงเอกสารที่กระจายอยู่ในเซิร์ฟเวอร์ที่กระจายกันอยู่ทั่วโลก เซอร์ทิม เบอร์เนอร์ส-ลี (Tim Berners-Lee) เป็นผู้คิดค้นเวิลด์ไวด์เว็บในปี ค.ศ. 1989[1, 2] ในขณะทำงานอยู่ที่ European Organization for Nuclear Research (CERN) จากนั้นได้เป็นผู้ก่อตั้ง World Wide Web Consortium (W3C) ที่ MIT หลังออกจาก CERN ในปี ค.ศ. 1994

สามสิ่งสำคัญที่ประกอบกันขึ้นเป็นเวิร์ลไวด์เว็บคือ

- Uniform Resource Identifier
- Hypertext Transfer Protocol
- Hypertext Markup Language

1. Uniform Resource Identifier (URI)

Uniform Resource Identifier หรือ URI (ในช่วงเริ่มแรกถูกเรียกว่า Universal Resource Identifier[3]) หน่วยย่อยที่สุดของเวิร์ลไวด์เว็บ เป็นการกำหนดชื่อให้กับรีซอร์สหรือทรัพยากรแต่ละตัวในระบบเวิร์ลไวด์เว็บด้วยสายอักขระ โดยที่ชื่อนั้นจะมีความเฉพาะจงและไม่มีโอกาสซ้ำกันกับรีซอร์ส

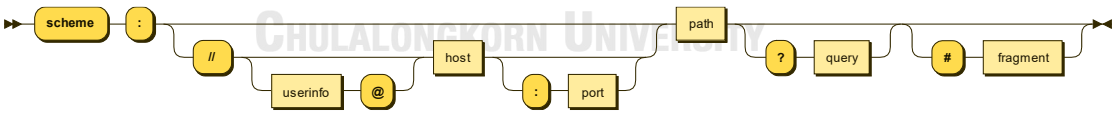
[1] รายละเอียดของเวิร์ลไวด์เว็บอธิบายอยู่ในเว็บไซต์แรกของโลกที่ CERN ซึ่งปัจจุบันนี้ยังอยู่ในรูปแบบดั้งเดิมเมื่อ 30 ปีที่แล้ว. แหล่งที่มา: http://info.cern.ch/hypertext/WWW/

[2] Hypertext. Wikipedia. Available from: https://en.wikipedia.org/wiki/Hypertext

[3] RFC 1630 - Universal Resource Identifiers in WWW: A Unifying Syntax for the Expression of Names and Addresses of Objects on the Network as used in the World-Wide Web. T. Berners-Lee. Available from: https://tools.ietf.org/html/rfc1630

อื่นใดในเครือข่ายอินเตอร์เน็ต ซึ่งรับประกันความเป็นหนึ่งเดียวด้วยกลไกการตั้งชื่อที่เป็นลำดับชั้น และในขณะเดียวกันก็เอื้ออำนวยให้สามารถเพิ่มขยายได้อย่างไม่มีขีดจำกัด (เปรียบเทียบแล้วจะแตกต่างจากการตั้งรหัสแบบอื่นที่มีจำนวนจำกัด เช่น ISBN – International Standard Book Number และ UPC – Universal Product Code ที่ใช้ระบุจำแนกชื่อหนังสือและผลิตภัณฑ์ หรือ IP address ทั้งเวอร์ชัน 4 และเวอร์ชัน 6 ซึ่งถึงแม้จะมีการกล่าวอ้างว่าสามารถมีได้มากกว่าจำนวนเม็ดทรายในโลกนี้ก็ตาม[4]) โดยจะแบ่งออกเป็นสองประเภทย่อยคือ Uniform Resource Location หรือ URL และ Uniform Resource หรือ URN ซึ่งมีความแตกต่างกันตรงที่ URL เป็นการระบุ “location” หรือ “ที่อยู่” ที่สามารถเข้าถึงได้ของรีซอร์ส ในขณะที่ URN เป็นการระบุ “name” หรือ “ชื่อ” ของรีซอร์สไม่ว่าจะเป็นสิ่งของในโลกจริงที่เป็นรูปธรรมหรือสิ่งที่เป็นนามธรรมก็ได้ แต่ในทางปฏิบัติเราก็สามารถใช้ URL ในการระบุ “ชื่อ” ของสิ่งที่เป็นรูปธรรมและนามธรรมโดยที่ไม่จำเป็นต้องให้มีการเข้าถึง “ที่อยู่” นั้นได้ทางอิเล็กทรอนิกส์ ซึ่งในกรณีหลังเรามักจะเรียก URL นั้นโดยรวมว่าเป็น URI เพื่อให้เห็นถึงการระบุ “identifier” อย่างชัดเจน ถึงแม้ว่ารูปแบบจะเป็น URL ไม่ใช่ URN อย่างไรก็ตามเนื่องจากใน RFC หลายตัวที่เกี่ยวกับเรื่องนี้ไม่ได้นิยามไว้อย่างครอบคลุมทุกกรณีและไม่สอดคล้องกันในบางจุด จึงยังมีความกำกวมถึงความถูกต้องในการเรียกชื่อทั้ง 3 สิ่งนี้กันอยู่[5] โดยในกรณีของซีแมนติกเว็บ เราใช้การระบุรีซอร์สทุกสิ่งทุกอย่างในรูปแบบของ URL เป็นส่วนใหญ่ และเรียกโดยรวมว่า URI ภาพที่ 1 แสดงถึงแผนภาพไวยากรณ์ของ URL ซึ่งมีตัวอย่างดังนี้

```
http://www.student.chula.ac.th/~57714613/index.html
http://dx.doi.org/10.13140/2.1.1205.6328
https://tools.ietf.org/html/rfc1737
```



ภาพที่ 1 แผนภาพไวยากรณ์ของ URL[6]

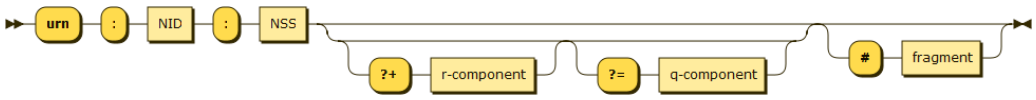
ในขณะที่แผนภาพไวยากรณ์ของ URN เป็นดังภาพที่ 2 ซึ่งมีตัวอย่างดังนี้

```
urn:ietf:rfc:1737
urn:isbn:0062515861
urn:iso:std:iso-iec:8859:-1:ed-1:en
```

[4] Internet traffic jam to be unstuck. ITworld. Available from: https://www.itworld.com/article/2794384/internet-traffic-jam-to-be-unstuck.html

[5] What's the Difference Between URI and URL? Daniel Miessler. Available from: https://danielmiessler.com/study/difference-between-uri-url/

[6] Uniform Resource Locator. Wikipedia. Available from: https://en.wikipedia.org/wiki/Uniform_Resource_Locator



ภาพที่ 2 แผนภาพไวยากรณ์ของ URN[7]

2. Hypertext Transfer Protocol (HTTP)

Hypertext Transfer Protocol หรือ HTTP[8] เป็นโปรโตคอลในระดับชั้นแอปพลิเคชัน ถูกออกแบบเพื่อใช้ในการร้องขอและนำส่งเอกสารโฮเปอร์เท็กซ์ รวมทั้งไฟล์สื่อต่าง ๆ ที่เกี่ยวข้อง เช่น รูปภาพ, ภาพเคลื่อนไหว และเสียง เป็นต้น เวิร์ลไวด์เว็บในช่วงแรกใช้ HTTP เวอร์ชันทดลองจนมาหยุดอยู่ที่เวอร์ชัน HTTP/0.9 ที่ออกมาในปี ค.ศ. 1991 และใช้ต่อเนื่องมา 5 ปีจึงได้ออกเวอร์ชันแรกที่กำหนดเป็นมาตรฐาน RFC 1945 คือ HTTP/1.0 ในปี ค.ศ. 1996 ซึ่งได้ถูกปรับปรุงแก้ไขเป็น HTTP/1.1 ในอีกหนึ่งปีถัดมา และเวิร์ลไวด์เว็บก็ใช้ HTTP เวอร์ชันนี้ตลอดมาเป็นเวลาเกือบ 20 ปี จึงได้มี HTTP/2.0 ออกมาในปี ค.ศ. 2015 ซึ่งต่อมาก็ได้ถูกพัฒนาเป็น HTTP/3.0 ในปี ค.ศ. 2018 อย่างไรก็ตามเวิร์ลไวด์เว็บโดยส่วนใหญ่ที่ใช้อยู่ทุกวันนี้ยังคงส่งผ่านข้อมูลกันด้วย HTTP/1.1 เป็นหลัก

3. Hypertext Markup Language (HTML)

Hypertext Markup Language หรือ HTML เป็นซีเรียลไลซ์ไฟล์ฟอร์แมตในการเก็บและส่งข้อมูลไฮเปอร์เท็กซ์ผ่านทาง HTTP บนอินเตอร์เน็ต HTML พัฒนามาจาก SGML ที่ตอนนั้นเป็น markup language มาตรฐาน ISO อยู่แล้ว[9] เมื่อ HTML เข้าไปอยู่ในหน่วยความจำจะอยู่ในโครงสร้างต้นไม้ที่เรียกว่า Document Object Model (DOM) เพื่อให้สะดวกในการบริหารจัดการ

4. Extensible Markup Language (XML)

Extensible Markup Language หรือ XML[10] เป็นซีเรียลไลซ์ไฟล์ฟอร์แมทที่มีโครงสร้างแบบยืดหยุ่นได้ เนื่องจากออกแบบให้รองรับการใช้งานได้หลากหลายรูปแบบ จึงมีการนำ XML ไปใช้ในการรับส่งข้อมูลในหลายแอปพลิเคชัน เช่น เว็บเซอร์วิสผ่านทาง SOAP[11] (Simple Object Access Protocol), SVG[12] (Scalable Vector Graphics), XHTML[13] (Extensible HyperText

---

[7] Uniform Resource Name. Wikipedia. Available from: https://en.wikipedia.org/wiki/Uniform_Resource_Name

[8] RFC 1945 - Hypertext Transfer Protocol -- HTTP/1.0. T. Berners-Lee, R. Fielding and H. Frystyk. Available from: https://tools.ietf.org/html/rfc1945

[9] ISO 8879:1986 - Information processing — Text and office systems — Standard Generalized Markup Language (SGML). ISO. Available from: https://www.iso.org/standard/16387.html

[10] Extensible Markup Language (XML) 1.0. W3C. Available from: https://www.w3.org/TR/xml/

[11] SOAP Specifications. W3C. Available from: https://www.w3.org/TR/soap/

[12] SVG Working Group. W3C. Available from: https://www.w3.org/Graphics/SVG/

[13] XHTML 1.0: The Extensible HyperText Markup Language. W3C. Available from: https://www.w3.org/TR/xhtml1/

Markup Language ซึ่งปัจจุบันถูกยุบรวมเข้ากับ HTML5[14]) รูปแบบไฟล์ของชุดโปรแกรม Office ต่าง ๆ รวมไปถึงซีแมนติกเว็บก็ใช้ XML ในการส่งผ่านข้อมูล ซึ่งจะกล่าวถึงในลำดับถัดไป

5. XML namespace, QName และ CURIE

XML namespace[15] เป็นการกำหนด prefix เพื่อนำไปใช้กับ URI ในเอกสาร XML เพื่อให้เอกสารกระชับและใช้งานได้สะดวก โดยการตั้งชื่อให้กับส่วนหัวของ element และ attribute ต่าง ๆ ที่มีที่อยู่แตกต่างกัน ด้วย prefix ที่ไม่ซ้ำกัน แล้วจากนั้นเมื่อเรานำไปใช้ จะใช้เพียงแค่รูปแบบสั้นคือ prefix:local ซึ่งจะเรียกว่า Qualified Name หรือ QName[16] หลังจากนั้นระบบจะสามารถแทน prefix เหล่านั้นรวมกับส่วนท้ายหรือ local name เพื่อกำหนดชื่อรีซอร์สแบบเต็มให้ ส่วน Compact URI หรือ CURIE[17] จะมีข้อแตกต่างจาก QName เล็กน้อยเนื่องจากถูกกำหนดมาทีหลัง โดยใช้งานได้กว้างขวางมากขึ้นเนื่องจากตัดข้อจำกัดบางอย่างของ XML ออกไป และถูกออกแบบมาอย่างรัดกุมมากขึ้น เราอาจจะกล่าวได้ว่า QName เป็นประเภทหนึ่งของ CURIE

6. Document Object Model (DOM)

Document Object Model[18] หรือ DOM โครงสร้างในการแสดงเลย์เอาท์ของส่วนประกอบต่าง ๆ บนเว็บเพจ เก็บในรูปแบบโครงสร้างต้นไม้ในหน่วยความจำ ตัวอย่างเช่น HTML ดังนี้

```
<html>
    <head>
        <title>My title</title>
    </head>
    <body>
        <a href=''>My link</a>
        <h1>My header</h1>
    </body>
</html>
```


เมื่อนำไป parse เข้าไปในเครื่องแล้วจะอยู่ในรูปแบบของ DOM ซึ่งมีลักษณะโครงสร้างต้นไม้ดังภาพที่ 3 โดยจะซ้อนกันเป็น cascade หรือเป็นชั้น ๆ (เราจึงสามารถใช้ Cascading Style Sheets หรือ CSS มาจัดการกำหนดรูปแบบมันได้โดยง่าย)

---

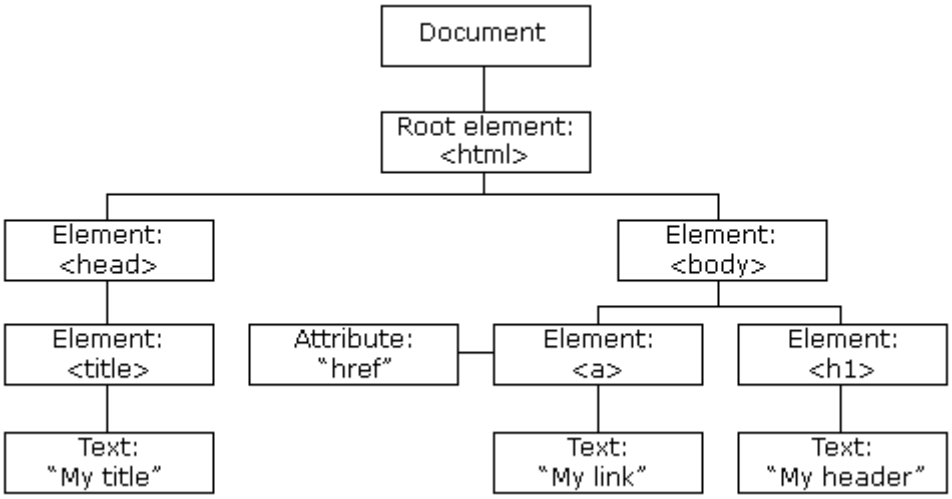
[14] HTML Standard - The XML syntax. WHATWG. Available from: https://html.spec.whatwg.org/#the-xhtml-syntax

[15] Namespaces in XML 1.0. W3C. Available from: https://www.w3.org/TR/xml-names/

[16] Using Qualified Names as Identifiers in XML Content. W3C. Available from: https://www.w3.org/2001/tag/doc/qnameids

[17] CURIE Syntax 1.0. W3C. Available from: https://www.w3.org/TR/curie/

[18] Document Object Model (DOM) Specifications. W3C. Available from: https://www.w3.org/DOM/DOMTR



ภาพที่ 3 ตัวอย่าง Document Object Model[19]

การใช้ DOM ทำให้เราสามารถค้นหา และเปลี่ยนแปลงแก้ไขโครงสร้างข้อมูลและรูปแบบที่แสดงบนหน้าเว็บ โดยผ่านทางภาษาโปรแกรมมิ่ง เช่น JavaScript และด้วยการใช้ event handler ที่ระบุไว้ในส่วนต่าง ๆ ของเว็บเพจ ทำให้เราสามารถสร้างการตอบสนองต่อเหตุการณ์ต่าง ๆ ได้ในแบบ interactive

7. XML Path Language (XPath)

XML Path Language หรือ XPath[20] เป็นภาษาคิวรีที่ใช้ในการเลือกโหนดใน XML หรือใน DOM ตั้งแต่ Level 3 ขึ้นไป โดยมองว่า XML หรือ DOM เป็นโครงสร้างต้นไม้คล้ายกับ directory path ของโครงสร้างระบบไฟล์ และเข้าถึง element ต่าง ๆ ในลักษณะเดียวกันคือใช้เครื่องหมาย / ในการเข้าถึงโหนดในแต่ละชั้น โดยมีเครื่องหมายพิเศษในการเลือกตามเงื่อนไข เช่น เลือกเฉพาะ attributes หรือความยาวตัวอักษร เป็นต้น

ซีแมนเติกเว็บ หรือเว็บเชิงความหมาย

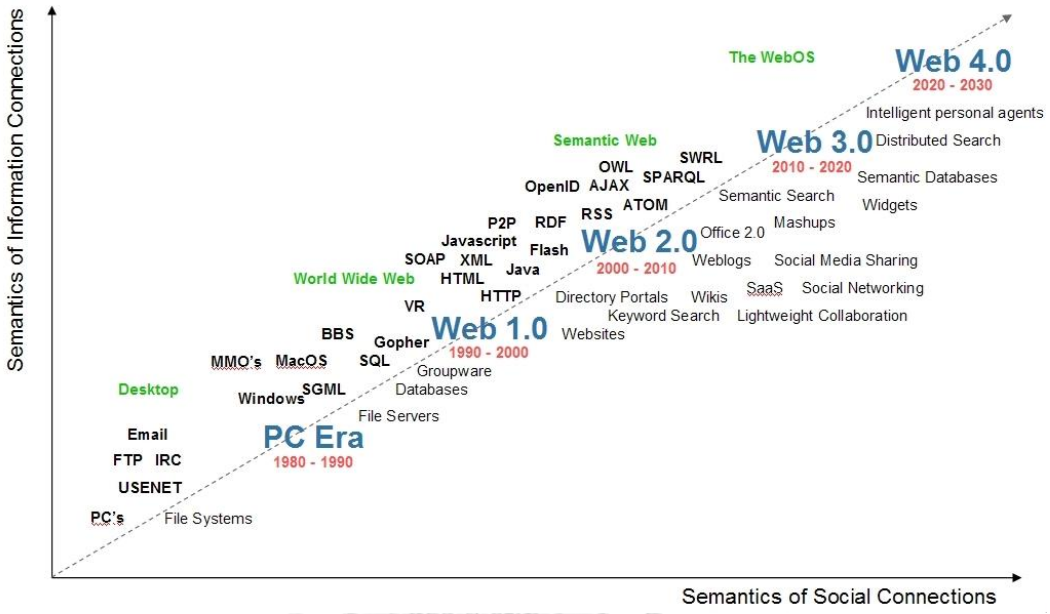
ก่อนจะกล่าวถึงซีแมนติกเว็บ (Semantic Web[21]) ขออธิบายถึงพัฒนาการของเว็บโดยสังเขป Nova Spivack[3] ได้สรุปถึงพัฒนาการที่สำคัญของเว็บดังภาพที่ 4 ช่วงแรกสุดคือเว็บ 1.0 ซึ่งส่วนใหญ่จะเป็นข้อมูลที่สร้างโดยสถาบันการศึกษาวิจัยและองค์กรธุรกิจต่าง ๆ และจะมีการเปลี่ยนแปลงเนื้อหาไม่บ่อยมากนัก หรืออาจจะเรียกว่าเป็นการเชื่อมต่อจากเครื่องคอมพิวเตอร์สู่ผู้ใช้ อันดับถัดมา

[19] JavaScript HTML DOM. W3Schools. Available from: https://www.w3schools.com/js/js_htmldom.asp

[20] XPath. W3C. Available from: https://www.w3.org/TR/xpath/

[21] Semantic Web. W3C. Available from: https://www.w3.org/standards/semanticweb/

คือเว็บ 2.0 จะมีเนื้อหาที่มีการเปลี่ยนแปลงอยู่ตลอดเวลาและเนื้อหาส่วนใหญ่จะมาจากผู้ใช้เว็บเป็นคนสร้างหรือบันทึกข้อมูลเข้าไป เช่น เว็บล็อกหรือบล็อก (blog) วิกิ (wiki) และเครือข่ายสังคม (social networks) เป็นต้น หรืออาจจะเรียกได้ว่าเป็นการเชื่อมต่อระหว่างผู้ใช้กับผู้ใช้ด้วยกันเอง ตอนนี้เราอยู่ในยุคของเว็บ 3.0 และซีแมนติกเว็บ ซึ่งจะมุ่งเน้นไปที่การเชื่อมโยงข้อมูลระหว่างเครื่องคอมพิวเตอร์ต่าง ๆ เข้าด้วยกันและมีการประมวลผลข้อมูลเหล่านั้นโดยคอมพิวเตอร์เองโดยตรง



ภาพที่ 4 พัฒนาการของเว็บเทคโนโลยี[3]

อย่างไรก็ตาม หลายคนไม่เห็นด้วยกับการกำหนดเลขเวอร์ชันให้กับเว็บ รวมถึง เบอร์เนอร์ส-ลี ซึ่งกล่าวว่าเว็บถูกสร้างมาด้วยจุดประสงค์เพื่อเชื่อมต่อผู้ใช้เข้าด้วยกันตั้งแต่แรกเริ่มอยู่แล้ว[4] โดยออกแบบมาเพื่อให้สามารถทำได้ทั้งอ่านและเขียนข้อมูลในขณะใช้งานโดยผู้ใช้ตั้งแต่โปรแกรมเว็บเบราเซอร์ตัวแรก[22] และเว็บ 2.0 เป็นเพียงแค่ jargon หรือ buzzword เท่านั้น ในขณะที่ซีแมนติกเว็บไม่ใช่เว็บเวอร์ชันที่ 3 แต่เป็นเว็บของข้อมูลที่เชื่อมต่อเครื่องคอมพิวเตอร์เข้าด้วยกันในระดับประมวลผลอย่างแท้จริง[23]

1. <u>ซีแมนติกเว็บคืออะไร</u>

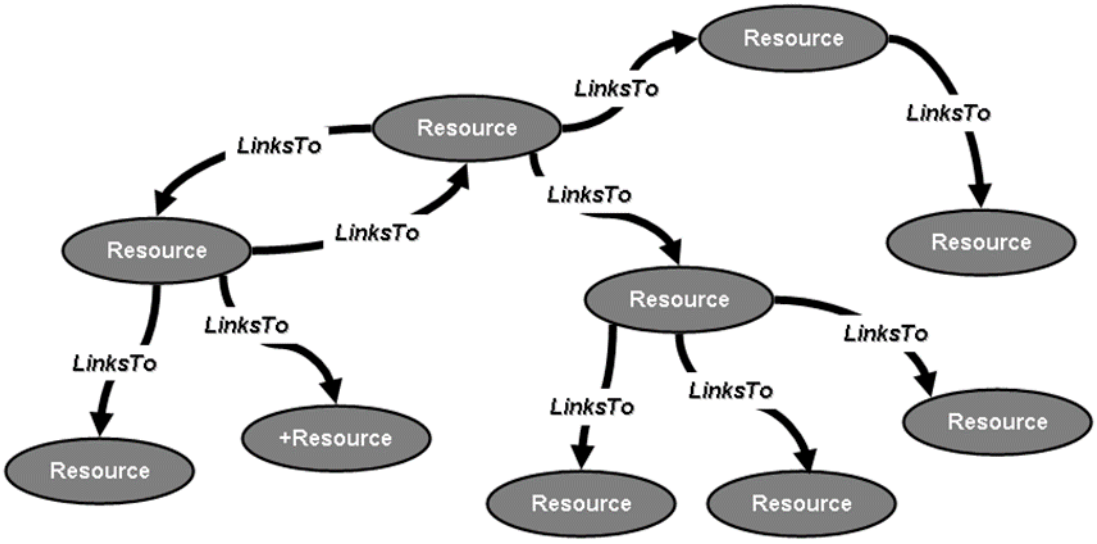
แท้จริงแล้วซีแมนติกเว็บไม่ใช่เทคโนโลยีใหม่ทั้งหมด เบอร์เนอร์ส-ลี เป็นผู้ที่ประกาศแนวคิดของซีแมนติกเว็บขึ้นในปี ค.ศ. 2001 ในบทความของนิตยสารไซเอนติฟิคอเมริกัน[5] ไว้ดังนี้ “ซีแมน

[22] Berners-Lee on the read/write web. BBC NEWS. Available from: http://news.bbc.co.uk/2/hi/technology/4132752.stm

[23] Developer Interviews: Tim Berners-Lee. Available from: https://www.ibm.com/developerworks/podcast/dwi/cm-int082206txt.html

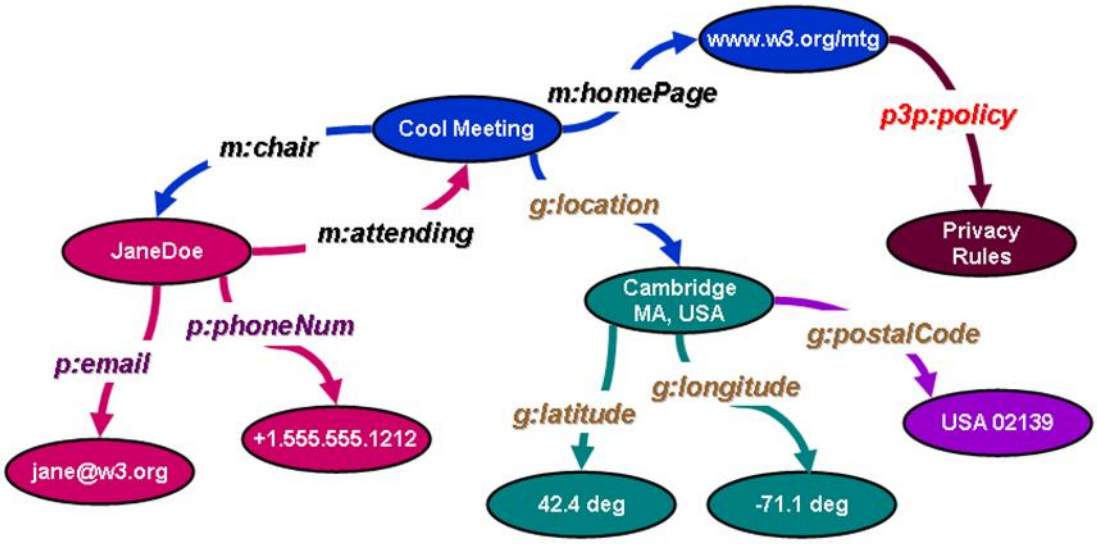
ติกเว็บเป็นส่วนขยายของเว็บที่เราใช้อยู่ในปัจจุบัน โดยข้อมูลสารสนเทศต่าง ๆ จะถูกกำหนดความหมายไว้อย่างเป็นแบบแผน เพื่อที่จะทำให้มนุษย์และคอมพิวเตอร์ทำงานสอดประสานร่วมกันได้โดยง่าย” หรือกล่าวอีกนัยหนึ่งคือ ซีแมนติกเว็บเป็นเว็บของข้อมูล ที่สามารถถูกประมวลผลในทางตรงหรือทางอ้อมได้โดยคอมพิวเตอร์ ตรงนี้เป็นประเด็นสำคัญ คือซีแมนติกเว็บจะไม่ใช่เว็บของมนุษย์อีกต่อไป แต่เป็นเว็บของเครื่องเป็นหลัก ส่วนมนุษย์จะได้ใช้ประโยชน์จากเครื่องอีกต่อหนึ่งได้ในรูปแบบใดนั้นเป็นอีกเรื่องหนึ่ง

Steve Bratt, CEO ของ W3C[6] เปรียบเทียบเวิลด์ไวด์เว็บกับซีแมนติกเว็บดังนี้ ตัวอย่างของเวิลด์ไวด์เว็บเป็นดังภาพที่ 5 ซึ่งมีการใช้ไฮเปอร์ลิงก์ในการเชื่อมโยงรีซอร์สต่าง ๆ ระหว่างเครื่องคอมพิวเตอร์ที่เชื่อมต่อกันด้วยอินเตอร์เน็ต ลิงก์แต่ละลิงก์จะมีทิศทางแต่ไม่มีความหมายอะไรระบุอยู่อย่างชัดเจน เป็นเพียงแค่มีไฮเปอร์เท็กซ์ลิงก์จากเว็บเพจหนึ่งไปอีกเว็บเพจหนึ่งเท่านั้น หรือเราอาจเรียกได้อีกอย่างหนึ่งว่ามันคือไฮเปอร์เท็กซ์เว็บก็ได้



ภาพที่ 5 การเชื่อมโยงระหว่างรีซอร์สโดยใช้ไฮเปอร์ลิงก์ในเวิลด์ไวด์เว็บ[6]

ส่วนตัวอย่างของซีแมนติกเว็บจะเป็นดังภาพที่ 6 คือมีการใส่ “ความหมาย” ลงไปในลิงก์ที่ใช้เชื่อมโยงรีซอร์สหรือ “วัตถุ” ต่าง ๆ เข้าด้วยกันเพื่อบอกว่ามันมีความสัมพันธ์กับรีซอร์สอื่นอย่างไรบ้าง โดยมองว่าแต่ละรีซอร์สเป็นส่วนหนึ่งของฐานข้อมูลแบบกระจายขนาดมหึมาบนอินเตอร์เน็ต ซึ่งสามารถที่จะประมวลผลโดยคอมพิวเตอร์และแสดงผลในหลากหลายรูปแบบตามที่ผู้ใช้ต้องการ แรกเริ่ม เบอร์เนอร์ส-ลี เรียกมันว่า GGG แทนที่จะเป็น WWW โดยย่อมาจาก Giant Global Graph นั่นเอง



ภาพที่ 6 การเชื่อมโยงระหว่างรีซอร์สโดยใช้ความสัมพันธ์ในซีแมนติกเว็บ[6]

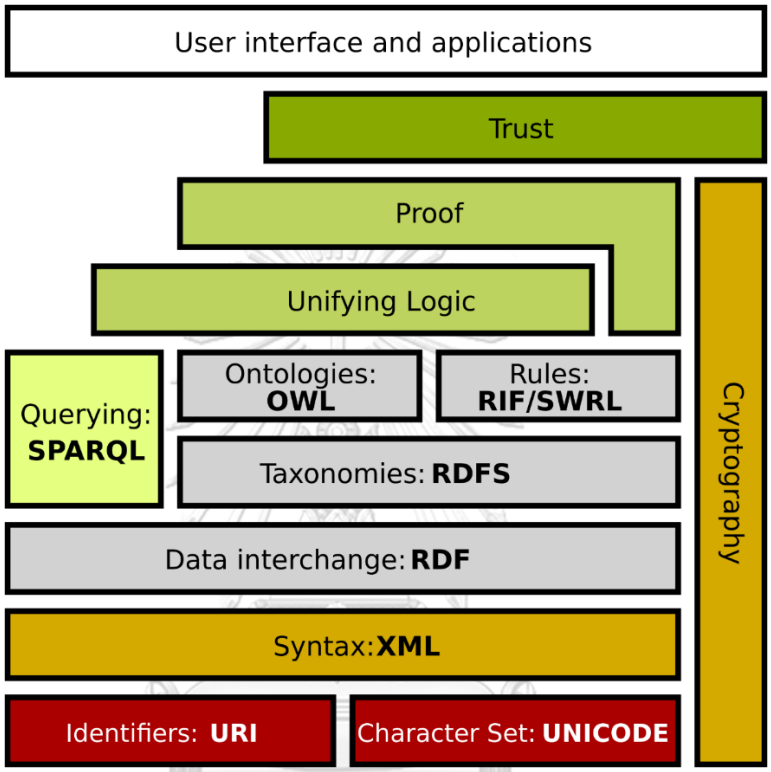
โดยสรุป ซีแมนติกเว็บเป็นเทคโนโลยีที่ซ้อนอยู่บนพื้นฐานของเว็บเทคโนโลยีเดิมและอินเตอร์เน็ต โดยอาศัยชุดของโปรโตคอลในระดับต่าง ๆ มาทำงานร่วมกันเพื่อทำให้สามารถเก็บโครงสร้างข้อมูลที่กระจายอยู่บนเว็บให้อยู่ในรูปแบบของความสัมพันธ์ ซึ่งข้อมูลหรือความสัมพันธ์นั้นสามารถเชื่อมโยงหากันข้ามระบบได้ผ่านเครือข่ายอินเตอร์เน็ต ขอยกตัวอย่างสั้น ๆ ถึงประโยชน์ที่ได้จากซีแมนติกเว็บ คือการค้นหาข้อมูลเกี่ยวกับโปรตีนที่มีผลต่อการรักษาโรคอัลไซม์เมอร์เท่าที่มีการศึกษากันทั่วโลกในปัจจุบัน ถ้าค้นหาโดยใช้เสิร์ชเอ็นจิน (search engine) ปกติอาจได้ผลลัพธ์ถึงมากกว่า 200,000 เอกสารจากเว็บไซต์ทั่วโลก ซึ่งเราไม่สามารถจะแยกแยะหรือค้นคว้าต่อได้หมด แต่หากค้นหาในซีแมนติกเว็บจะได้ผลลัพธ์มาเพียง 20 กว่ารายชื่อของโปรตีนที่มีผลต่อโรคอัลไซม์เมอร์จากซีแมนติกเว็บของนักวิจัยต่าง ๆ ที่ทำการค้นคว้าเรื่องโรคอัลไซม์เมอร์และแบ่งปันแลกเปลี่ยนข้อมูลกันอยู่ในปัจจุบัน[24]

2. องค์ประกอบของซีแมนติกเว็บ

เช่นเดียวกับการให้บริการอื่น ๆ บนอินเตอร์เน็ต ซึ่งส่วนใหญ่จะเป็นการบูรณาการส่วนประกอบที่เป็นมาตรฐานหรือมีใช้กันอยู่แล้วโดยทั่วไป ในกรณีของซีแมนติกเว็บจะประกอบไปด้วยส่วนประกอบต่าง ๆ ที่เป็นมาตรฐานอยู่แล้ว เช่น Uniform Resource Identifier (URI), Extensible Markup Language (XML) และมาตรฐานอื่น ๆ โดยที่เฟรมเวิร์คบางอย่างได้ถูกพัฒนา

[24] The next web. TED Talk. Available from: https://www.ted.com/talks/tim_berners_lee_the_next_web/transcript

ต่อยอดหรือปรับปรุงเปลี่ยนแปลงมาจากสิ่งที่มีอยู่เดิม เช่น Resource Description Framework (RDF), Web Ontology Language (OWL) และ SPARQL Protocol and RDF Query Language (SPARQL) ซึ่งระดับชั้นของส่วนประกอบดังกล่าว หรือเรียกอีกอย่างว่าซีแมนติกเว็บเทคโนโลยีสแต็กเป็นดังภาพที่ 7



ภาพที่ 7 ระดับชั้นของเทคโนโลยีที่ทำงานร่วมกันเป็นซีแมนติกเว็บ[25]

3. Entity–Attribute–Value (EAV) Model

ก่อนจะกล่าวถึง RDF ที่เป็นหัวใจหลักของ Semantic Web ในหัวข้อถัดไป จะขออธิบายถึงแนวคิดพื้นฐานของ RDF ที่เรียกว่า EAV Model พอสังเขป โดยจะยกตัวอย่างข้อมูลบางส่วนจากหนังสือแนะนำท่องเที่ยวไทยเล่มหนึ่ง

หากเราจะออกแบบตารางเพื่อเก็บข้อมูลของร้านอาหารแบบง่าย ๆ โดยมีรายละเอียดคือ ชื่อร้าน, ที่อยู่, ประเภทอาหาร, ช่วงราคา และรายการอาหารแนะนำ ก็จะได้ตารางดังภาพที่ 8 คือเป็นตารางในลักษณะ flat table หรือเป็นแบบแบบราบ ที่อยู่ในรูปแบบ unnormalized ซึ่งในฟิลด์สุดท้ายที่เป็นอาหารแนะนำสามารถรองรับได้แค่หนึ่งรายการเท่านั้น หรือเราอาจจะประยุกต์ให้ฟิลด์

[25] Semantic Web. Wikipedia. Available from: https://en.wikipedia.org/wiki/Semantic_Web

รายการอาหารแนะนำสามารถรองรับได้หลายรายการก็อาจจะใช้วิธีง่าย ๆ คือ แบ่งแต่ละรายการอาหารด้วยจุลภาค หรือเครื่องหมาย comma ก็จะได้ตัวอย่างดังภาพที่ 9 ซึ่งการรวมข้อมูลหลายรายการไว้ในฟิลด์เดียวกันหรือเรียกว่า repeating group[26] จะมีข้อจำกัดหลายประการ ยกตัวอย่าง เช่น การปรับปรุงแก้ไขเพิ่มลดรายการ การค้นหา การเรียงลำดับ เป็นต้น

Restaurant Name	Address	Cuisine	Price	Highlight
Maela Plaphao	Singburi	Thai	$	Fried Fish
Jae Ngek	Sakaeo	Vietnamese	$$	Nem Nuong
Sweet Home	Nakhonphanom	American	$$$	Steak
Zapp	Yasothon	Thai	$$	Papaya Salad

ภาพที่ 8 Tabular Data ในลักษณะ Flat Table

Restaurant Name	Address	Cuisine	Price	Highlights
Maela Plaphao	Singburi	Thai	$	Fried Fish, Steamed Shrimp
Jae Ngek	Sakaeo	Vietnamese	$$	Nem Nuong, Spring Roll, Fried Chicken
Sweet Home	Nakhonphanom	American	$$$	Steak, Salad, Sandwich, Hamburger
Zapp	Yasothon	Thai	$$	Papaya Salad, Larb, Nam Tok

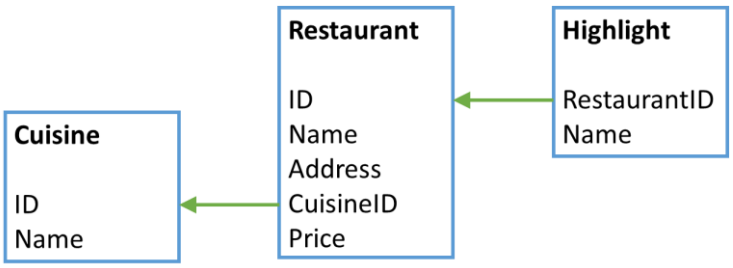
ภาพที่ 9 Tabular Data ตัวอย่างการบันทึกรายการอาหารแนะนำมากกว่าหนึ่งรายการ

จากข้อจำกัดดังกล่าว เราอาจทำการกระจายตารางหรือ normalization[27] ออกมาเป็นตารางย่อยที่ประกอบกันเป็นฐานข้อมูลเชิงสัมพันธ์หรือ relational database[7] ก็จะได้แผนภูมิโครงสร้างดังภาพที่ 10 โดยมีข้อมูลตัวอย่างในแต่ละตารางดังภาพที่ 11 ซึ่งการแยกตารางที่ใช้เก็บประเภทอาหารและรายการอาหารแนะนำออกมาไว้ในตารางต่างหากในรูปแบบ third normal form[28] จะทำให้เราสามารถจัดการข้อมูลในตารางทั้งคู่ได้อย่างสะดวกและมีประสิทธิภาพมากขึ้น รวมไปถึงลดการซ้ำซ้อนของข้อมูล ซึ่งโดยรวมทั้งหมดถือเป็นข้อเด่นของระบบฐานข้อมูลเชิงสัมพันธ์

[26] Designs that violate 1NF. Available from: https://en.wikipedia.org/wiki/First_normal_form#Designs_that_violate_1NF

[27] Database normalization. Wikipedia. Available from: https://en.wikipedia.org/wiki/Database_normalization

[28] Third normal form. Wikipedia. Available form: https://en.wikipedia.org/wiki/Third_normal_form



ภาพที่ 10 แผนภูมิแสดงความสัมพันธ์ของแต่ละตารางในฐานข้อมูลตัวอย่าง

Restaurant

ID	Name	Address	CuisID	Price
1	Maela Plaphao	Singburi	1	$
2	Jae Ngek	Sakaeo	2	$$
3	Sweet Home	Nakhonphanom	3	$$$
4	Zapp	Yasothon	1	$$

Cuisine

ID	Name
1	Thai
2	Vietnamese
3	American

Highlight

RestID	Name
1	Fried Fish
1	Steamed Shrimp
2	Nem Nuong
2	Spring Roll
2	Fried Chicken
3	Steak
3	Salad
3	Sandwich
3	Hamburger
4	Papaya Salad
4	Larb
4	Nam Tok

ภาพที่ 11 ตัวอย่างข้อมูลในแต่ละตารางของฐานข้อมูลเชิงสัมพันธ์

อย่างไรก็ตามการออกแบบโครงสร้างฐานข้อมูลเชิงสัมพันธ์ต้องทำการวิเคราะห์ความต้องการในการเก็บข้อมูลให้ครบถ้วนเพียงพอตั้งแต่ก่อนการออกแบบ ไม่เช่นนั้นแล้วการปรับปรุงแก้ไขโครงสร้างฐานข้อมูลจะไม่สามารถทำได้โดยง่าย ยกตัวอย่างเช่น หากเราต้องการให้ฐานข้อมูลเชิงสัมพันธ์ของเราสามารถบันทึกข้อมูลร้านขายของที่ระลึกเพิ่มเติมนอกเหนือไปจากร้านอาหาร โดยที่จำเป็นต้องรองรับในกรณีที่ร้านเดียวกันสามารถที่จะขายได้ทั้งอาหารและของที่ระลึก ตัวอย่างรายละเอียดเกี่ยวกับข้อมูลร้านขายของที่ระลึกเป็นดังภาพที่ 12 ซึ่งประกอบไปด้วย หมายเลขไอดีประจำรายการ, ชื่อร้าน, ที่อยู่, สินค้าอุปโภคหรือบริโภค และประเภทของที่ระลึก หากเราจะนำตารางนี้ไปเพิ่มเข้าไปในโครงสร้างตรง ๆ ก็จะต้องสร้าง link table (หรือ map table)[29] เพื่อใช้เชื่อมโยงข้อมูลตารางหลักสองตารางนี้เข้าด้วยกันดังภาพที่ 13

[29] Associative entity. Wikipedia. Available from: https://en.wikipedia.org/wiki/Associative_entity

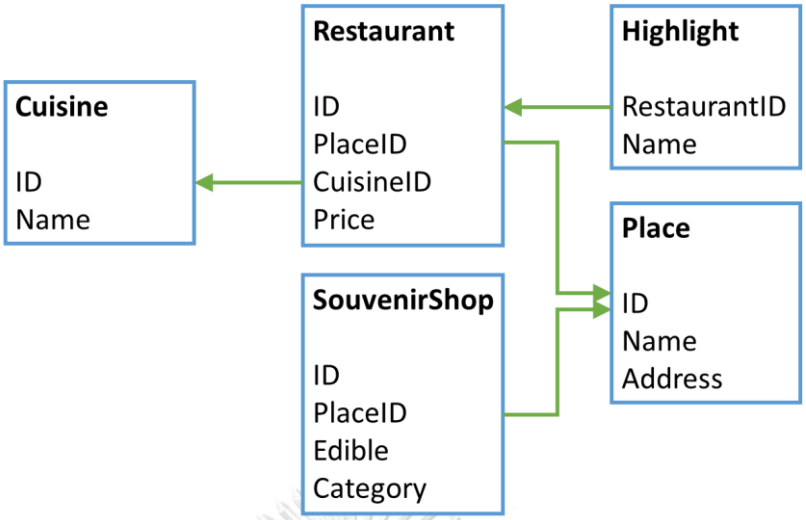
SouvenirShop				
ID	**Name**	**Address**	**Edible**	**Category**
1	Ban Bencharong	Samutsongkhram	No	Decoration
2	Khanombankhunkaew	Nakhonpathom	Yes	Confectionery
3	Sweet Home	Nakhonphanom	No	Jewelry

ภาพที่ 12 ตัวอย่างรายละเอียดเกี่ยวกับข้อมูลร้านขายของที่ระลึก

ภาพที่ 13 การสร้าง map table เพื่อใช้เชื่อมตารางหลักสองตารางเข้าด้วยกัน

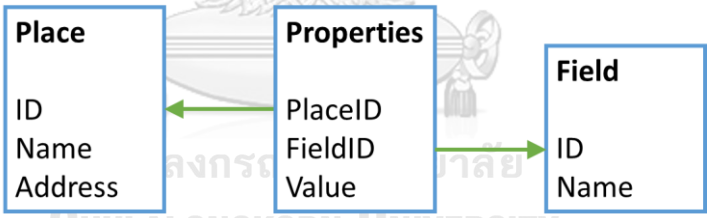
จะเห็นได้ว่าวิธีดังกล่าวเป็นการแก้ปัญหาแบบ ad-hoc หรือการที่ไม่ได้ออกแบบมาเตรียมรองรับไว้ล่วงหน้า ซึ่งก็จะทำให้ฐานข้อมูลเชิงสัมพันธ์เกิดความซ้ำซ้อนและใช้งานได้อย่างไม่มีประสิทธิภาพ ดังนั้นสิ่งที่ควรจะต้องทำคือเราต้องทำการ schema refactoring[30] โดยการออกแบบปรับปรุงโครงสร้างฐานข้อมูลใหม่เพื่อให้รองรับข้อมูลในรูปแบบที่เพิ่มเติมเข้ามา โดยเราอาจจะปรับปรุงแก้ไขโครงสร้างฐานข้อมูลได้ดังภาพที่ 14 โดยที่เราทำการแยกฟิลด์ข้อมูลที่ซ้ำซ้อนกันในตารางหลักสองตารางแล้วนำมาสร้างเป็นตารางใหม่ที่มีรายระเอียดของข้อมูลเป็น ชื่อร้าน และที่อยู่ โดยเชื่อมตารางหลักทั้งสองเข้ามาที่ตารางใหม่นี้เพื่อสร้างความสัมพันธ์ที่ครบถ้วนตามความต้องการ ซึ่งท้ายที่สุดแล้วในระบบที่ใหญ่ขึ้น ฐานข้อมูลเชิงสัมพันธ์ก็จะมีความซับซ้อนมากขึ้นตามไปด้วยอย่างหลีกเลี่ยงไม่ได้

[30] Database refactoring. Wikipedia. Available from: https://en.wikipedia.org/wiki/Database_refactoring



ภาพที่ 14 แผนภูมิโครงสร้างฐานข้อมูลหลังจากที่ได้ทำการ refactoring schema

ทางแก้ปัญหาอีกลักษณะหนึ่งคือการสร้าง properties table เพื่อใช้เก็บ customized field มาเพิ่มความยืดหยุ่นในการเก็บข้อมูล ซึ่งถือว่าเป็นการ denormalization[31] รูปแบบหนึ่ง จะลองนำตัวอย่างตารางในภาพที่ 14 มาทำการ denormalization โดยการสร้าง properties table เพื่อใช้เชื่อมระหว่างตารางหลักคือ Place กับตาราง customized field จะได้โครงสร้างดังภาพที่ 15 และตัวอย่างข้อมูลหลังจากที่ได้ปรับโครงสร้างแล้วจะเป็นดังภาพที่ 16



ภาพที่ 15 การใช้ properties table ในการเก็บ customized field

[31] Denormalization. Wikipedia. Available from: https://en.wikipedia.org/wiki/Denormalization

Place

ID	Name	Address
1	Maela Plaphao	Singburi
2	Jae Ngek	Sakaeo
3	Sweet Home	Nakhonphanom
4	Zapp	Yasothon
5	Ban Bencharong	Samutsongkhram

Properties

PlaceID	FieldID	Value
1	1	Thai
1	2	$
2	1	Vietnamese
2	2	$$
3	1	American
3	2	$$$
3	3	No
3	4	Jewelry

Field

ID	Name
1	Cuisine
2	Price
3	Edible
4	Category

ภาพที่ 16 ตัวอย่างข้อมูลที่ปรับโครงสร้างมาใช้ properties table และ customized field

จะเห็นได้ว่าโครงสร้างที่ได้จะถูกลดทอนความหมายหรือ semantic เชิง schema ในรูปแบบโครงสร้างความสัมพันธ์ลงโดยนำเอาความหมายของข้อมูลออกจากชื่อตารางและลดระดับลงไปเก็บไว้แค่เพียงระดับฟิลด์ แต่โครงสร้างที่ได้จะมีความยืดหยุ่นมากขึ้น เช่น หากเราต้องการเพิ่มการเก็บข้อมูลว่าร้านไหนมีการเล่นดนตรีสดหรือไม่ และเล่นดนตรีในแนวไหน หากใช้วิธีการเพิ่มตารางหรือเพิ่มฟิลด์ดังตัวอย่างที่ผ่านมา โดยต้องทำการ refactoring ก็จะยิ่งเพิ่มความซับซ้อนของโครงสร้าง ดังนั้นหากเราย้าย relation หรือ schema มาเก็บไว้ที่ตัวข้อมูลการเพิ่มความสัมพันธ์ใหม่ก็เพียงแค่เพิ่มข้อมูลของความสัมพันธ์นั้นเข้าไปในตารางโดยที่ไม่จำเป็นต้องแก้ไขโครงสร้างข้อมูลเลย ดังตัวอย่างในภาพที่ 17

Place

ID	Name	Address
1	Maela Plaphao	Singburi
2	Jae Ngek	Sakaeo
3	Sweet Home	Nakhonphanom
4	Zapp	Yasothon
5	Ban Bencharong	Samutsongkhram

Properties

PlaceID	FieldID	Value
1	1	Thai
1	2	$
2	1	Vietnamese
2	2	$$
3	1	American
3	2	$$$
3	3	No
3	4	Jewelry
3	5	Yes
3	6	Country

Field

ID	Name
1	Cuisine
2	Price
3	Edible
4	Category
5	LiveMusic
6	MusicGenre

ภาพที่ 17 เมื่อเพิ่มฟิลด์เพื่อใช้เก็บข้อมูลเพิ่มเติมเกี่ยวกับร้านที่มีการเล่นดนตรีสด

และหากพิจารณาต่อไปจะเห็นได้ว่าตารางหลัก Place ที่ใช้เก็บชื่อร้านค้าที่เป็นคีย์หลักของฐานข้อมูลก็ยังสามารถลดทอนลงไปเป็นส่วนหนึ่งของ customized filed ได้อีกด้วย ดังในภาพที่ 18

Properties		
PlaceID	FieldID	Value
1	1	Thai
1	2	$
1	7	Maela Plaphao
1	8	Singburi
2	1	Vietnamese
2	2	$$
2	7	Jae Ngek
2	8	Sakaeo
3	1	American
3	2	$$$
3	3	No
3	4	Jewelry
3	5	Yes
3	6	Country
3	7	Sweet Home
3	8	Nakhonphanom

Field	
ID	Name
1	Cuisine
2	Price
3	Edible
4	Category
5	LiveMusic
6	MusicGenre
7	Name
8	Address

ภาพที่ 18 เมื่อนำตารางหลักมาเก็บไว้เป็น customized field ด้วย

เมื่อมาถึงขั้นนี้จะเห็นได้ว่าไม่มีความจำเป็นที่เราต้องแยกตาราง customized field ออกมาต่างหาก เราสามารถนำข้อมูล Field Name ไปแทนที่ FieldID ในตาราง properties table ได้เลย ผลลัพธ์ที่ได้จะเป็นดังภาพที่ 19

Properties		
PlaceID	Field	Value
1	Cuisine	Thai
1	Price	$
1	Name	Maela Plaphao
1	Address	Singburi
2	Cuisine	Vietnamese
2	Price	$$
2	Name	Jae Ngek
2	Address	Sakaeo
3	Cuisine	American
3	Price	$$$
3	Edible	No
3	Category	Jewelry
3	LiveMusic	Yes
3	MusicGenre	Country
3	Name	Sweet Home
3	Address	Nakhonphanom

ภาพที่ 19 เมื่อนำชื่อฟิลด์ในตาราง customized field มารวมไว้ใน table เดียวกัน

ซึ่งโมเดลโครงสร้างฐานข้อมูลที่มีตารางเดียวที่ได้ในขั้นสุดท้ายนี้ เราอาจเรียกอีกอย่างว่า Entity–Attribute–Value (EAV) model[32] หรือในบางกรณีจะถูกเรียกว่า open schema หรือ vertical database model นั่นเอง โดยในแต่ละ record หรือ row ของตารางจะอยู่ในลักษณะของ attribute–value pair[33] หรือ key–value pair ซึ่งในหัวข้อถัดไปที่จะกล่าวถึง RDF หรือหน่วยย่อยที่สุดของโครงสร้างที่ใช้เก็บข้อมูลของซีแมนติกเว็บ ก็จะถูกเก็บไว้โดยใช้โมเดลโครงสร้างลักษณะนี้นั่นเอง

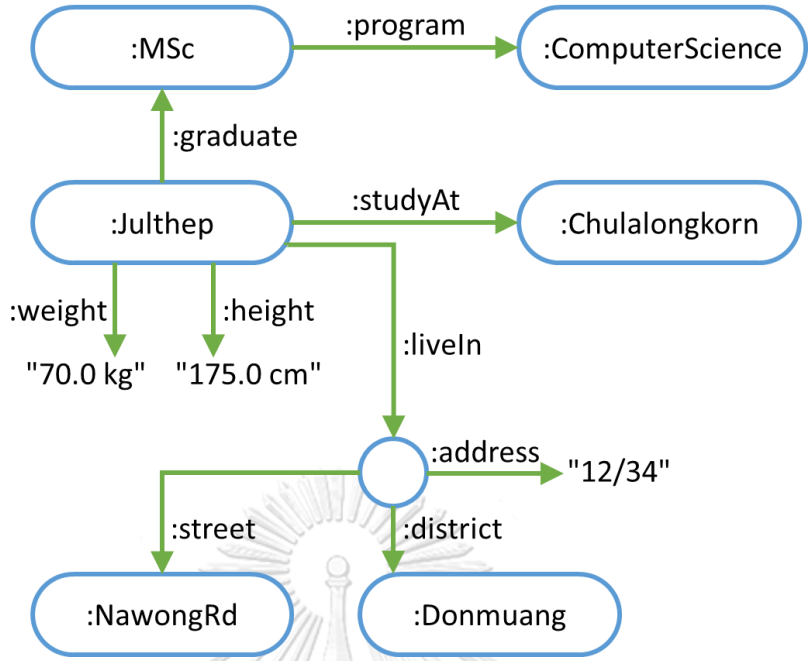
4. RDF (Resource Description Framework)

Resource Description Framework[34] เป็นหัวใจหลักในการเก็บชิ้นส่วนย่อยที่สุดของข้อเท็จจริงต่าง ๆ ในฐานความรู้ที่เชื่อมโยงโดยซีแมนติกเว็บ เปรียบเสมือนเป็นอะตอมของซีแมนติกเว็บ โดยพื้นฐานที่สุดแล้ว RDF คือ “ประโยค” ในรูปแบบทริปเปิล (triple) ที่มีข้อมูลอยู่สามส่วนคือ (ประธาน, กริยา, กรรม) หรือ (subject, predicate, object) แสดงความสัมพันธ์จากซับเจ็กต์หนึ่งไปที่อ็อบเจ็กต์หนึ่งด้วยเพรดิเคตหนึ่ง ซึ่งทั้งซับเจ็กต์และอ็อบเจ็กต์จะเป็นชื่อของรีซอร์สที่อยู่ในรูปของ URI (ในกรณีของอ็อบเจ็กต์สามารถเป็นค่าคงที่หรือ literal ได้) และเพรดิเคตจะแสดงถึง property หรือ attribute ที่เชื่อมความสัมพันธ์ระหว่างซับเจ็กต์และอ็อบเจ็กต์โดยอยู่ในรูปแบบ URI เช่นเดียวกัน เมื่อแต่ละทริปเปิลถูกเชื่อมโยงเข้าด้วยกันจะเป็นโครงสร้างกราฟ RDF ซึ่งอาจจะเรียกว่ากราฟความหมาย (semantic graph) หรือกราฟความรู้ (knowledge graph)

[32] Entity–attribute–value model. Wikipedia. Available from: https://en.wikipedia.org/wiki/Entity–attribute–value_model

[33] Attribute–value pair. Wikipedia. Available from: https://en.wikipedia.org/wiki/Attribute–value_pair

[34] RDF 1.1 Concepts and Abstract Syntax. W3C. Available from: https://www.w3.org/TR/rdf11-concepts/



ภาพที่ 20 ตัวอย่างซีแมนติกกราฟที่อธิบายถึงนิสิตคนหนึ่งของจุฬาลงกรณ์มหาวิทยาลัย

RDF ใช้แสดงถึงความสัมพันธ์ระหว่างข้อมูลในซีแมนติกเว็บ และบอกถึงความหมายของความสัมพันธ์นั้น ๆ จะขอยกตัวอย่างโครงสร้างความสัมพันธ์ขนาดเล็กที่ประกอบด้วย entity เพียงไม่กี่ตัวที่เกี่ยวกับนิสิตคนหนึ่งของจุฬาลงกรณ์มหาวิทยาลัยในรูปของซีแมนติกกราฟ แสดงดังภาพที่ 20 ซึ่งหากนำมาเขียนให้อยู่ของ RDF ทริปเปิลก็จะได้ดังนี้

(:Julthep, :graduate, :MSc)

(:Julthep, :studyAt, :Chulalongkorn)

(:Julthep, :weight, "70.0 kg")

(:Julthep, :height, "175.0 cm")

(:Julthep, :liveIn, _)

(:MSc, :program, :ComputerScience)

(_, :address, "12/34")

(_, :street, :NawongRd)

(_, :district, :Donmuang)

จะเห็นได้ว่าจำนวน RDF ทริปเปิลจะมีจำนวนเท่ากับความสัมพันธ์หรือ edges ที่อยู่ในกราฟ ซึ่งต่อไปในเล่มนี้จะใช้รูปแบบมาตรฐาน Turtle ซึ่งเป็นซับเซ็ตของ Notation3 ในการเขียนทริปเปิล (รายละเอียดจะกล่าวในหัวข้อถัดไป) โดยจะได้ RDF ทริปเปิลดังนี้

```
@prefix : <http://example.org/sample#> .

:Julthep  :graduate :MSc .
:Julthep  :studyAt  :Chulalongkorn .
:Julthep  :weight   "70.0 kg" .
:Julthep  :height   "175.0 cm" .
:Julthep  :liveIn   _:xyz7890 .
:MSc      :program  :ComputerScience .
_:xyz7890 :address  "12/34" .
_:xyz7890 :street   :NawongRd .
_:xyz7890 :district :Donmuang .
```

4.1 การรองรับความต้องการที่มากกว่า binary predicate

ดังที่กล่าวมาแล้ว RDF แต่ละหน่วยมีลักษณะเป็นทริปเปิล ที่ประกอบด้วย ซับเจ็กต์, เพรดิเคต และ อ็อบเจ็กต์ ยกตัวอย่างเช่น ทริปเปิล (x, P, y) หากนำมาเขียนในรูปแบบ first-order logic หรือ predicate logic แบบในภาษา Prolog จะเป็นดังนี้ P(x, y) มีความหมายว่า เพรดิเคต P เชื่อมความสัมพันธ์ระหว่าง ซับเจ็กต์ x กับ อ็อบเจ็กต์ y จากที่แสดงให้เห็นนี้จะพบว่า RDF รองรับเพียงแค่ binary predicate หรือเรียกว่า valence 2 หากจะแปลงเพรดิเคตที่มี valence มากกว่า 2 เช่น P(x, y, z) มาเป็น RDF จำเป็นต้องประยุกต์โดยใช้หลายทริปเปิลมาประกอบกัน ขอยกตัวอย่างเพรดิเคตที่มี valence 4 เช่น broker(สมปอง, บ้านเลขที่ 5, สมชาย, สมหญิง) มีความหมายว่า “สมปองเป็นนายหน้าขายบ้านเลขที่ 5 ของสมชายให้กับสมหญิง” หากนำมาเขียนใหม่ให้เป็นเพียงแค่ valence 2 จะได้เป็น 4 เพรดิเคต เช่น

broker(การขายบ้านครั้งที่ 23, สมปอง)

asset(การขายบ้านครั้งที่ 23, บ้านเลขที่ 5)

seller(การขายบ้านครั้งที่ 23, สมชาย)

buyer(การขายบ้านครั้งที่ 23, สมหญิง)

หรือหากนำมาเขียนในในรูปแบบ RDF ทริปเปิลจะได้ดังนี้

```
:HouseSell23 :broker :Sompong .
:HouseSell23 :asset  :HouseNo5 .
:HouseSell23 :seller :Somchai .
:HouseSell23 :buyer  :Somying .
```

4.2 Blank node หรือ anonymous resource

จากกรณีดังกล่าว RDF จึงมีคุณสมบัติพิเศษที่สำคัญมากอย่างหนึ่งที่เรียกว่า blank node เพื่อเป็นโหนดเชื่อมระหว่างซับเจ็กต์ที่มีหลายอ็อบเจ็กต์ และรวมกลุ่มเข้าไว้ด้วยกันเป็นชุดโดยที่เราจะไม่แบ่งแยก ยกตัวอย่างเช่น

```
:Julthep  :liveIn   :Address1 .
:Address1 :address  "12/34" .
:Address1 :street   :NawongRd .
:Address1 :district :Donmuang .
:Address1 :province :Bangkok .
:Address1 :country  :Thailand .
:Address1 :postCode "10210" .
```

โดยเราจะไม่เชื่อม :address "12/34" หรือ :street :NawongRd ไปที่ :Julthep ตรง ๆ เพราะหากเกิดกรณีที่เรามีหลายที่อยู่เชื่อมไปที่รีซอร์สตัวเดียวกัน ข้อมูลอ็อบเจ็กต์ก็จะสับสนปะปนกันไปหมด ดังนั้นเราจึงสร้าง :Address1 มาเป็นโหนดที่ใช้ในการเชื่อมเพิ่มขึ้นมาหนึ่งขั้น ซึ่งในความเป็นจริงเราไม่ได้ความสำคัญกับ :Address1 เราเพียงต้องการเก็บข้อมูลว่า :Julthep :liveIn ที่ :address คือบ้านเลขที่ 12/34, :street คือ :NawongRd ไปจนถึงรหัสไปรษณีย์ :postCode คือ 10210 เท่านั้นที่เป็นสาระสำคัญ ดังนั้น :Address1 จึงสามารถกำหนดให้เป็น anonymous หรือ blank node ได้ ซึ่งระบบจะสร้าง resource ID ขึ้นมาแบบ arbitrary หรือสุ่มขึ้นมาเป็น temporary URI ที่ไม่ซ้ำกับรีซอร์สใด ๆ ในเดต้าเซ็ต และรีซอร์สโหนดนี้จะไม่สามารถอ้างถึงได้จากภายนอกเดต้าเซ็ตมาที่ตัวมันเอง จำเป็นต้องเข้าถึงผ่าน URI ของรีซอร์สอื่นที่เกี่ยวข้องเท่านั้น ซึ่งถ้าหากมีการถ่ายโอนข้อมูล RDF ชุดนี้ข้ามระบบ รีซอร์สไอดีของ blank node ก็จะถูกสุ่มขึ้นมาใหม่เพื่อไม่ให้ซ้ำกับไอดีที่อาจจะมีอยู่แล้วในฐานข้อมูลเดิมของระบบปลางทาง แต่อย่างที่กล่าวมาข้างต้น เราจะไม่สนใจที่ชื่อไอดีเหล่านั้น เราจะสนใจเพียงแค่ความสัมพันธ์ระหว่างไอดีนั้นกับอ็อบเจ็กต์ที่เชื่อมอยู่ด้วยกันทั้งชุด

สำหรับรูปแบบมาตรฐาน Turtle เราจะสามารถสร้าง blank node ได้ 2 วิธี วิธีแรกคือกำหนดชื่อชั่วคราวขึ้นมาเป็น blank node เช่น

```
:Julthep  :liveIn   _:xyz7890 .
_:xyz7890 :address  "12/34" .
_:xyz7890 :street   :NawongRd .
_:xyz7890 :district :Donmuang .
_:xyz7890 :province :Bangkok .
_:xyz7890 :country  :Thailand .
_:xyz7890 :postCode "10210" .
```

หรือวิธีที่สองคือใช้เครื่องหมาย [และ] ในการสร้าง blank node เช่น

```
:Julthep :liveIn [
      :address  "12/34" ;
      :street   :NawongRd ;
      :district :Donmuang ;
      :province :Bangkok ;
      :country  :Thailand ;
      :postCode "10210" ]
```

แต่ไม่ว่าจะเขียนแสดง RDF ทริปเปิลโดยใช้วิธีใด ข้อมูลที่อยู่ในระบบจะเก็บไว้ในโครงสร้างแบบเดียวกันดังตัวอย่างใน RDF กราฟตามภาพที่ 20 ซึ่งโดยมาตรฐานจะใช้สัญลักษณ์วงกลมที่ไม่มี label เป็นการแสดงถึง blank node ในการแสดง RDF กราฟ

4.3 Triplestore

Triplestore หรือเรียกอีกอย่างว่า RDF Store คือระบบฐานข้อมูลที่ออกแบบมาเพื่อใช้เก็บชุดข้อมูลในรูปแบบ RDF ทริปเปิลโดยเฉพาะ ซึ่งโดยส่วนใหญ่ triplestore จะใช้ภาษาคิวรีที่เรียกว่า SPARQL ในการสืบค้นข้อมูล ซึ่งจะกล่าวถึงในหัวข้อถัดไป

5. Serialization Format/Syntax

แท้จริงแล้ว RDF เป็นโครงสร้างของทริปเปิลที่เป็น abstract model ในลักษณะของ directed graph ซึ่งหากอยู่ในหน่วยความจำสามารถอิมพลิเมนท์ได้หลากหลายวิธีขึ้นอยู่กับไลบรารีต่าง ๆ แต่หากจะนำ RDF กราฟนั้นมาแปลงให้อยู่ในรูปแบบข้อมูลเพื่อบันทึกเก็บไว้หรือส่งต่อ (เรียกว่า serialization หรือ marshalling) จะสามารถทำได้ในหลากหลายรูปแบบดังต่อไปนี้

5.1 RDF/XML

RDF/XML เป็นรูปแบบมาตรฐานแรกเริ่มในการ serialize RDF ที่ค่อนข้างจะเทอะทะ เนื่องจากเป็นการนำเอา RDF มาเขียนด้วย XML ดังนั้น syntax ที่ได้จะมีแท็กครอบหน้าและหลังเป็นจำนวนมาก ยกตัวอย่างเช่น กราฟในภาพที่ 20 เมื่อนำมาเขียนในรูปแบบ RDF/XML จะได้ออกมาดังนี้

```
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns="http://example.org/sample#">
  <rdf:Description rdf:about="http://example.org/sample#Julthep">
    <graduate>
      <rdf:Description rdf:about="http://example.org/sample#MSc">
        <program rdf:resource=
                      "http://example.org/sample#ComputerScience"/>
      </rdf:Description>
    </graduate>
    <studyAt rdf:resource="http://example.org/sample#Chulalongkorn"/>
```

```
    <height>175.0 cm</height>
    <weight>70.0 kg</weight>
    <liveIn>
      <rdf:Description>
        <address>12/34</address>
        <street rdf:resource="http://example.org/sample#NawongRd"/>
        <district rdf:resource="http://example.org/sample#Donmuang"/>
      </rdf:Description>
    </liveIn>
  </rdf:Description>
</rdf:RDF>
```

5.2 N-Triples

N-Triples[35] เป็นรูปแบบมาตรฐานที่เรียบง่ายที่สุดในการ serialize RDF โดย N-Triples มีหลักการคือนำส่วนประกอบทั้งสามส่วนของทริปเปิลมาเขียนเรียงกันโดยคั่นด้วย whitespace และปิดท้ายแต่ละทริปเปิลด้วยมหัพภาค (.) แล้วขึ้นบรรทัดใหม่ เนื่องจากไม่รองรับการใช้ namespace ดังนั้น URI ของแต่ละส่วนของทริปเปิลจะถูกเขียนแบบเต็มโดยให้อยู่ในเครื่องหมาย < และ >

การกำหนด blank node ใน N-Triples สามารถทำได้โดยวิธีการตั้งชื่อชั่วคราวขึ้นมาเพื่อแทน blank node โดยกำหนด XML namespace prefix ด้วยเครื่องหมาย _ (underscore หรือเครื่องหมายขีดเส้นใต้) ยกตัวอย่างเช่น กราฟในภาพที่ 20 เมื่อนำมาเขียนด้วย N-Triples จะได้ออกมาดังนี้

```
<http://example.org/sample#Julthep>
            <http://example.org/sample#graduate>
                        <http://example.org/sample#MSc> .
<http://example.org/sample#Julthep>
            <http://example.org/sample#studyAt>
                        <http://example.org/sample#Chulalongkorn> .
<http://example.org/sample#Julthep>
            <http://example.org/sample#weight> "70.0 kg" .
<http://example.org/sample#Julthep>
            <http://example.org/sample#height> "175.0 cm" .
<http://example.org/sample#Julthep>
            <http://example.org/sample#liveIn> _:xyz7890 .
<http://example.org/sample#MSc>
            <http://example.org/sample#program>
                        <http://example.org/sample#ComputerScience> .
_:xyz7890 <http://example.org/sample#address> "12/34" .
_:xyz7890 <http://example.org/sample#street>
                        <http://example.org/sample#NawongRd> .
_:xyz7890 <http://example.org/sample#district>
                        <http://example.org/sample#Donmuang> .
```

[35] RDF 1.1 N-Triples. W3C. Available from: https://www.w3.org/TR/n-triples/

5.3 Notation3 (N3) และ Turtle (TRTL – Terse RDF Triple Language)

Notation3 หรือ N3[8] เป็นรูปแบบที่เป็นซุปเปอร์เซ็ตของ N-Triples ออกแบบมาเพื่อให้ใช้งานสะดวกกว่า RDF/XML ในขณะเดียวกันก็มีคุณสมบัติพิเศษเพิ่มเติมอีกหลายอย่างที่นอกเหนือไปจากแค่การรองรับ RDF ส่วน Terse RDF Triple Language – TRTL หรือ Turtle[36] เป็น syntax ที่ตัดทอนมาจาก N3 เพื่อให้เรียบง่ายและสะดวกในการใช้งานมากขึ้น ซึ่งนับว่าเป็น syntax ที่นิยมใช้กันมากที่สุดรูปแบบหนึ่งในการ serialize RDF

นอกจากเครื่องหมายมหัพภาค (.) ที่เป็นการระบุจุดสิ้นสุดของแต่ละทริปเปิลตาม N-Triples แล้ว ยังมีเครื่องหมายอัฒภาค (;) ที่เป็นการระบุว่าทริปเปิลถัดไปมีซับเจ็กต์เป็นตัวเดียวกัน ดังนั้นจึงสามารถที่จะละการระบุซับเจ็กต์ได้ และเครื่องหมายจุลภาค (,) เป็นการระบุว่าทริปเปิลถัดไปมีทั้งซับเจ็กต์และเพรดิเคตคู่เดียวกัน แตกต่างกันเพียงแค่อ็อบเจ็กต์ดังนั้นจึงระบุเพียงแค่อ็อบเจ็กต์เท่านั้น เครื่องหมายเหล่านี้ทำให้ syntax ของ N3 และ Turtle แลดูเรียบง่ายสะอาดตาเพื่อให้มนุษย์สามารถอ่านได้โดยง่าย

การกำหนด blank node ใน N3 และ Turtle นอกจากจะใช้วิธีตาม N-Triples แล้วก็สามารถใช้ syntactic sugar สำหรับสร้าง blank node โดยใช้เครื่องหมาย [และ] ดังที่ได้กล่าวไปในหัวข้อที่แล้ว กราฟในภาพที่ 20 เมื่อนำมาเขียนด้วย N3 หรือ Turtle จะได้ออกมาดังนี้

```
@prefix : <http://example.org/sample#> .

:Julthep :graduate :MSc ;
    :studyAt :Chulalongkorn ;
    :weight "70.0 kg" ;
    :height "175.0 cm" ;
    :liveIn [
        :address "12/34" ;
        :street :NawongRd ;
        :district :Donmuang
    ] .
:MSc :program :ComputerScience .
```

คุณสมบัติที่สำคัญอีกอย่างหนึ่งของ N3 และ Turtle คือการสร้าง RDF collection หรือ RDF list โดยใช้เครื่องหมายวงเล็บเปิดและวงเล็บปิดตามต้นแบบ DAML[37] ซึ่งนำเอารูปแบบมาจาก S-expression[38] หรือ symbolic expression ของภาษา Lisp ตัวอย่างเช่น หากต้องการสร้าง RDF

[36] RDF 1.1 Turtle. W3C. Available from: https://www.w3.org/TR/turtle/

[37] DARPA Agent Markup Language. Available from: http://www.daml.org/

[38] S-expression. Wikipedia. Available from: https://en.wikipedia.org/wiki/S-expression

list ที่ชื่อว่า :List1 ให้มีสมาชิกคือ 1, 2 และ 0 ตามลำดับ หากใช้ N-Triples จำเป็นต้องใช้ถึง 7 บรรทัดในการสร้างคือ

```
:List1      owl:sameAs _:ListItem1
_:ListItem1 rdf:first  1 .
_:ListItem1 rdf:rest   _:ListItem2 .
_:ListItem2 rdf:first  2 .
_:ListItem2 rdf:rest   _:LastItem .
_:LastItem  rdf:first  0 .
_:LastItem  rdf:rest   rdf:nil .
```

แต่หากใช้ N3 หรือ Turtle จะใช้เพียงแค่ 1 บรรทัดเท่านั้นคือ

```
:List1 owl:sameAs ( 1 2 0 ) .
```

โดย syntactic sugar นี้จะสร้าง RDF collection หรือ RDF list ที่มีจำนวน 7 triples ในลักษณะเดียวกันให้โดยอัตโนมัติ

5.4 RDFa (RDF in attributes)

RDFa หรือ RDF in attributes[39] เป็นรูปแบบมาตรฐานที่กำหนดโดย W3C เพื่อนำเอาข้อมูล RDF ทริปเปิลฝังไว้ใน HTML โดยอยู่ในรูปของ attributes ของ HTML แท็ก ซึ่ง attributes ของ RDFa โดยทั่วไปมีลักษณะคล้ายกับ RDF/XML แต่จะดูเรียบง่ายกว่า ข้อเด่นที่สำคัญคือเราสามารถนำเอา attributes เหล่านี้ไปผนวกรวมหรือเสริมเพิ่มเติมเข้ากับแท็กของเนื้อหาเดิมที่อยู่ในรูปแบบ HTML อยู่แล้ว ซึ่งเป็นการเพิ่มความหมายหรือ semantic เข้าไปที่ตัวเอกสารเลยโดยตรง ตัวอย่างกราฟในภาพที่ 20 เมื่อนำมาเขียนด้วย RDFa โดยสมมติว่าแท็ก คือแท็กเดิมที่อยู่ในเอกสาร HTML จะได้ออกมาดังนี้


```
<div xmlns="http://www.w3.org/1999/xhtml"
  prefix="rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns#
          rdfs: http://www.w3.org/2000/01/rdf-schema#"
          : http://example.org/sample#>
  <div typeof="rdfs:Resource" about=
                                "http://example.org/sample#Julthep">
    <div rel=":graduate">
      <div typeof="rdfs:Resource" about=
                                    "http://example.org/sample#MSc">
        <div rel=":program" resource=
                        "http://example.org/sample#ComputerScience">
        </div>
      </div>
    </div>
    <div rel=":studyAt" resource=
```

[39] RDFa 1.1 Primer. W3C. Available from: https://www.w3.org/TR/rdfa-primer/

```
                                "http://example.org/sample#Chulalongkorn">
    </div>
    <div property=":weight" content="70.0 kg"></div>
    <div property=":height" content="175.0 cm"></div>
    <div rel=":liveIn">
      <div typeof="rdfs:Resource">
        <div property=":address" content="12/34"></div>
        <div rel=":street" resource=
                                "http://example.org/sample#NawongRd">
        </div>
        <div rel=":district" resource=
                                "http://example.org/sample#Donmuang">
        </div>
      </div>
    </div>
  </div>
</div>
```

6. SPARQL (SPARQL Protocol and RDF Query Language)[40]

SPARQL คือภาษาคิวรีหรือภาษาที่ใช้ในการสืบค้นข้อมูลที่มีรูปแบบไวยกรณ์คล้าย ๆ กับ SQL (Structured Query Language ภาษาคิวรีที่ใช้สืบค้นข้อมูลจากฐานข้อมูลเชิงสัมพันธ์หรือ relational database) ใช้ในการดึงข้อมูลจากฐานข้อมูลที่เก็บทริปเปิล (หรือเรียกว่า triplestore)

ตัวอย่างเช่น คำสั่ง SPARQL ดังนี้

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX dbpedia: <http://dbpedia.org/>
PREFIX dbpedia2: <http://dbpedia.org/property/>
PREFIX : <http://dbpedia.org/resource/>

SELECT ?property ?value
WHERE {
      :Chulalongkorn_University ?property ?value .
}
```

เริ่มต้นเรากำหนด namespace prefix ของรีซอร์สที่เราต้องการใช้ลงไป เพื่อจะให้ง่ายในการระบุ URI ต่าง ๆ ในการเขียนคิวรี และเพื่อให้ดูง่ายในการแสดงผลลัพธ์ เพราะว่าส่วนประกอบต่าง ๆ ของ triple จะเป็น URI ทั้งหมด ซึ่ง namespace prefix ในรูปแบบของ SPARQL ก็จะมีความคล้ายคลึงกันกับ namespace prefix ในรูปแบบ N3/Turtle หรือ RDF/XML

[40] SPARQL 1.1 Query Language. W3C. Available from: https://www.w3.org/TR/sparql11-query/

หลังจากนั้นก็จะเป็นส่วนของคำสั่งคิวรี ซึ่งโดยทั่วไปจะมีอยู่ 4 คำสั่งคือ SELECT, CONSTRUCT, ASK และ DESCRIBE ในตัวอย่างที่ยกมาเราจะทำการ SELECT เพื่อสืบค้นหารีซอร์สที่ชื่อว่า Chulalongkorn_University จากฐานข้อมูลเดต้าเซ็ตของ DBpedia ว่ามี property และ value อะไรบ้าง SPARQL endpoint (เป็นเกตเวย์ที่ใช้ในการติดต่อส่งคำสั่งสืบค้นไปยัง SPARQL engine และส่งผลลัพธ์ที่ได้กลับมา) ก็จะแสดงรายการเพรดิเคตและอ็อบเจ็กต์ (รวมไปถึง literal) มาทั้งหมดที่ค้นหาได้ ดังตัวอย่างในภาพที่ 21 ซึ่งแสดงเพียงบางส่วน โดยจะแสดงรายการมาเป็นพร็อปเพอร์ตี้หรือเพรดิเคต และอ็อบเจ็กต์ตามชื่อ key fields ที่เราตั้งไว้ในคิวรีคือ property กับ value

node	property	value
<http://dbpedia.org/resource/Chulalongkorn_University>	dbpedia:ontology/wikiPageExternalLink	<http://www.chula.ac.th/cuen/index.htm>
	dbpedia:ontology/founder	:King_Vajiravudh
	dbpedia:ontology/president	:Pirom_Kamol-Ratanakul
	dbpedia:ontology/type	:Public_university
	dbpedia:ontology/officialSchoolColour	"Pink"@en
	dbpedia:ontology/city	:Bangkok
	foaf:homepage	<http://www.chula.ac.th/cuen/index.htm>
	<http://purl.org/dc/terms/subject>	:Category:Chulalongkorn_University
	<http://purl.org/dc/terms/subject>	:Category:ASEAN_University_Network
	<http://purl.org/dc/terms/subject>	:Category:Universities_in_Bangkok
	<http://purl.org/dc/terms/subject>	:Category:Association_of_Pacific_Rim_Universities
	<http://purl.org/dc/terms/subject>	:Category:Educational_institutions_established_in_1917
	<http://purl.org/dc/terms/subject>	:Category:Education_in_Bangkok
	dbpedia2:established	"1917-03-26"^^xsd:date
	dbpedia2:campus	:Urban_area
	dbpedia2:students	36199
	<http://www.georss.org/georss/point>	"13.73826 100.532413"@en
	dbpedia2:type	:Public_university
	dbpedia2:city	:Bangkok
	dbpedia2:wikiPageUsesTemplate	:Template:Infobox_university
	dbpedia2:wikiPageUsesTemplate	:Template:Navboxes
	dbpedia2:name	"Chulalongkorn University"@en

ภาพที่ 21 ผลลัพธ์ที่ได้จาก SPARQL คิวรีตัวอย่างเพียงบางส่วน

SPARQL มีคุณสมบัติพิเศษอย่างหนึ่งที่เรียกว่า Property Path[41] เพื่อให้การคิวรีสามารถระบุลงไปเป็น path ของพร็อปเพอร์ตี้ต่อเนื่องลงไปได้หลายชั้นโดยไม่จำเป็นต้องสนใจอ็อบเจ็กต์ที่เป็นตัวเชื่อมของทริปเปิลระหว่าง path เหล่านั้น จะขออธิบายด้วยการยกตัวอย่าง property path ในบทถัดไปที่กล่าวถึงการนำ property path ไปใช้ในการคิวรีข้อมูลตัวอย่างที่ได้จากงานวิจัย

[41] SPARQL 1.1 Property Paths. W3C. Available from: https://www.w3.org/TR/sparql11-property-paths/

7. RDFS (RDF Schema)

RDF Schema หรือ RDFS[42] เป็นส่วนสำคัญอีกส่วนหนึ่งในการกำหนดชนิดของ schema ของ RDF ดังที่ได้กล่าวไปในหัวข้อที่ผ่านมา โดยหลักการแล้ว RDF เก็บอยู่ในโครงสร้างแบบ EAV model หรือ open schema ซึ่งในตัวข้อมูลเองเราจะไม่เห็นโครงสร้างความสัมพันธ์เนื่องจาก schema เป็นแบบเปิดหรือแบบยืดหยุ่น ดังนั้นการที่เราสร้าง RDF ทริปเปิล เราต้องใส่ schema ลงไปเพื่อจะบอกว่าพร็อปเพอร์ตี้ที่เราใช้ในการอธิบาย RDF ทริปเปิลนั้น แต่ละตัวมีลักษณะเป็นอย่างไร

ภาพที่ 22 ตัวอย่าง RDF Schema เพื่อระบุ schema ของพร็อปเพอร์ตี้ :studyAt ในภาพที่ 20

ตัวอย่างเช่นพร็อปเพอร์ตี้ :studyAt ตามตัวอย่างในภาพที่ 20 เราสามารถกำหนด schema เพื่อกำหนดว่า domain ของ :studyAt จะต้องเป็น :Student และ range คือ :University ซึ่ง RDFS เป็นเหมือน metaformat ของ RDF อีกทีนึง โดยที่ตัวมันเองก็อธิบายด้วย RDF ทริปเปิลเช่นเดียวกัน ดังนั้นเราก็สามารถแสดงด้วย RDF กราฟได้ดังภาพที่ 22 ซึ่งสามารถเขียนเป็น RDF ทริปเปิลได้ดังนี้

```
:studyAt rdfs:domain     :Student.
:studyAt rdfs:range      :University.
:Student rdfs:subClassOf foaf:Person.
```

พอเรากำหนด RDF Schema ดังนี้แล้วเราก็สามารถจะอนุมานหรือ infer อะไรบางอย่างตามหลักของ semantic network ได้ ยกตัวอย่างเช่น RDF ทริปเปิลในกราฟตามภาพที่ 20 ที่ระบุว่า :Julthep :studyAt :Chulalongkorn เราก็สามารถจะอนุมานได้ว่า :Chulalongkorn เป็น type :University ดังนั้นจึงหมายความว่า :Julthep เรียนที่มหาวิทยาลัยแห่งหนึ่งคือ :Chulalongkorn และ :Julthep ก็เป็น type :Student ในขณะที่ :Student ก็เป็น subclass ของ foaf:Person ด้วย ดังนั้นพอเรารู้สองอย่างนี้ เราก็สามารถจะอนุมานต่อไปได้ว่า :Julthep ก็เป็น foaf:Person คนหนึ่งอีกด้วย ซึ่งคือการ inference ต่อเนื่องกันไปนั่นเอง

[42] RDF Schema 1.1. W3C. Available from: https://www.w3.org/TR/rdf-schema/

ลิงก์เดต้า และข้อมูลเปิดแบบห้าดาว

มีการกล่าวว่า ซีแมนติกเว็บถึงแม้ว่าจะเป็นไอเดียที่เรียบง่าย แต่ก็ยังคงไม่ได้ถูกนำไปใช้งานจริงอย่างกว้างขวาง[9] ลิงก์เดต้า (Linked Data[43]) เป็นข้อหนดแนวทางปฏิบัติในการเปิดเผย, แบ่งปัน และเชื่อมโยงชิ้นส่วนข้อมูล, สารสนเทศ หรือความรู้บนซีแมนติกเว็บโดยใช้ URI และ RDF [10] ดังนั้นลิงก์เดต้าจึงเปรียบเสมือนเป็นแอปพลิเคชันหนึ่งที่นำเอาเทคโนโลยีซีแมนติกเว็บไปใช้ กล่าวได้ว่าซีแมนติกเว็บโดยตัวมันเองเป็นแค่เทคโนโลยี ในทำนองเดียวกันกับเทคโนโลยีของเว็บ การจะนำเอาไปใช้ทำงานต่าง ๆ ขึ้นกับว่าเรานำไปพัฒนาต่ออย่างไร ลิงก์เดต้าคือการประยุกต์เอาเทคโนโลยีซีแมนติกเว็บมาเก็บข้อมูลเสมือนเป็นฐานข้อมูลแบบกระจายหรือ distributed database ขนาดมหึมา ส่วน Linked Open Data Project (LOD)[44] โดย Chris Bizer และ Richard Cyganiak มีเป้าหมายเพื่อขยายเว็บด้วยข้อมูลที่ใช้ร่วมกันโดยการกระจายชุดข้อมูลแบบเปิดในรูปแบบ RDF บนซีแมนติกเว็บ และสร้าง RDF ลิงก์เชื่อมโยงระหว่างข้อมูลเปิดจากแหล่งข้อมูลที่กระจายกันอยู่[11] โดยมีการกำหนดลำดับขั้นของการเปิดเผยข้อมูลในลิงก์เดต้าระบุเป็นจำนวนดาว (★) ดังนี้[45] (แต่ละขั้นเป็นซุปเปอร์เซ็ตของระดับที่ต่ำกว่า)

★ เป็นขั้นแรกสุด มีข้อกำหนดเพียงแค่เปิดเผยข้อมูลเป็นสาธารณะในลักษณะ open license หรือ free license[46] ไม่ว่าจะรูปแบบใดก็ตาม เช่น ไฟล์ที่ได้มาจากการสแกนภาพมาและยังไม่ได้ทำการ OCR (Optical Character Recognition) เป็นต้น

★★ ขั้นนี้มีข้อกำหนดว่าข้อมูลที่เปิดเผยนั้นต้องอยู่ในรูปแบบที่ไม่ใช่ข้อมูลไร้โครงสร้าง ยกตัวอย่างเช่น ไฟล์ Excel เวอร์ชันใดก็ตาม เป็นต้น

★★★ ขั้นนี้กำหนดว่าข้อมูลที่อยู่ในรูปแบบเชิงโครงสร้างนั้นต้องอยู่ในมาตรฐานเปิด (non-proprietary) เช่น ไฟล์ CSV (Comma-Separated Values) หรือไฟล์ Excel เวอร์ชันตั้งแต่ 2007 ขึ้นไปที่มีนามสกุลเป็น XLSX (หากเป็นนามสกุล XLS ซึ่งยังเป็นรูปแบบ proprietary ถือว่าเป็นแค่ 2 ดาว)

---

[43] Linked Data - Connect Distributed Data across the Web. Linked Data. Available from: http://linkeddata.org/

[44] The Linked Open Data Cloud. Available from: https://lod-cloud.net/

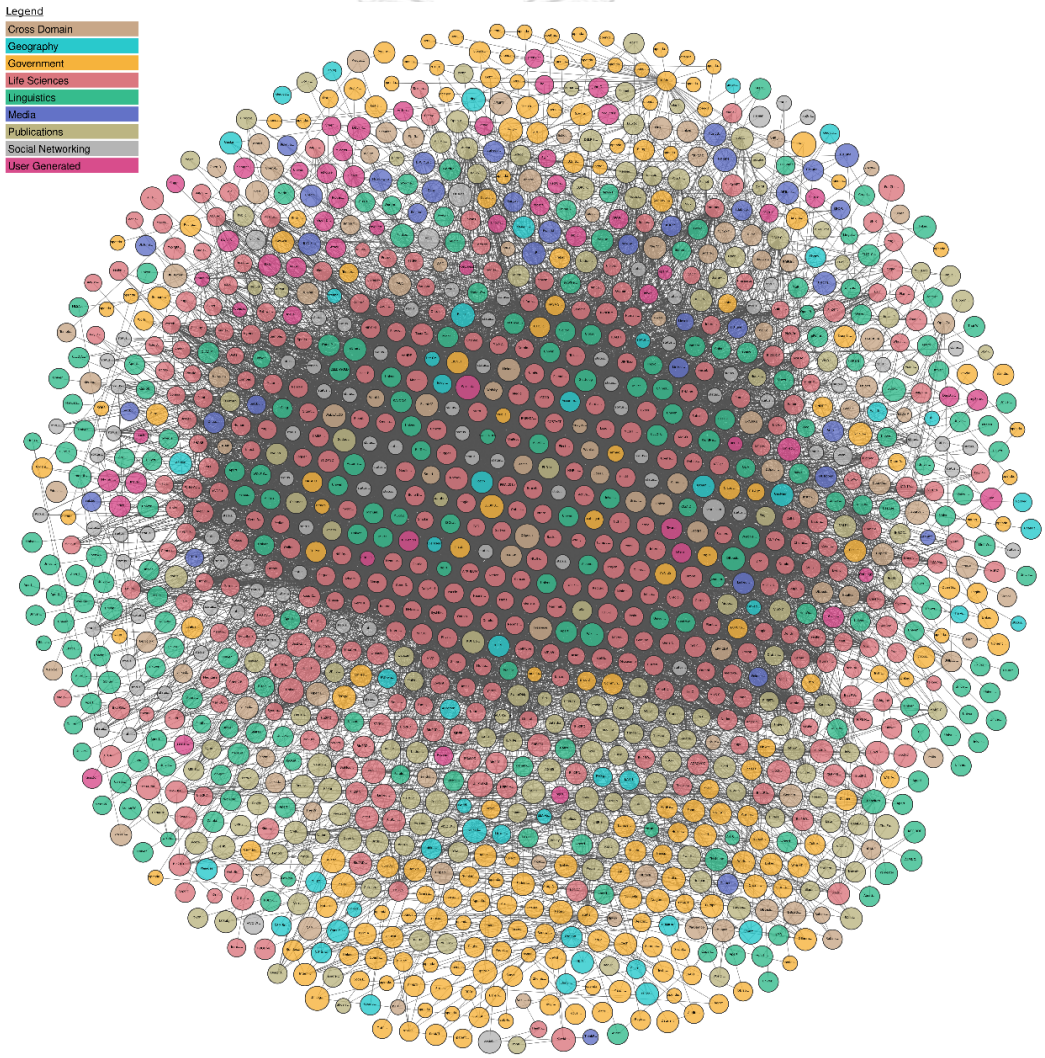
[45] 5-star Open Data. Available from: https://5stardata.info/en/

[46] Free license. Wikipedia. Available from: https://en.wikipedia.org/wiki/Free_license

★★★★ กำหนดว่าข้อมูลต้องเป็นไปตามมาตรฐานเปิดในรูปแบบของซีแมนติกเว็บ เช่น ใช้ URI และ RDF ในการเปิดเผยข้อมูล

★★★★★ ต้องเชื่อมโยงข้อมูลนั้นเข้ากับข้อมูลเปิดอื่น ๆ เพื่อเป็นลิงก์เดต้าโดยสมบูรณ์

ทุกวันนี้ Linked Open Data (LOD) cloud จะมีเดต้าเซ็ตใหม่เพิ่มมากขึ้นเรื่อย ๆ ภาพที่ 23 แสดงถึงการเชื่อมโยงกันของเดต้าเซ็ตต่าง ๆ ในลิงก์เดต้าระดับ 5 ดาว มีจำนวน 1,269 เดต้าเซ็ตที่เชื่อมต่อเข้าด้วยกันผ่าน 16,201 เส้นทาง นับรวมแล้วทั้งหมดมากกว่าหนึ่งแสนล้านทริปเปิลที่มีการเชื่อมโยงถึงกันด้วย RDF ลิงก์จำนวนมากกว่าหนึ่งพันห้าร้อยล้านลิงก์



ภาพที่ 23 LOD cloud เดือนพฤศจิกายน 2020[47]

[47] The Linked Open Data Cloud. Available from: https://lod-cloud.net/

วิกิ, วิกิพีเดีย และมีเดียวิกิ

วิกิ (wiki) เป็นแนวคิดที่สร้างขึ้นโดย Ward Cunningham เพื่อเป็นเครื่องมือที่ใช้ในการแบ่งปันความรู้ระหว่างผู้ใช้ด้วยกันบนเว็บโดยที่ผู้ใช้ไม่จำเป็นต้องมีความรู้เกี่ยวกับการใช้ HTML ในการสร้างเว็บเพจ คำว่า “wiki” มาจากภาษาฮาวาย มีความหมายว่า “รวดเร็ว” โปรแกรมวิกิโปรแกรมแรกที่ถูกสร้างขึ้นคือ WikiWikiWeb โดย Cunningham ในปี ค.ศ. 1995 หลังจากนั้นก็มีระบบซอฟท์แวร์วิกิเกิดขึ้นมากมายบนฮาร์ดแวร์และโอเอสที่แตกต่างกัน[48] รวมถึงประเภทเนื้อหาที่แบ่งปันกันในแต่ละเว็บไซต์วิกิต่าง ๆ ก็มีความหลากหลายเฉพาะเจาะจงไปในแต่ละโดเมนที่มีผู้ใช้แต่ละกลุ่มนั้น ๆ สนใจ แต่ระบบวิกิที่มีผู้ใช้มากที่สุดและเนื้อหาหลากหลายที่สุดก็คือวิกิพีเดีย

วิกิพีเดีย (Wikipedia[49]) ถูกสร้างขึ้นโดย Jimmy Wales ซึ่งในตอนแรกได้พัฒนาระบบสารานุกรมออนไลน์ขึ้นในปี ค.ศ. 2000 ชื่อว่า Nupedia[50] ที่สามารถเข้าถึงได้บนเว็บ ผู้ใช้สามารถแบ่งปันความรู้ได้โดยผ่านระบบ peer review เพื่อตรวจสอบความถูกต้องของข้อมูล (ลักษณะเดียวกันกับการตีพิมพ์ผลงานทางวิชาการ) แต่ด้วยระบบที่มีความซับซ้อนถึง 7 ขั้นตอนในการอนุมัติเพื่อเพิ่มเติมเนื้อหา ทำให้ในปีแรกมีเพียงแค่ 21 บทความถูกเพิ่มเติมเข้าไปในระบบ แต่ Wales ก็ไม่ยอมหยุดยั้งที่จะสร้างระบบสารานุกรมออนไลน์ที่มีปริมาณเนื้อหาเป็นจำนวนมากให้ได้ จึงได้สร้างระบบวิกิพีเดียขึ้นมาในปี ค.ศ. 2001 ซึ่งครั้งนี้เขาได้เปลี่ยนแนวความคิดเกี่ยวกับการตรวจสอบความถูกต้องของบทความโดยไม่ต้องมี peer review แต่ได้นำระบบวิกิมาใช้เพื่อให้มีการเพิ่มเติมเนื้อหาได้ในแบบทันทีทันใด และให้กลุ่มของผู้ใช้ด้วยกันเป็นผู้ตรวจทานเนื้อหาด้วยกันเอง ด้วยการเพิ่มเติมเนื้อหาที่ง่ายสะดวกรวดเร็ว และแนวคิดเพิ่มเติมไปก่อนค่อยมาตรวจสอบทีหลัง ทำให้มีเนื้อหาถึง 200 บทความถูกสร้างขึ้นในเดือนแรก และขยายเพิ่มจนเป็น 18,000 บทความในปีแรก ด้วยความนิยมของวิกิพีเดียที่มีปริมาณผู้ใช้และเนื้อหาที่เพิ่มมากขึ้นเรื่อย ๆ ในที่สุด Nupedia ก็ถูกปิดตัวลงในปี ค.ศ. 2003 ปัจจุบันนี้วิกิพีเดียแบ่งเป็นระบบย่อยแยกไปในแต่ละภาษา ซึ่งตอนนี้มีอยู่มากกว่า 300 ภาษา[51]

ส่วนมีเดียวิกิ (MediaWiki[52]) คือชื่อของซอฟท์แวร์ที่ขับเคลื่อนวิกิพีเดีย พัฒนาขึ้นโดย Magnus Manske และ Lee Daniel Crocker ด้วยภาษา PHP และแจกจ่ายสู่สาธารณะในลักษณะ open-source ทำให้นอกจากวิกิพีเดียแล้ว ก็ยังมีเว็บวิกิอื่น ๆ ทำงานอยู่บนระบบมีเดียวิกิเป็นจำนวน

[48] List of wiki software. Wikipedia. Available from: https://en.wikipedia.org/wiki/List_of_wiki_software

[49] Wikipedia. Available from: https://www.wikipedia.org/

[50] Nupedia. Available from: https://en.wikipedia.org/wiki/Nupedia

[51] List of Wikipedias. Wikipedia. Available from: https://en.wikipedia.org/wiki/List_of_Wikipedias

[52] MediaWiki. Available from: https://www.mediawiki.org/

มาก เว็บที่เป็นที่รู้จักมากที่สุดรองลงมา[53] คือวิเกียแฟนดอม (Wikia Fandom[54]) ซึ่งกำเนิดโดย Jimmy Wales ผู้สร้างวิกิพีเดียเช่นกัน ปัจจุบันมีผู้ใช้เฉพาะที่ลงทะเบียนแล้วมากกว่า 25 ล้านคน[55] โดยที่แต่ละเดือนจะมีผู้ใช้มากกว่า 200 ล้านคนเข้าใช้ชุมชนต่าง ๆ ในวิเกียแฟนดอมที่มีจำนวนมากกว่า 4 แสนชุมชน[56]

ส่วนประกอบของวิกิพีเดีย สามารถแบ่งออกได้เป็น 2 ส่วนประกอบหลัก ส่วนแรกคือส่วนติดต่อกับผู้ใช้ ซึ่งมีส่วนของ header รวมส่วนที่ใช้ล็อกอิน, เมนู, แท็บ, footer เป็นต้น และอีกส่วนหนึ่งคือส่วนเนื้อหาบทความ ซึ่งบทความในวิกิพีเดียโดยทั่วไปมักจะประกอบไปด้วยส่วนประกอบดังต่อไปนี้

- article title – ชื่อของบทความ หรือหัวเรื่อง
- abstract – บทนำหรือบทคัดย่อของบทความ จะปรากฏเป็นย่อหน้าแรกบนสุดของบทความ
- Infobox template – กล่องข้อมูลสรุปเกี่ยวกับบทความ ซึ่งจะแตกต่างกันไปตามแต่ละประเภทของบทความ
- geolocation – ตำแหน่งทางภูมิศาสตร์ของสิ่งที่ตรงกับชื่อของบทความ เช่น ประเทศ ยอดเขา น้ำตก อนุสรณ์สถาน ฯลฯ
- table of contents – สารบัญ ซึ่งจะสร้างขึ้นโดยอัตโนมัติจาก section headers
- section headers – หัวข้อเรื่องย่อยซึ่งจะมีได้หลายลำดับชั้น
- languages – ลิงก์เชื่อมโยงไปที่บทความเดียวกันในภาษาอื่นที่อยู่บนเครื่องเซิร์ฟเวอร์ของภาษานั้นๆ
- Wikitext – เนื้อหาในรูปแบบเรียงความ ซึ่งเป็นข้อควรปฏิบัติของวิกิพีเดีย ที่บทความจะต้องอยู่ในรูปแบบเรียงความให้มากที่สุดเท่าที่จะเป็นไปได้ เพื่อความสะดวกในการติดตามเรื่องราวโดยผู้ใช้ และ Wikitext ในอีกความหมายหนึ่งคือชื่อของ markup language ของมีเดียวิกิ ที่ใช้ในการกำหนดรูปแบบการแสดงเนื้อหาบทความของวิกิพีเดีย
- Wikilink – ลิงก์เชื่อมโยงภายในระหว่างบทความต่าง ๆ ที่มีการอ้างอิงถึงกัน

[53] Wikia อยู่อันดับที่ 48 เมื่อเดือนสิงหาคม 2018. แหล่งที่มา: https://web.archive.org/web/20180815220852/https://www.alexa.com/siteinfo/wikia.com

[54] FANDOM powered by Wikia. Available from: http://www.wikia.com/fandom

[55] Statistics. FANDOM powered by Wikia. Available from: https://community.fandom.com/wiki/Special:Statistics

[56] About FANDOM. Available from: https://www.fandom.com/about

- pictures – รูปภาพขนาดย่อ และเป็นลิงก์เชื่อมโยงไปที่รูปภาพขนาดเต็มรวมถึงรายละเอียดที่เกี่ยวกับรูปภาพนั้น
- tables – ข้อมูลที่ถูกนำเสนอในรูปแบบตาราง
- lists – รายการ ทั้งแบบมีหมายเลขลำดับ และไม่มีหมายเลข
- references – รายการข้อมูลอ้างอิง
- related articles/see also – รายการบทความที่มีความสัมพันธ์เกี่ยวข้องกันกับหัวเรื่อง
- external links – ลิงก์เชื่อมโยงไปสู่เว็บไซต์ภายนอกวิกิพีเดีย
- Navbox template – กล่องนำทางไปยังบทความที่อยู่ในชุดเดียวกันหรือเกี่ยวข้องกัน
- categories – หมวดหมู่ต่าง ๆ ที่เกี่ยวกับบทความ

บทที่ 3

งานวิจัยที่เกี่ยวข้อง

มีงานวิจัยหลายชิ้นที่เกี่ยวข้องกับการสกัดข้อมูลจากวิกิพีเดียไปสู่ลิงก์เดต้า บางงานวิจัยเกี่ยวข้องกับการสกัดข้อมูลประเภทตารางในรูปแบบต่าง ๆ เช่น สเปรดชีต ฐานข้อมูลเชิงสัมพันธ์ ฯลฯ แล้วนำมาแปลงเป็นข้อมูล RDF งานวิจัยเหล่านั้นสามารถนำแบ่งออกเป็นกลุ่มได้ดังนี้

งานวิจัยที่เกี่ยวข้องกับการสร้างข้อเท็จจริงเข้าสู่ลิงก์เดต้า

มีงานวิจัยมากมายที่เกี่ยวข้องกับการบรรจุข้อเท็จจริงเข้าสู่ลิงก์เดต้า โครงการที่มีการกล่าวถึงมากที่สุดคือ DBpedia, YAGO, Freebase, Wikidata และ OpenCyc โดยได้มีงานวิจัย survey เปรียบเทียบออกมาหลายชิ้น[12] [13] [14] [15]และยังมีงานวิจัยอีกหลายชิ้นนอกจากนี้ที่มีความเกี่ยวข้อง ซึ่งสามารถแบ่งออกเป็นกลุ่มได้ดังนี้

1. <u>การสกัดข้อเท็จจริงจากส่วนประกอบต่าง ๆ ในวิกิพีเดีย</u>

DBpedia[1] เป็นงานวิจัยร่วมกันของ Free University of Berlin และ Leipzig University แห่งเยอรมนี[16] มีวัตถุประสงค์เพื่อทำการสกัดข้อมูลเชิงโครงสร้างของวิกิพีเดีย เช่น Infoboxes และ categories (รวมถึงข้อมูลไร้โครงสร้างบางส่วนเช่น abstract โดยนำมาทั้งชุดโดยไม่มีการสกัดคัดแยกข้อความในบทคัดย่อออกเป็นข้อมูลย่อย)[17] DBpedia รองรับข้อมูลของวิกิพีเดียในหลายๆ ภาษา[18] เป้าหมายของโครงการนี้คือการเป็นแกนกลางในการเชื่องโยงฐานข้อมูลอื่น ๆ ของลิงก์เดต้าทั้งหมดเข้าด้วยกัน[19] ผลลัพธ์ที่ได้คือมากกว่า 3 พันล้านข้อเท็จจริงเกี่ยวกับ 4.58 ล้านหัวข้อ ซึ่งแบ่งเป็นเกือบ 600 ล้านข้อเท็จจริงจากบทความในวิกิพีเดียภาษาอังกฤษ ส่วนที่เหลือมากกว่า 2.5 พันล้านข้อเท็จจริงจากวิกิพีเดียภาษาอื่น ๆ

ด้วยข้อจำกัดในการสกัดข้อมูลจากตาราง เช่น วิธีในการจำแนกตารางชนิดต่าง ๆ และการกำหนดชื่อให้กับข้อมูล DBpedia จึงเลือกที่จะสกัดเฉพาะข้อมูลที่เป็นโครงสร้างอยู่แล้วเท่านั้น[17] แต่อย่างไรก็ตามได้มีการเพิ่มเติมความสามารถในเฟรมเวิร์ครุ่นถัดมาให้สามารถสกัดข้อมูลตารางในวิกิ

[1] DBpedia. Available from: https://wiki.dbpedia.org/

พีเดียได้[16] แต่ก็เป็นเพียงแค่การสกัดข้อมูลตารางมาทั้งตารางในรูปแบบ HTML เท่านั้น ยังไม่ได้มีการสกัดข้อเท็จจริงจากตารางเหล่านั้นให้เป็นทริปเปิลย่อย ๆ ที่สามารถนำไปคิวรีต่อโดยใช้ SPARQL ได้โดยตรง แต่อย่างไรก็ตาม DBpedia ได้สนับสนุนให้มีการเข้าร่วมพัฒนาอัลกอริทึมในการสกัดข้อมูลจากตารางและรายการจากวิกิพีเดียโดยใช้เฟรมเวิร์คของ DBpedia โดยเสนอโครงการเข้าร่วมในงาน Google Summer of Code (GSoC) ตั้งแต่กลางปี ค.ศ. 2016[23] และยังเข้าโครงการอีกครั้งในปี ค.ศ. 2017[45] โดยไม่มีโครงการต่อเนื่องในปีถัดมา แต่ยังไม่มีการตีพิมพ์ผลงานทางวิชาการที่เกี่ยวข้อง และยังไม่มีการนำไปใช้จริงในการสกัดเดต้าเซ็ตของ DBpedia ในเวอร์ชันล่าสุด

Isbell และ Butler ได้ตีพิมพ์งานวิจัยชิ้นหนึ่งที่สร้างขึ้นที่ Digital Media Systems Laboratory ของบริษัท HP[20] โดยทำการศึกษาการแปลงข้อมูลจาก Infoboxes ของวิกิพีเดียซึ่งเป็นข้อมูลเชิงโครงสร้างเป็นหลัก แต่ก็มีบางส่วนครอบคลุมไปถึงข้อมูลกึ่งโครงสร้างและไร้โครงสร้างอีกด้วย

YAGO[6] เป็นงานวิจัยของ Max Planck Institute for Informatics แห่งเยอรมนี[21] มีวัตถุประสงค์เพื่อทำการสกัดข้อมูลเชิงโครงสร้างจากวิกิพีเดียในส่วนของ categories โดยการนำข้อมูล Synsets จากโครงการ WordNet ของ Princeton University มาใช้ร่วมด้วย ผลลัพธ์ที่ได้คือ 120 ล้านข้อเท็จจริงจาก 10 ล้านหัวข้อ

BabelNet[22] และ MENTA (Multilingual Entity Taxonomy)[23] สกัดข้อเท็จจริงจากวิกิพีเดียและ WordNet เช่นเดียวกันกับ YAGO แต่ BabelNet และ MENTA มีเป้าหมายเพื่อสร้างฐานความรู้แบบหลายภาษาเป็นหลัก

2. <u>การบันทึกข้อเท็จจริงลงไปในฐานความรู้ด้วยมือ</u>

Freebase[7] ของบริษัท Metaweb Technologies[24] เป็นฐานความรู้บนเว็บที่ผู้ใช้ร่วมกันแบ่งปันข้อมูลเชิงโครงสร้างลงไปในระบบโดยตรงผ่านหน้าเว็บที่ออกแบบมาสำหรับการบันทึกและตรวจสอบแก้ไขข้อมูลโดยเฉพาะ[25] (แตกต่างจาก DBpedia และ YAGO ที่ข้อมูลเชิงโครงสร้างถูกแปลงมาจากวิกิพีเดียอีกทอดหนึ่ง) หลังจากถูกเข้าซื้อกิจการโดยบริษัท Google ในปี ค.ศ. 2010

[2] The List Extractor. GSoC 2016. from: https://summerofcode.withgoogle.com/archive/2016/projects/5451876061413376/

[3] The Table Extractor. GSoC 2016. from: https://summerofcode.withgoogle.com/archive/2016/projects/4881737138044928/

[4] The List Extractor. GSoC 2017. from: https://summerofcode.withgoogle.com/archive/2017/projects/6628605295591424/

[5] The Table Extractor. GSoC 2017. from: https://summerofcode.withgoogle.com/archive/2017/projects/6608069714771968/

[6] YAGO. Max-Planck-Institut für Informatik. Available from: http://mpi-inf.mpg.de/yago/

[7] Freebase. Available from: https://web.archive.org/web/20160501004947/http://www.freebase.com/

ข้อมูล[8]ก็ถูกโอนย้ายเข้าสู่ Wikidata ในปี ค.ศ. 2014 และในปี ค.ศ. 2016 Freebase ก็ถูกปิดเพื่อยุบรวมบริการเข้าสู่ Knowledge Graph ของ Google และพัฒนาต่อมาเป็น Knowledge Vault ซึ่งเป็นงานวิจัยของ Google โดยมีวัตถุประสงค์เพื่อสร้างกระบวนการอัตโนมัติในการสร้างฐานความรู้มาจากเว็บโดยตรง[26]

อย่างไรก็ตาม นอกจากความรู้ที่ผู้ใช้ร่วมกันสร้างเข้าไปในระบบผ่านเว็บแล้ว Freebase ยังได้รวบรวมข้อมูลหลายส่วนจากวิกิพีเดีย[27] (รวมไปถึง NNDB – Notable Names Database[9], FMD – Fashion Model Directory[10] และ MusicBrainz[11]) เพื่อเป็นข้อมูลตั้งต้น (seed data) จำนวนมาก ก่อนจะปิดตัวลง Freebase สะสมข้อเท็จจริงไว้ถึง 2.4 พันล้านข้อเท็จจริงใน 44 ล้านหัวข้อ

Wikidata[12] เป็นโครงการหนึ่งของ Wikimedia Foundation[28] เพื่อเป็นฐานความรู้แบบเปิดโดยให้ผู้ใช้สามารถบันทึกข้อเท็จจริงลงไปเองด้วยมือผ่านระบบที่ออกแบบให้ใช้งานได้ง่ายคล้ายกับวิกิพีเดีย แนวคิดที่น่าสนใจอย่างหนึ่งของ Wikidata คือ สามารถเก็บข้อเท็จจริงที่มีความขัดแย้งกันได้ หากไม่สามารถหาข้อสรุปได้ว่าข้อเท็จจริงใดมีความถูกต้องมากกว่ากัน[29] เพราะ “ความน่าเชื่อถือ” ของข้อมูลใน Wikidata (รวมทั้งวิกิพีเดีย) ไม่ได้ให้ความสำคัญกับ “ความถูกต้อง” ของข้อมูลมากไปกว่า “การพิสูจน์ที่มา” ของข้อมูลนั้นได้ ตัวอย่างเช่น ข้อมูลจำนวนประชากรของมุมไบมีจำนวน 12.5 ล้านคน เมื่ออ้างอิงตามสำนักงานสถิติของอินเดีย แต่มีจำนวนถึง 20.5 ล้านคน หากยึดตามตัวเลขประมาณการของสหประชาชาติ ไม่ใช่หน้าที่ของชุมชน Wikidata ที่จะไปค้นหาความถูกต้องว่าแท้จริงเป็นอย่างไร Wikidata ใช้วิธีที่ง่ายกว่าคือเก็บข้อมูลนี้ไว้ทั้งหมดควบคู่ไปกับแหล่งที่มาของข้อมูลนั้น และเป็นหน้าที่ของผู้ใช้ข้อมูลเองที่จะเลือกหยิบชิ้นไหนไปใช้งาน ปัจจุบัน Wikidata มี 30 ล้านข้อเท็จจริงเกี่ยวกับ 14 ล้านหัวข้อ จะเห็นได้ว่าทั้ง DBpedia และ Wikidata เป็นการแปลงข้อมูลจาก Wikipedia มาเป็น structured data โดยใช้วิธีการที่แตกต่างกัน[30] อย่างไรก็ตาม Wikidata ก็ได้ถูกนำไปแปลงเป็นส่วนหนึ่งของ DBpedia ใน WikiData[31] และ ProFusion dataset[32]

[8] Freebase : Free Data : Free Download, Borrow and Streaming. Internet Archive. Available from: https://archive.org/details/freebase

[9] NNDB – Tracking the entire world. Available from: https://nndb.com/

[10] The Fashion Model Directory (FMD). Available from: https://www.fashionmodeldirectory.com/

[11] MusicBrainz – The Open Music Encyclopedia. Available from: https://musicbrainz.org/

[12] Wikidata. Available from: https://www.wikidata.org/

Cyc[13] เป็นโครงการสร้างฐานความรู้ขนาดใหญ่โดย Douglas B. Lenat ในปี ค.ศ. 1984[33] เป้าหมายเพื่อสร้างชุดข้อเท็จจริงจำนวนมากด้วยมือ ส่วน OpenCyc[14] เป็นเวอร์ชันย่อของ Cyc ที่ลดทอนขนาดของฐานความรู้และเปิดให้ใช้เป็นสาธารณะ[34] อย่างไรก็ตาม OpenCyc ได้ปิดตัวลงในปี ค.ศ. 2017 อย่างไรก็ตาม ResearchCyc[15] ได้เปิดให้บริการโดยไม่คิดค่าใช้จ่ายสำหรับการศึกษาค้นคว้าวิจัย แต่ในที่สุดก็หยุดพักการให้บริการในปี ค.ศ. 2019[35]

3. การแปลงข้อมูลจากรูปแบบอื่นมาเป็น RDF

RDF123 โดย Han และคนอื่น ๆ[36] เป็นเครื่องมือที่ใช้แปลงข้อมูลที่อยู่ในรูปแบบสเปรดชีตให้อยู่ในรูปแบบ RDF ซึ่งแนวคิดบางอย่างสามารถนำมาใช้ในการแปลงข้อมูลตารางมาเป็น RDF ได้ นอกจากนั้น RDB2RDF Survey Paper[37] ของ W3 Incubator Group กล่าวถึงงานวิจัยหลายโครงการที่เกี่ยวข้องกับการแปลงข้อมูลจากฐานข้อมูลเชิงสัมพันธ์ให้อยู่ในรูปแบบ RDF ถึงแม้ว่างานวิจัยเหล่านี้ไม่ได้กล่าวถึงการแปลงข้อมูลจากวิกิพีเดีย แต่ก็สามารถนำความรู้เหล่านั้นมาใช้ศึกษาการแปลงข้อมูลในรูปแบบตารางของวิกิพีเดียได้

นอกจากนี้ยังมี recommendation ของ W3C โดย CSV on the Web Working Group[16] อีกหลายฉบับที่กล่าวถึงการแปลงข้อมูล record set ในรูปแบบ CSV ไปสู่รูปแบบอื่นเช่น RDF หรือ JSON

งานวิจัยที่เกี่ยวข้องกับการแปลงข้อมูลตารางและรายการไปสู่รูปแบบอื่น

งานวิจัยที่เกี่ยวข้องกับการแปลงข้อมูลตารางและรายการไปสู่รูปแบบอื่น มีหลายวิธีการ เช่น

Yang และ Luk[38, 39] เสนอกระบวนวิธีในการแปลงตารางบนเว็บมาเป็นข้อมูลในลักษณะ key–value pair และหนทางแก้ไขปัญหาของการสกัดข้อมูลจากตารางในกรณีต่าง ๆ

งานวิจัยของ Pivk, Cimiano และ Sure[40] ใช้วิธีนำข้อมูลจากตารางบนเว็บแปลงให้อยู่ในรูปแบบ F-logic (frame logic) ซึ่งเป็นการรีพรีเซนต์เฟรมในรูปแบบหนึ่งซึ่งสามารถนำมาใช้ในซีแมนติกเว็บได้

[13] Cycorp Inc. Available from: https://www.cyc.com/

[14] OpenCyc. Cycorp. Available from: https://web.archive.org/web/20170227201513/http://www.opencyc.org/

[15] ResearchCyc. Cycorp. Available from: https://web.archive.org/web/20190407175843/http://www.cyc.com/researchcyc/

[16] CSV on the Web Working Group Wiki. W3C. Available from: https://www.w3.org/2013/csvw/wiki/Main_Page

TANGO (Table ANalysis for Generating Ontologies) เป็นงานวิจัยของ Embley[41], Tijerino และคนอื่นๆ[42, 43] เป้าหมายคือการแปลงข้อมูลตารางเข้ามาอยู่ในรูปแบบออนโทโลยีโดยตรง

TEGRA (Table Extraction by Global Record Alignment) เป็นงานวิจัยของ Chu และคนอื่นๆ[44] กล่าวถึงความท้าทายในการสกัดข้อมูลเชิงโครงสร้างจากตารางในเว็บเพจซึ่งในบางกรณี “ตาราง” ที่ปรากฏให้เห็นบนหน้าเว็บไม่ใช่ตารางในรูปแบบ HTML table แต่อาจจะอยู่ในรูปแบบของ HTML list หรือในลักษณะอื่น ๆ

DeExcelerator เป็นงานวิจัยของ Eberius และคนอื่น ๆ[45] เพื่อใช้เป็นเฟรมเวิร์คในการสกัดข้อมูลเชิงโครงสร้างจากตาราง HTML และสเปรดชีต

งานวิจัยของ Venetis และคนอื่น ๆ[46] แก้ปัญหาเรื่องการจัดการให้ซีแมนติกและออนโทโลยี โดยใช้วิธีเพิ่มเติมคลาสเข้าไปที่หัวสดมภ์ของตารางตรง ๆ ด้วยมือ ซึ่งสามารถแก้ปัญหาโดยไม่ต้องทำ schema matching เอง แต่ผู้ใช้ต้องมีความรู้ในการเพิ่มเติมข้อมูลเข้าไป

WebTables[47, 48] เป็นโครงการของ Google Research[17] เพื่อสกัดข้อมูลเชิงโครงสร้างจากตาราง HTML บนเว็บเพจ โดยทำการสืบค้นข้อมูลตาราง HTML จำนวน 14.1 พันล้านตาราง พบว่ามีเพียง 154 ล้านตารางเท่านั้นที่มีคุณภาพเพียงพอที่จะสกัดข้อมูลเชิงโครงสร้างมาใช้ได้[49] เนื่องจากตาราง HTML ส่วนใหญ่บนเว็บถูกนำไปใช้ในการกำหนดเลย์เอาต์ของหน้าเว็บแต่ไม่ได้ถูกนำไปใช้นำเสนอข้อมูลในรูปแบบตารางอย่างแท้จริง[50] WebTables ใช้วิธีเดียวกับงานวิจัยอื่นๆ ก่อนหน้านั้นในการใช้ตัวจำแนกที่ปรับแต่งให้เน้นไปที่ recall มากกว่า precision คัดกรองตารางจากเว็บเพจออกมาให้ได้มากที่สุดก่อน แล้วค่อยใช้วิธีคัดเลือกเฉพาะตารางที่มีเฮดเดอร์เพียงบรรทัดเดียวเท่านั้นเพื่อให้ง่ายในการสกัดข้อมูลเชิงโครงสร้างมาใช้ โดยตัดตารางอื่น ๆ ที่มีความซับซ้อนกว่านี้ทิ้งไปทั้งหมด หลังจากนั้นโครงการนี้ได้ถูกพัฒนาต่อเป็นระบบ Octopus[51] เพื่อช่วยในการสนับสนุนการสืบค้นข้อมูลบนเว็บได้อย่างมีประสิทธิภาพมากขึ้น

ที่ Google อีกเช่นกัน Elmeleegy, Madhavan และ Halevy [52] ได้นำ WebTables มาใช้ช่วยสนับสนุนระบบตัวอย่างที่ชื่อว่า ListExtract ในการสกัดข้อมูลจากรายการจำนวน 100,000 จากเว็บแล้วนำมาแปลงลงฐานข้อมูลเชิงสัมพันธ์ ส่วน Wong และคนอื่น ๆ [53] ทดลองใช้เครื่อง

[17] Google Tables. Google Research. Available from: https://research.google.com/tables

1,000 เครื่องสกัดข้อเท็จจริงจำนวน 10.1 พันล้านทูเปิลจาก 1 พันล้านเว็บเพจด้วยอัลกอริธึมแบบขนานโดยใช้เวลาไม่ถึง 6 ชั่วโมง

Fusion Tables[18] [54] เป็นโครงการของ Google Research[19] ออกแบบมาเพื่อให้ผู้ใช้สามารถอัพโหลดข้อมูลขึ้นไปบนเว็บเพื่อใช้ในการวิเคราะห์ข้อมูลด้วยเครื่องมือต่าง ๆ และในปัจจุบันยังสามารถเรียกใช้งานได้บน Google Docs[20] ได้ด้วย

โครงการ Web Data Commons[21] หรือ WDC [55] เป็นโครงการสกัดข้อมูลเชิงโครงสร้างจาก Common Crawl[22] ซึ่งเป็นคลังข้อมูลเว็บเพจที่มีขนาดใหญ่ที่สุดที่มีการเปิดให้ใช้เป็นสาธารณะ ส่วนหนึ่งของโครงการ WDC คือ Web Table Corpora[23] เป็นการสกัดเฉพาะข้อมูลเชิงโครงสร้างที่ได้มาจากตาราง HTML ที่อยู่ในเว็บเพจที่มีอยู่ใน Common Crawl ซึ่งจนถึงในปัจจุบันนี้ Web Table Corpora มีแจกจ่ายให้ดาวน์โหลดอยู่ 2 ชุด ชุดแรกคือ 2012 Corpus ซึ่งสกัด 147 ล้านตารางมาจาก 3.5 พันล้านเว็บเพจใน 2012 Common Crawl และชุดที่สองคือ 2015 Corpus ซึ่งสกัด 233 ล้านตารางมาจาก 1.78 พันล้านเว็บเพจใน July 2015 Common Crawl ซึ่งในชุดที่สองนี้มีเมตะเดต้าเกี่ยวกับทุกตารางเก็บไว้ให้ด้วย ในขณะที่ชุดแรกไม่มีเก็บไว้

WDC Web Table Corpus ได้ถูกนำไปใช้ประโยชน์มากมาย ยกตัวอย่างเช่น การนำไปใช้วัดประสิทธิภาพการทำ schema matching ในหลาย ๆ ระดับของ table (เช่น table-to-class, row-to-instance และ attribute-to-property matching) ด้วยวิธีการต่าง ๆ ที่ก่อนหน้านี้ใช้ dataset ที่แตกต่างกันทำให้การเปรียบเทียบเป็นไปได้ยาก[56]

งานวิจัยที่ใกล้เคียงกับข้อเสนอที่สุดคือ WikiTables[24] เป็นเครื่องมือในการดึงข้อมูลจากตารางในวิกิพีเดียมาใช้ในการค้นพบข้อเท็จจริงใหม่ๆ ที่ซ่อนอยู่ ผลผลิตที่ได้จากงานวิจัยนี้อีกชิ้นหนึ่งคือชุดข้อเท็จจริงที่สกัดได้จากตารางในวิกิพีเดียจำนวน 15 ล้านทูเปิล[57]

[18] FAQ: Google Fusion Tables. Google. Available from: https://support.google.com/fusiontables

[19] Google AI Blog: Google Fusion Tables. Google. Available from: https://ai.googleblog.com/2009/06/google-fusion-tables.html

[20] Google Docs. Available from: https://docs.google.com/

[21] Web Data Commons. Available from: http://webdatacommons.org/

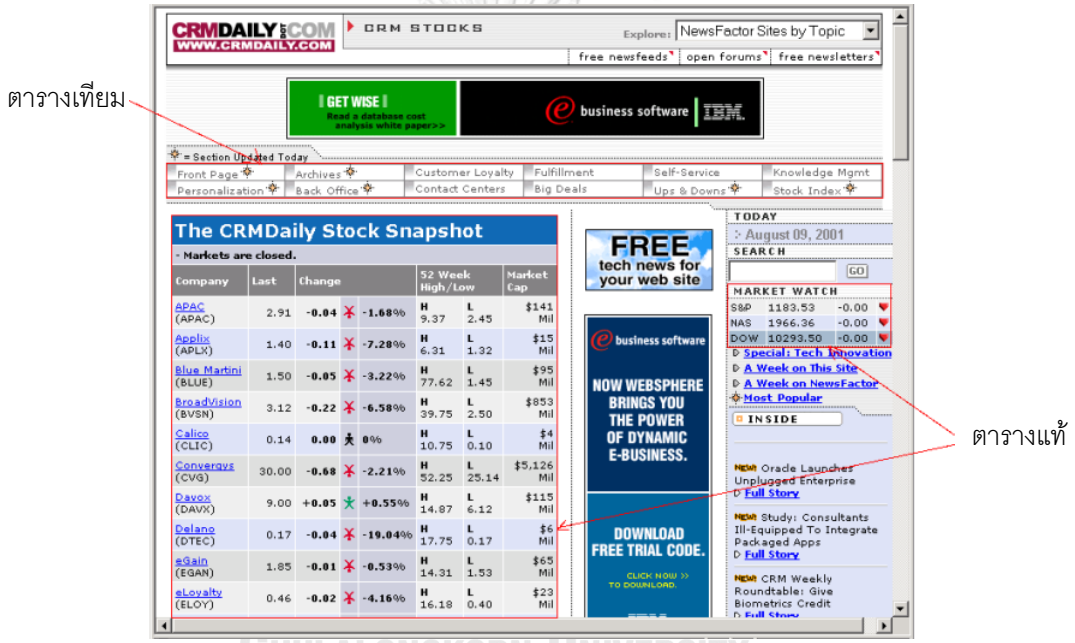
[22] Common Crawl. Available from: http://commoncrawl.org/

[23] Web Table Corpora. WDC. Available from: http://webdatacommons.org/webtables/

[24] WikiTables. Available from: http://downey-n1.cs.northwestern.edu/public/

1. <u>การใช้ machine learning เข้ามาร่วมด้วย</u>

งานวิจัยของ Galkin, Mouromtsev และ Auer[58], Wang, Phillips และ Haralick[59] มีการใช้เทคนิค machine learning วิธีการต่าง ๆ เข้ามาช่วยในการแก้ปัญหาการแยกแยะตารางในเว็บว่าตารางใดเป็น “ตารางแท้” คือตารางที่ใช้เก็บเนื้อหาข้อมูลจริง ๆ และ “ตารางเทียม” คือเป็นตารางเพื่อใช้เพียงการกำหนดรูปแบบเลย์เอาต์ของหน้าเพจ ดังตัวอย่างในภาพที่ 24 ซึ่ง Adelfio และ Samet[60] จะเรียกว่า relational table กับ non-relational table ตามลำดับ ตามหลักการที่กำหนดโดย Codd[7] ซึ่งจะเรียกแต่ละเซลล์ที่ header ว่า attributes และเรียกชุดข้อมูลที่อยู่ในแถวถัดมาว่าทูเปิล (tuples)



ภาพที่ 24 ตัวอย่างเปรียบเทียบตารางแท้และตารางเทียม[50]

Kushmerick, Weld และ Doorenbos[61, 62] พัฒนาเทคนิค wrapper induction ซึ่งเป็น supervised learning รูปแบบหนึ่งในการสร้าง wrapper แบบอัตโนมัติแทนการสร้าง wrapper ด้วยมือเพื่อใช้ในการสกัดข้อมูลจากเว็บเพจ Thamvijit และคนอื่น ๆ[63] นอกจากใช้เทคนิคของ Yang ในการสกัดข้อมูลที่เกี่ยวกับบุคคลจากตารางและรายการแล้ว ยังนำเทคนิค wrapper induction มาใช้เพื่อสกัดข้อมูลที่อยู่ในรูปแบบเรียงความอีกด้วย ส่วน Cohen, Hurst และ Jensen[64] พัฒนา wrapper learning system ที่ชื่อว่า WL^2 ในการสกัดข้อมูลตารางและรายการจากเว็บโดยเฉพาะ

Chen, Tsai และ Tsai[65] ใช้วิธีนับจำนวนเซลล์ที่มีความคล้ายกันในแง่ของความยาวตัวอักษรและชนิดของข้อมูลเมื่อเปรียบเทียบกับคอลัมน์ข้างเคียงในการสร้างกฎให้ตัวจำแนก ในขณะที่ Wu และคนอื่น ๆ[66] ใช้วิธีดูความคล้ายกันของตารางเพื่อนำมาใช้ในการสกัดข้อเท็จจริง ที่น่าสนใจคือการนำเอาเทคนิค dynamic time warping (DTW) ที่ส่วนใหญ่ใช้ในเรื่องที่เกี่ยวกับงานด้านรู้จำเสียงพูดมาประยุกต์ใช้ในการเปรียบเทียบตารางจากต้นไม้ DOM ในไฟล์ HTML

Wang และ Hu[50] พัฒนาฟีเจอร์ต่าง ๆ ขึ้นมาหลายตัวโดยดูจากโครงสร้างของตารางและความยาวรวมทั้งชนิดของข้อมูลในเซลล์ แล้วนำเข้าไปให้ decision tree (DT) และ support vector machine (SVM) เป็นตัวจำแนก ส่วนงานวิจัยของ Augenstein[67] ใช้ข้อเท็จจริงเดิมที่มีอยู่แล้วในลิงก์เดต้าเช่น Freebase และ Wikidata ในการเทรนตัว distant supervision[68] NER classifier[69] เพื่อให้ระบบเรียนรู้วิธีการที่จะสกัดความรู้ใหม่ ๆ เพิ่มเติมเข้ามาในฐานความรู้โดยอัตโนมัติ ในขณะที่ Adelfio และ Samet[60], WWT ของ Gupta และ Sarawagi[70], Hanifah และ Akbar[71], Pinto และคนอื่น ๆ[72] ใช้ conditional random fields (CRF) มาทำการจำแนกเพื่อช่วยแยกแยะสกีมาต่าง ๆ เช่น ส่วนประกอบ, คุณลักษณะ และชนิดของข้อมูล เป็นต้น

2. <u>การใช้ฐานข้อมูลขนาดใหญ่มาช่วยในการจำแนก</u>

งานของ Zhang และคนอื่น ๆ[73, 74] นำข้อมูลจาก DBpedia มาใช้เป็นฐานความรู้หลักเพื่อประยุกต์ใช้ในงานด้านวัสดุศาสตร์ (materials science) โดยมีการสกัดข้อมูลเพิ่มเติมจากตารางในวิกิพีเดียเนื่องจากพบว่าใน DBpedia ยังมีจำนวนข้อเท็จจริงไม่เพียงพอต่อการนำไปใช้ประกอบการวิเคราะห์ในรายละเอียด ในขณะที่ Limaye, Sarawagi และ Chakrabarti[75] ใช้ YAGO ในการกำหนดชื่อและชนิดของข้อมูลให้กับแต่ละคอลัมน์ของตาราง

งานวิจัยของ J. Wang และคนอื่น ๆ[76] ใช้ฐานข้อมูล taxonomy ชื่อว่า Probase[77, 78] ของ Microsoft Research[25] ในการสกัดข้อมูลตารางและหาความสัมพันธ์ของตาราง ที่น่าสนใจคือ Probase ที่มีขนาด 2.7 ล้านข้อเท็จจริงนั้นไม่ได้สร้างขึ้นด้วยมือแต่สร้างขึ้นโดยอัตโนมัติโดยใช้วิธีการเชิงสถิติ[79] และที่ไมโครซอฟท์เช่นกัน He, Ganjam และ Chu[80] ใช้ table corpus ขนาดใหญ่ที่สกัดมาจากเว็บจำนวน 100 ล้านตาราง หลังจากนั้น Wang และ He[81] ได้เพิ่มเติม table corpus ขนาดใหญ่อีกชุดหนึ่งเพิ่มเข้าไปอีกโดยสกัดมาจากสเปรดชีตภายในอินทราเน็ตของบริษัทขนาดใหญ่

[25] Probase. Microsoft Research. Available from: https://www.microsoft.com/en-us/research/project/probase/

แห่งหนึ่งมีขนาด 5 แสนตารางเพื่อนำมาใช้ในการสังเคราะห์หาความสัมพันธ์ระหว่างคู่ของข้อมูลเพื่อนำมาใช้ตรวจสอบความสมบูรณ์ของข้อมูลในตารางเป้าหมาย

3. บริการบนเว็บและผลิตภัณฑ์สำเร็จรูป

มีบริการบนเว็บหลายแห่งที่มีความสามารถในการดึงข้อมูลตารางที่ผู้ใช้ระบุ แล้วนำมาแปลงให้อยู่ในรูปแบบอื่น ๆ เช่น CSV และ XML เช่น Import.io[26] และ ScraperWiki[27] (ซึ่งตอนนี้เปลี่ยนชื่อมาเป็น QuickCode[28] ซึ่งนอกจากมีบริการสกัดข้อมูลจากเว็บแล้วยังเป็นแพลทฟอร์มในการพัฒนาระบบวิเคราะห์ข้อมูลบนเว็บด้วย Python และ R อีกด้วย)

นอกจากนี้ในปัจจุบันซอฟท์แวร์สำเร็จรูปต่าง ๆ หลายผลิตภันฑ์เริ่มมีการนำเอาคุณสมบัติการสกัดข้อมูลตารางจากเว็บมาใช้เพื่อนำเข้าข้อมูลมาใช้ในการวิเคราะห์ เช่น Microsoft Power BI[29], Power Query[82] หรือ Get & Transform[30] และ Tableau[31] รวมถึงสิทธิบัตรบางชิ้นของ IBM[83] ก็มีการนำเอาการสกัดข้อมูลตารางมาเพื่อเป็นข้อเท็จจริงประกอบการให้คำตอบของระบบถามตอบในโครงการที่ชื่อว่า DeepQA[32] เพื่อนำไปใช้กับระบบ Watson อีกด้วย

งานวิจัยที่เกี่ยวข้องกับการกำหนดซีแมนติกลงไปในเนื้อความตัวอักษร

KnowItAll[84, 85], TextRunner[86, 87], TIPSTER[33], ATLAS[88] และ GATE[34][89, 90] เป็นงานวิจัยเกี่ยวกับ natural language processing (NLP) ซึ่งสามารถนำมาใช้ในการสกัดและแปลงข้อมูลไร้โครงสร้างจากข้อความในตารางและรายการมาเป็นข้อมูลเชิงโครงสร้างในรูปแบบกราฟ มีความพยายามที่จะกำหนดมาตรฐานในการแลกเปลี่ยนข้อมูล NIF (NLP Interchange Format)[35][91] เป็นออนโทโลยีที่กำหนดโดยโครงการ NLP2RDF ของ University of Leipzig เพื่อรองรับการแลกเปลี่ยนข้อมูลระหว่างเครื่องมือทางด้าน NLP ต่าง ๆ มาเป็น RDF และลิงก์เดต้า

[26] Web Data Integration. Available from: https://www.import.io/

[27] ScraperWiki. Available from: https://scraperwiki.com/

[28] QuickCode. Available from: https://quickcode.io/

[29] Power BI – Microsoft Power Platform. Available from: https://powerbi.microsoft.com/

[30] Get & Transform in Excel. Microsoft. Available from: https://support.office.com/en-us/article/get-transform-in-excel-881c63c6-37c5-4ca2-b616-59e18d75b4de

[31] Tableau Business Intelligence and Analytics Software. Available from: https://www.tableau.com/

[32] The DeepQA Research Team. IBM. Available from: https://researcher.watson.ibm.com/researcher/view_group.php?id=2099

[33] TIPSTER Text Program. NIST. Available from: https://itl.nist.gov/iaui/894.02/related_projects/tipster/

[34] GATE – General Architecture for Text Engineering. Available from: https://gate.ac.uk/

[35] NIF 2.0 Core Ontology. Available from: https://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core/nif-core.html

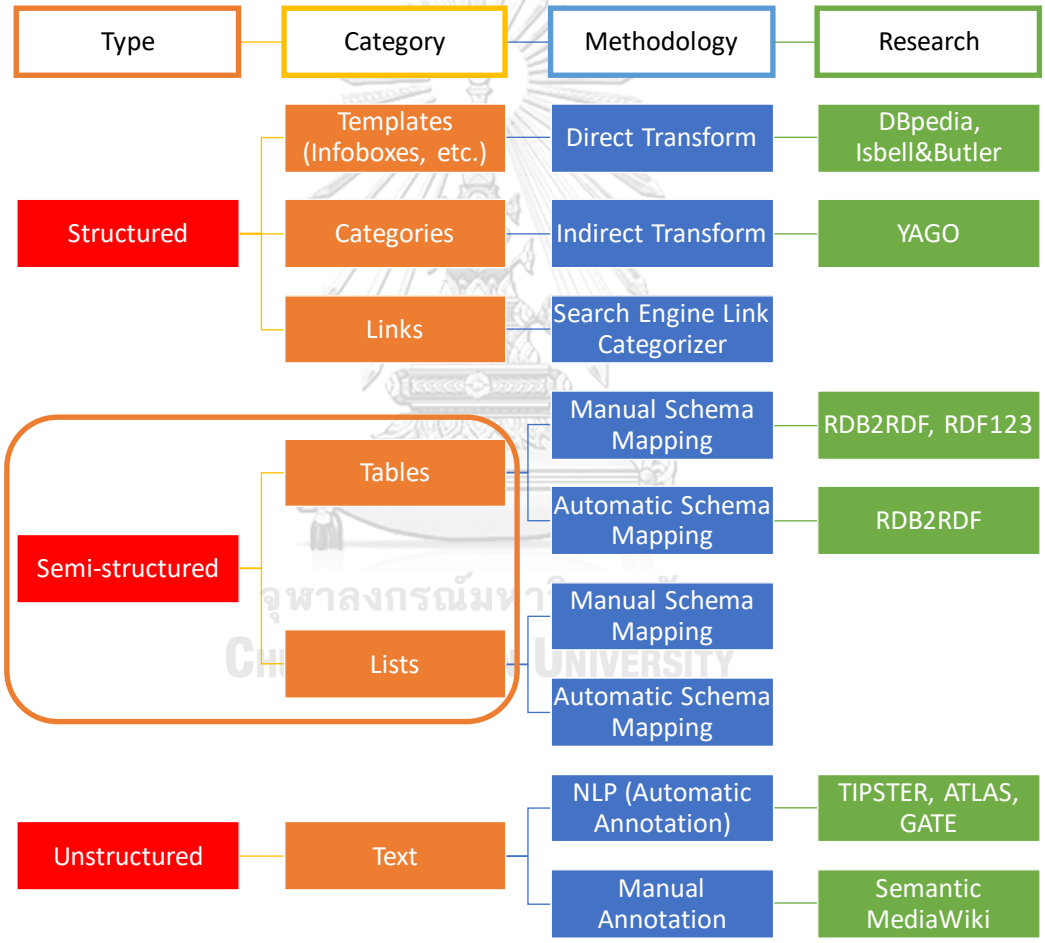
Semantic MediaWiki[36][92] เป็นส่วนเสริมของ MediaWiki ออกแบบมาเพื่อทำให้ผู้ใช้วิกิพีเดีย หรือผู้ใช้งานระบบอื่น ๆ ที่ทำงานบน MediaWiki สามารถแทรก “ความหมาย” ลงไปใน Wikitext ได้โดยตรงและสามารถนำข้อมูลซีแมนติกนั้นกลับมาในใช้รูปแบบ RDF[93] สิ่งที่น่าสนใจในงานวิจัยนี้คือวิธีการจัดการกับความไม่แน่นอนหรือความผิดพลาดของข้อมูล[94] เนื่องจากการที่ผู้ปรับปรุงข้อมูลซีแมนติกบางคนอาจจะเป็นผู้ใช้ทั่วไปที่ยังไม่มีความชำนาญในการกำหนดรูปแบบของข้อมูล Semantic MediaWiki จะพยายามปรับรูปแบบของข้อมูลที่บันทึกเข้าไปนั้นให้ถูกต้องเท่าที่จะเป็นไปได้ หรือไม่เช่นนั้นก็จะแสดงข้อความผิดพลาดขึ้นมาให้ผู้ใช้ทำการปรับแก้ไข

[36] Semantic MediaWiki (SMW). Available from: https://www.semantic-mediawiki.org/

บทที่ 4

การแปลงตารางและรายการเป็นข้อมูลเปิดห้าดาว

บทนี้จะกล่าวถึงแนวคิดและการออกแบบเดต้าโมเดลรวมถึงสเป็คของ vocabulary ตามมาตรฐาน RDFS เพื่อนำไปใช้ในการแปลงข้อมูล โดยขอบเขตของงานวิจัยนี้จะมุ่งเน้นไปที่ข้อมูลประเภท semi-structure ที่เป็นตารางและรายการ ดังแสดงในภาพที่ 25



ภาพที่ 25 ขอบเขตของข้อมูลที่จะสกัดมาใช้ในงานวิจัยนี้

การรีพรีเซนต์ข้อมูลในรูปแบบตารางและรายการ

การรีพรีเซนต์ข้อมูลในรูปแบบตารางและรายการสามารถทำได้หลากหลายรูปแบบที่ถูกกำหนดเป็นมาตรฐานโดยสถาบันต่าง ๆ เช่น

- International Organization for Standardization (ISO)[1] / International Electrotechnical Commission (IEC)[2]
- Internet Engineering Task Force (IETF)[3] Request for Comments (RFC)[4]
- World Wide Web Consortium (W3C)[5] Recommendations (REC)[6]
- Internet Assigned Numbers Authority (IANA)[7] Multipurpose Internet Mail Extensions (MIME) media types[8]

โดยจะยกตัวอย่างเฉพาะรูปแบบมาตรฐานที่เป็นที่รู้จักและใช้กันโดยทั่วไป และสามารถรองรับการรีพรีเซนต์ข้อมูลตารางและรายการได้ ดังนี้

- Delimiter-Separated Values (DSV) เช่น Comma-Separated Values (CSV) ตาม IETF RFC 4180 [95] และรูปแบบอื่น ๆ เช่น Tab-Separated Values (TSV)[9]
- Markup languages เช่น HTML table ตาม IETF RFC 1942 [96] (ซึ่งพัฒนามาจาก USDOD CALS ISG EPC[10] SGML Table Model[11]) และ HTML list ตาม IETF RFC 1866 [97]
- Lightweight markup languages เช่น Wikitext table[12] และ list[13] รวมไปถึง markdown languages ต่าง ๆ ที่ก่อนหน้านี้ไม่ได้มีกำหนดมาตรฐาน หลังจากนั้นจึงได้มี

[1] ISO – International Organization for Standardization. Available from: https://www.iso.org/

[2] IEC – International Electrotechnical Commission. Available from: https://www.iec.ch/

[3] IETF – Internet Engineering Task Force. Available from: https://www.ietf.org/

[4] RFCs. IETF. Available from: https://www.ietf.org/standards/rfcs/

[5] World Wide Web Consortium (W3C). Available from: https://www.w3.org/

[6] All Standards and Drafts. W3C. Available from: https://www.w3.org/TR/

[7] IANA – Internet Assigned Numbers Authority. Available from: https://www.iana.org/

[8] Media Types. IANA. Available from: https://www.iana.org/assignments/media-types/media-types.xhtml

[9] Tab-separated-values. IANA. Available from: https://www.iana.org/assignments/media-types/text/tab-separated-values

[10] United States Department of Defense; Continuous Acquisition and Life-cycle Support; Industry Steering Group; Electronic Publishing Committee.

[11] Table Models Specs/Documents. OASIS. Available from: https://www.oasis-open.org/specs/tablemodels.php

[12] Table. Wikipedia. Available from: https://en.wikipedia.org/wiki/Help:Table

[13] List. Wikipedia. Available from: https://en.wikipedia.org/wiki/Help:List

IETF RFC 7763 ให้กับรูปแบบหลัก และ RFC 7764 สำหรับรูปแบบย่อย ๆ ที่ดัดแปลงแตกแขนงออกมาจากรูปแบบดั้งเดิม

- Office document format ทั้ง OASIS OpenDocument Format (ODF) table และ list ตาม ISO/IEC 26300[14] ของ Open Office รวมไปถึง Microsoft Office Open XML (OOXML) table และ list ตาม ISO/IEC 29500[15] ที่แต่เดิมเป็นรูปแบบ proprietary ใน Microsoft Office ก่อนเวอร์ชัน 2007 นอกจากนี้ ISO/IEC ยังได้ออกรายงานฉบับหนึ่งที่เปรียบเทียบฟอร์แมททั้งสองรูปแบบนี้รวมทั้งแนวทางการแปลงข้อมูลระหว่างกันใน ISO/IEC TR 29166[16]

ความไม่ครบถ้วนในการสกัดข้อมูลเข้าสู่ลิงก์เดต้า

เมื่อเปรียบเทียบกับปริมาณข้อมูลทั้งหมดที่มีอยู่ในวิกิพีเดีย จะพบว่ามีข้อมูลเพียงแค่บางส่วนเท่านั้นที่ได้ถูกแปลงแล้วให้อยู่ในรูปแบบของลิงก์เดต้า เช่น abstract, Infoboxes และ categories ยังมีข้อมูลอีกเป็นจำนวนมากที่ยังไม่ได้ถูกสกัดและแปลงรูป ในงานวิจัยนี้ผู้วิจัยมุ่งเน้นไปที่เนื้อหาที่สำคัญที่สุดสองส่วนหลัก ๆ ในวิกิพีเดียคือ ตารางและรายการ

ขอเริ่มต้นด้วยการยกตัวอย่างการเข้าถึงข้อมูลที่ถูกแปลงให้อยู่ในรูปแบบลิงก์เดต้าอยู่แล้ว เช่น DBpedia หากเราต้องการสืบค้นข้อมูล “จุฬาลงกรณ์มหาวิทยาลัย” หรือ “*Chulalongkorn University*” ในฐานข้อมูล DBpedia เราอาจจะใช้คิวรี SPARQL เพื่อจะนำเอามาแค่ abstract จะเป็นดังนี้


```
PREFIX dbr: <http://dbpedia.org/resource/>
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT ?value
WHERE {
      dbr:Chulalongkorn_University dbo:abstract ?value.
}
```

ซึ่งผลลัพธ์ที่ได้จะออกมาแบบนี้

“*Chulalongkorn University, officially abbreviated as CU and commonly abbreviated as Chula, is the oldest university under the Thai modern educational*

[14] ISO/IEC 26300-1 – OpenDocument Schema. ISO. Available from: https://www.iso.org/standard/66363.html

[15] ISO/IEC 29500-1 – Fundamentals and Markup Language. ISO. Available from: https://www.iso.org/standard/71691.html

[16] ISO/IEC TR 29166:2011 – Guidelines for translation between ISO/IEC 26300 and ISO/IEC 29500 document formats. ISO. Available from: https://www.iso.org/standard/45245.html

system, founded in 1917 by King Vajiravudh (Rama VI) who named it after his father, King Chulalongkorn (Rama V). It is one of the best universities in Thailand and Southeast Asia according to several university rankings. It comprises nineteen faculties and institutes. Its campus occupies a vast area in downtown Bangkok. Diplomas were traditionally handed out at graduation by the King of Thailand, created and begun by King Prajadhipok (Rama VII). But at present, King Bhumibol Adulyadej (Rama IX) delegates the role to one of his daughters, Princess Maha Chakri Sirindhorn."

ซึ่ง DBpedia ได้ทำการเก็บ abstract มาทั้งย่อหน้าลงไปตรง ๆ แบบ full text ไม่มีโครงสร้าง สมมติว่าเราพบข้อความน่าสนใจเกี่ยวกับการจัดอันดับของมหาวิทยาลัย ซึ่งบอกว่า "จุฬาลงกรณ์มหาวิทยาลัยเป็นหนึ่งในมหาวิทยาลัยที่ดีที่สุดในประเทศไทยและเอเชียตะวันออกเฉียงใต้ จากการจัดอันดับมหาวิทยาลัยหลายแหล่ง" (*"It is one of the best universities in Thailand and Southeast Asia according to several university rankings."*) เราไม่อาจใช้คิวรี SPARQL ไปค้นหาข้อมูลจาก DBpedia เพิ่มเติมถึงอันดับเหล่านี้ได้ เนื่องจาก DBpedia ไม่มีข้อมูลเหล่านี้ถูกแปลงเก็บเอาไว้ เราจำเป็นต้องเข้าไปสืบค้นจากข้อมูลบทความต้นฉบับที่วิกิพีเดีย[17] ซึ่งจะพบตารางการจัดอันดับอยู่ดังตารางที่ 1

ตารางที่ 1 ตารางอันดับของจุฬาลงกรณ์มหาวิทยาลัยที่ปรากฏในวิกิพีเดีย

University World Ranking in 2005 – 2012 by QS World University Rankings

	2012	2011	2010	2009	2008	2007	2006	2005
Overall	201	171	180	138	166	223	161	161
Arts & Humanities	113	69	78	49	119	136	–	–
Natural Sciences	140	138	186	136	171	159	–	–
Engineering & IT	106	104	101	78	86	100	–	–
Social Sciences	80	68	78	51	72	83	–	–
Life Sciences	92	78	130	51	108	138	–	–

[17] Chulalongkorn University. Wikipedia. Available: https://en.wikipedia.org/wiki/Chulalongkorn_University?oldid=736429624

หรือหากเราพบข้อความน่าสนใจเรื่องโครงสร้างหน่วยงานของมหาวิทยาลัย เช่น “ประกอบไปด้วย 19 คณะและสถาบันต่าง ๆ” (*“It comprises nineteen faculties and institutes.”*) เราก็จะไม่สามารถสืบค้นจาก DBpedia ได้เช่นกัน แต่เราจะพบว่ามีข้อมูลนี้เป็นรายการแบบหลายลำดับขั้นอยู่ในวิกิพีเดียดังนี้

- List of faculties and institutes
    - Health Sciences
        - Faculty of Allied Health Sciences
        - Faculty of Dentistry
        - Faculty of Medicine
        - Faculty of Nursing
        - Faculty of Pharmaceutical Sciences
        - Faculty of Psychology
        - Faculty of Sport Sciences
        - Faculty of Veterinary Science
    - Sciences and Technologies
        - Faculty of Architecture
        - Faculty of Engineering
        - Faculty of Science
    - Social Sciences and Humanities
        - Faculty of Arts
        - Chulalongkorn Business School
        - Faculty of Communication Arts
        - Faculty of Economics
        - Faculty of Education
        - Faculty of Fine and Applied Arts
        - Faculty of Law
        - Faculty of Political Science

ซึ่งหมายความว่ายังมีข้อมูลอีกหลายส่วนที่ไม่ได้อยู่ในรูปแบบของ RDF หรือไม่ได้อยู่ในรูปแบบของลิงก์เดต้าหรือ LOD cloud มาก่อน ดังนั้นจึงเป็นเรื่องน่าสนใจหากเราสามารถหาทางนำข้อมูลตารางและรายการเหล่านี้มาแปลงให้อยู่ในรูปแบบ RDF เราจะสามารถใช้คิวรี SPARQL ในการสืบค้นข้อมูลเพิ่มเติมได้อีกมากมาย

เพื่อที่จะจัดเตรียมโครงสร้างที่เหมาะสมที่จะจัดเก็บข้อมูลตารางและรายการที่มาจากวิกิพีเดีย ผู้วิจัยจำเป็นต้องออกแบบและจัดทำสกีมา (schema) เพื่อให้มี vocabulary เพิ่มเติมสำหรับ

สร้าง RDF ที่ได้จากการสกัดข้อมูล (การทดลองสกัดข้อมูลตารางและรายการเพื่อแปลงออกมาเป็น RDF ทริปเปิลและตัวอย่างผลลัพธ์ที่ได้ แสดงอยู่ใน[98])

การสร้างรายการด้วย RDF Schema

เพื่อให้เข้าใจการสร้างรายการด้วย RDF Schema ได้ง่ายขึ้น เราจะมาลองสร้าง RDF ทริปเปิล โดยใช้ RDF Schema เพื่อสร้างรายการง่าย ๆ ตามตัวอย่างดังต่อไปนี้

- List Item 1
- List Item 2
  - List Item 2,1
  - List Item 2,2
    - List Item 2,2,1
- List Item 3

ซึ่งตัวอย่าง RDF ทริปเปิล ที่ใช้ในการรีพรีเซนต์รายการข้างต้นโดยใช้ RDF Schema เป็นดังรายการที่ 1 โดยต่อไปนี้เราจะใช้ Turtle เป็นหลักในการอธิบาย ซึ่ง Turtle ชุดนี้แสดงให้เห็นว่ารีซอร์ส ex:ListExample ที่สร้างจากรายการตัวอย่างนี้มี tlp:member อยู่หลายตัวชี้ไปที่โหนดของไอเท็มของรายการ โดยที่แต่ละโหนดของ tlp:member ก็มี tlp:index ใส่กำกับไว้ด้วย ในขณะเดียวกันที่โหนดของไอเท็มบางโหนดก็ยังมี tlp:member ชี้ต่อลงไปยังโหนดของไอเท็มที่ลึกลงไปอีก นี่เป็น RDF Schema ที่ผู้วิจัยออกแบบขึ้นมาแบบโครงสร้างต้นไม้ ซึ่งเป็นโมเดลแบบแรกที่เรียกว่า Index Member หรือ IM model

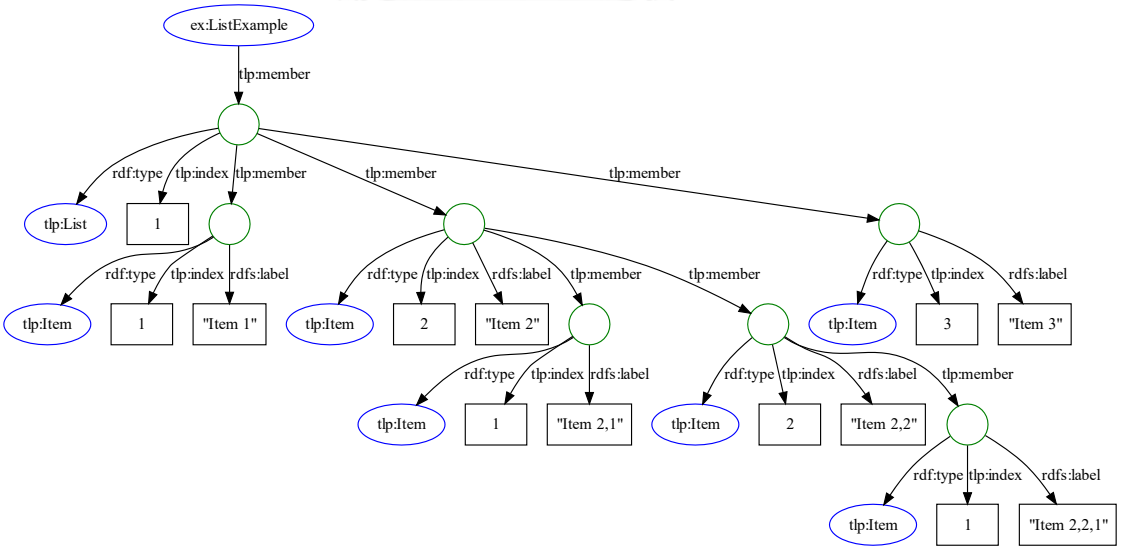

```
@prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix ex:   <http://example.org/sample#> .
@prefix tlp:  <http://purl.org/tulip/ns#> .

ex:ListExample
    tlp:member _:List1 .
_:List1 rdf:type tlp:List ;
    tlp:index 1 ;
    tlp:member _:Item1, _:Item2, _:Item3 .
_:Item1 rdf:type tlp:Item ;
    tlp:index 1 ;
    rdfs:label "List Item 1" .
_:Item2 rdf:type tlp:Item ;
    tlp:index 2 ;
    rdfs:label "List Item 2" ;
    tlp:member _:Item21, _:Item22 .
_:Item21 rdf:type tlp:Item ;
```

```
    tlp:index 1 ;
    rdfs:label "List Item 2,1" .
_:Item22 rdf:type tlp:Item ;
    tlp:index 2 ;
    rdfs:label "List Item 2,2" ;
    tlp:member _:Item221 .
_:Item221 rdf:type tlp:Item ;
    tlp:index 1 ;
    rdfs:label "List Item 2,2,1" .
_:Item3 rdf:type tlp:Item ;
    tlp:index 3 ;
    rdfs:label "List Item 3" .
```

รายการที่ 1 RDF ทริปเปิลของรายการตัวอย่างโดยใช้ RDF Schema

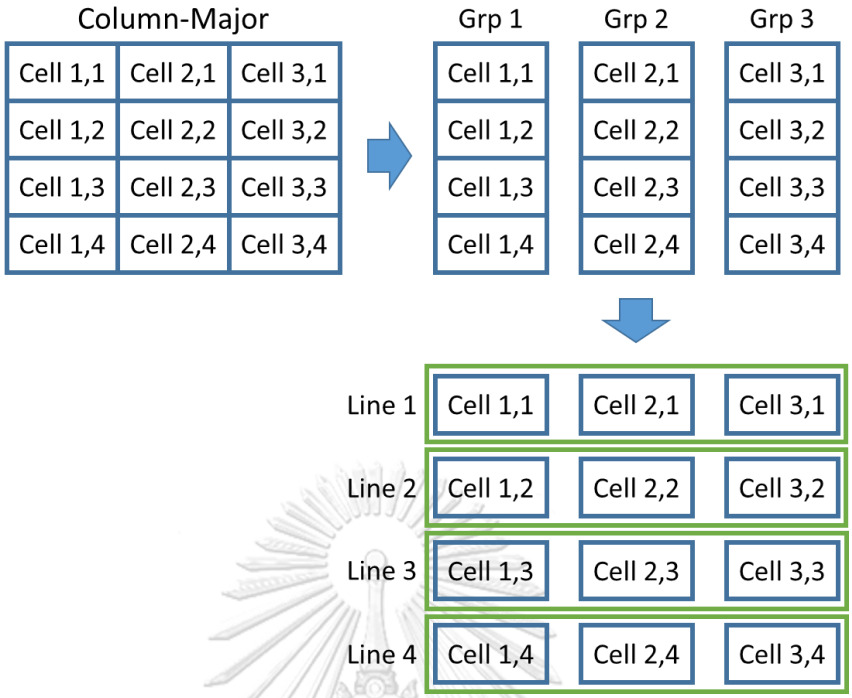
ลองนำ RDF ทริปเปิล ชุดนี้มาแสดงในรูปแบบ RDF กราฟ จะได้ดังภาพที่ 26 คือโหนดแรกที่เป็นรีซอร์สโหนด จะมี tlp:member เป็น blank node หนึ่งโหนด ดังที่กล่าวไปแล้วว่าโหนดของ RDF สามารถกำหนดให้เป็น blank node ได้ เราจึงกำหนดไปเลยว่าข้อมูลชุดนี้เป็นข้อมูลชุดเดียวกันภายใต้ blank node เดียวกันนี้ ซึ่งแท้จริงแล้วข้อมูลเหล่านี้จะเป็น RDF ทริปเปิล แยกออกจากกัน คือข้อมูลจะประกอบไปด้วยแต่ละบรรทัดของ RDF ทริปเปิล ซึ่งจะมีบรรทัดของ tlp:index และจะมีบรรทัดที่แสดงถึง tlp:member แต่ละตัวด้วย เช่น ข้อมูลชุดนี้มี tlp:member กำหนดเลยว่าโหนดอะไรเป็น member ของโหนดไหน และ index มันคือหมายเลขอะไร การแทนข้อมูลในลักษณะนี้ทำให้เราสามารถแปลงย้อนไปหรือย้อนกลับมาสู่รูปแบบตามข้อมูลต้นฉบับได้ ทำให้เราเก็บโครงสร้างของรายการมาอยู่ในรูปแบบของ RDF กราฟได้อย่างครบถ้วน



ภาพที่ 26 RDF กราฟของรายการตัวอย่างโดยใช้ RDF Schema

Column-Major และ Row-Major

โดยแท้จริงแล้วตารางก็จะมีลักษณะที่คล้ายกันกับรายการ แต่การรีพรีเซนต์ตารางจะมีเทคนิคเพิ่มเติม เนื่องจากรายการถึงแม้ว่าอาจจะมีหลายชั้น แต่ทุกชั้นก็เป็นเพียงแค่ 1 มิติจึงมีความตรงไปตรงมา ในขณะที่ตารางมีลักษณะเป็น 2 มิติ เราจึงจำเป็นต้องจัดการมันก่อน โดยที่จะต้องกำหนดลงไปว่า เราจะรีพรีเซนต์มันในแบบไหน ยกตัวอย่างเช่น HTML เป็น row based ก็จะเริ่มต้นด้วยแท็ก <tr> แล้วต่อด้วย <td> หรือ <th> เพื่อเป็นการกำหนดข้อมูลแต่ละเซลล์ ซึ่งในระยะหลายปีมานี้เริ่มมีแนวคิดหนึ่งที่ได้รับความนิยมเรียกว่า column orient หรือแบบ columnar เช่น columnar database ซึ่งก็จะมีข้อได้เปรียบในหลายจุด ก่อนหน้านี้ฐานข้อมูลเชิงสัมพันธ์หรือ relational database ที่ใช้ SQL จะเก็บข้อมูลเป็น row หรือเก็บเป็น record ข้อมูลของชุดข้อมูลเดียวกัน ไม่ว่าจะเป็น attributes อะไรก็ตามที่อยู่ในชุดเดียวกัน ก็จะเก็บอยู่ใน record เดียวกัน แต่แนวคิดใหม่ของ columnar database (ซึ่งเป็นประเภทหนึ่งของกลุ่มฐานข้อมูลแนวใหม่ที่เรียกว่า NoSQL คือไม่ได้เก็บข้อมูลในลักษณะของฐานข้อมูลเชิงสัมพันธ์) เป็นการเก็บข้อมูลเชิง column แทน โดยการเก็บข้อมูลที่เป็น attributes เดียวกันให้อยู่ในเซ็ตเดียวกัน ดังนั้นไม่ว่าข้อมูลจะมีกี่ record ก็ตาม ซึ่งมีแต่ละฟิลด์เป็นข้อมูลหลายประเภทแตกต่างกันไป ข้อมูลแต่ละฟิลด์ก็จะอยู่ในชุดเดียวกันทั้งหมด ซึ่งด้วยแนวคิดใหม่นี้จะมีข้อดีหลายอย่าง เช่น สามารถ optimize ได้สะดวก compress ได้ง่ายขึ้น เพราะข้อมูลเป็นชนิดเดียวกันในแต่ละเซ็ตของข้อมูล และยังสามารถทำ sorting ได้อย่างรวดเร็ว โดยรวมคือสามารถบริหารจัดการข้อมูลได้ในแบบที่สะดวกกว่าเดิมสำหรับข้อมูลบางประเภท ดังนั้นผู้วิจัยจึงออกแบบ data model ให้สามารถรองรับได้ทั้งคู่ นอกเหนือไปจาก row based แบบดั้งเดิมแล้ว ยังสามารถเก็บข้อมูลในลักษณะ column based ได้ด้วย โดยเราสามารถกำหนดลงไปได้ว่าต้องการเก็บแบบไหน เนื่องจากโครงสร้างมีความยืดหยุ่น จะเป็น column-major หรือเป็น row-major ก็ได้ แต่ในที่นี้จะอธิบายแยกให้เห็นว่าเราจะเก็บ column-major แบบไหนและเก็บ row-major อย่างไร

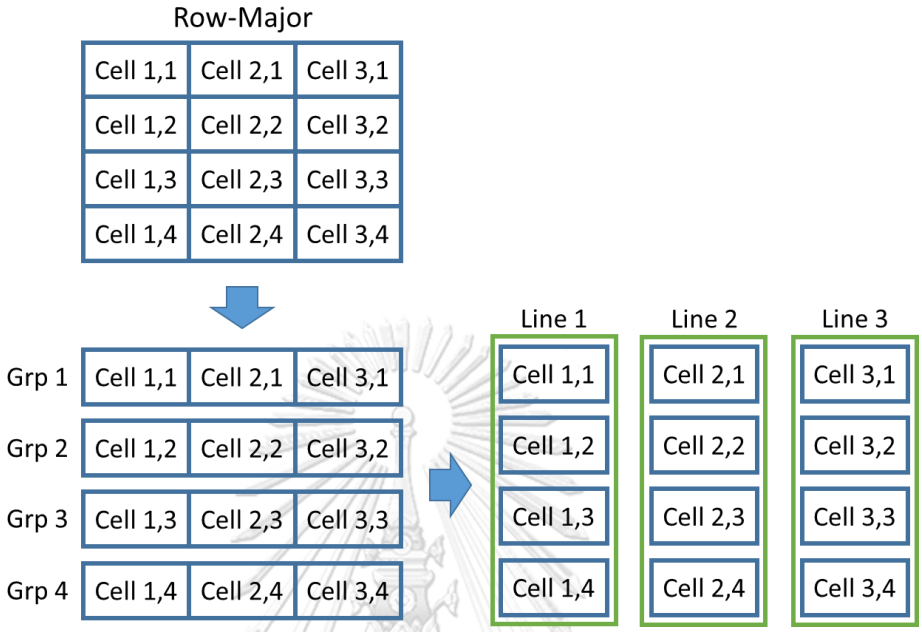


ภาพที่ 27 การแบ่งเซลล์ข้อมูลออกเป็นสดมภ์ก่อนแล้วจึงแบ่งแถวในลักษณะ column-major

ภาพที่ 27 แสดงให้เห็นการแบ่งเซลล์ข้อมูลในลักษณะ column-major คือการแยกตารางให้แบ่งเป็นกลุ่มตามสดมภ์ก่อน หลังจากนั้นพอแบ่งกลุ่มตามสดมภ์เสร็จเรียบร้อย ก็จะได้ชุดของเซลล์แต่ละสดมภ์ หากเราจะอ้างอิงแถว เราก็จะมองให้เป็นเสมือน virtual row โดยการเอาข้อมูลจากตำแหน่งเดียวกันของแต่ละกลุ่ม ก็คืออันดับเดียวกันของแต่ละสดมภ์มาไว้ด้วยกันเพื่อเป็นข้อมูลแถว หมายความว่าหากเรามีวิธีเก็บแบบนี้แล้ว เราสามารถจะจับเซลล์มาผสมใหม่อย่างไรก็ได้ เพียงแต่ในตัวอย่างที่แสดงให้เห็นนี้ เราแบ่งแยกเก็บตามสดมภ์ก่อน

แต่อย่างที่กล่าวไว้ข้างต้น เราสามารถที่จะเก็บเรียงตามแถวก่อนก็ได้เช่นเดียวกัน เพราะว่าใน RDF Schema ที่ออกแบบไว้จะมีพร็อปเพอร์ตี้ที่สามารถกำหนดได้ว่าตอนนี้ข้อมูลที่เราเก็บเป็น column-major หรือเก็บเป็น row-major ดังนั้นเราก็จะสามารถพลิกแกนไปมาได้ ซึ่งจะขยายความเรื่องนี้ในลำดับต่อไปเรื่องคุณสมบัติพิเศษของ Schema ที่ผู้วิจัยออกแบบสร้างขึ้น โดยในภาพที่ 28 แสดงให้เห็นว่าเมื่อเป็น row-major เราจะแบ่งเป็นแถวแนวนอนก่อน คือเป็น row ซึ่งแบบนี้ก็จะมีความคล้ายกับตารางใน HTML หรือฐานข้อมูลเชิงสัมพันธ์ คือเก็บเป็น records หลังจากนั้นก็ทำนองเดียวกัน หากเราจะเอาแต่ละกลุ่มมาดูเฉพาะบางสดมภ์ หรือในกรณีของฐานข้อมูลเชิงสัมพันธ์ คือ attribute เราก็จับตำแหน่งเดียวกันในแต่ละกลุ่ม ซึ่งจะได้เป็น line ที่เป็นการรวมสดมภ์ออกมา โดยในกรณีนี้ ผู้วิจัยใช้คำว่า group กับ line ไม่ใช่ column กับ row เพื่อป้องกันไม่ให้เกิดการสับสน

เพราะเราสามารถกำหนดให้ group และ line เป็น column และ row หรือสลับกันก็ได้ ขึ้นอยู่กับการเลือกเก็บข้อมูลเป็น column-major หรือ row-major



ภาพที่ 28 การแบ่งเซลล์ข้อมูลออกเป็นแถวก่อนแล้วจึงแบ่งเป็นสดมภ์ในลักษณะ row-major

การสร้างตารางด้วย RDF Schema

เราจะอธิบายการสร้างตารางด้วย RDF Schema ด้วยการทดลองสร้างตารางเล็ก ๆ ขนาด 3 column x 3 row โดยใช้ตัวอย่างดังนี้

Cell Content 1,1	Cell Content 2,1	Cell Content 3,1
Cell Content 1,2	Cell Content 2,2	Cell Content 3,2
Cell Content 1,3	Cell Content 2,3	Cell Content 3,3

เนื่องจาก RDF Schema ที่ผู้วิจัยออกแบบขึ้น สามารถรีพลีเซนต์ตารางได้ทั้งแบบ column-major (หมายถึงจะอ้างอิงด้วยสดมภ์ก่อนจึงตามด้วยแถว) และแบบ row-major (คืออ้างอิงด้วยแถวก่อนแล้วตามด้วยสดมภ์) แต่ในกรณีนี้จะยกตัวอย่างด้วยการรีพรีเซนต์ตารางด้วย RDF Schema แบบ column-major ดังนั้นข้อมูลตัวอย่างในเซลล์ของตารางจึงนำหน้าด้วยเลข column แล้วตามด้วยเลข row สอดคล้องกันเพื่อให้เข้าใจได้โดยง่าย

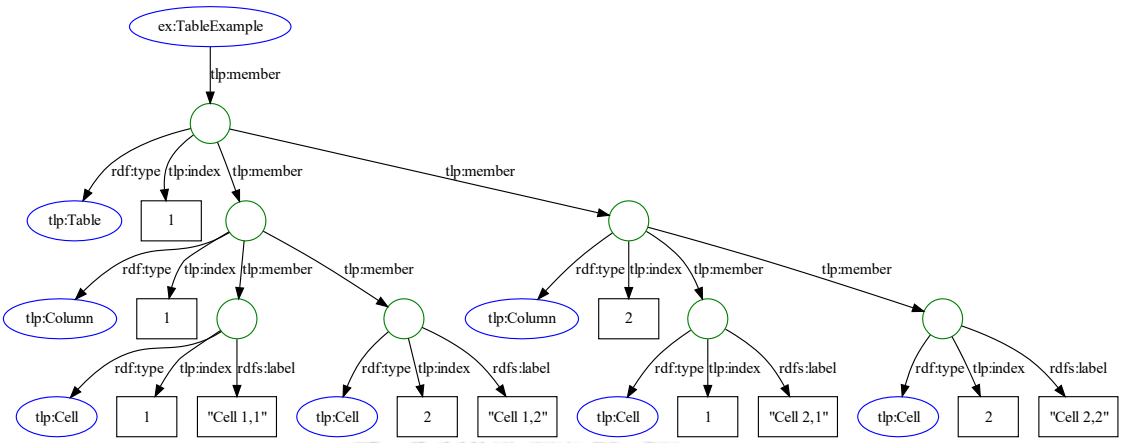
ตัวอย่าง RDF ทริปเปิล ที่ใช้ในการรีพรีเซนต์ตารางขนาด 3 x 3 ข้างต้นโดยใช้ RDF Schema ที่ผู้วิจัยออกแบบเป็นดังรายการที่ 2 และเช่นเดียวกันกับการเก็บรายการ การเก็บตารางเราก็มี tlp:member เหมือนกัน และจากที่เราจัดเรียงข้อมูลตาม column-major เราก็ใส่ index ลงไปตามลำดับชั้นเริ่มจากสดมภ์แรกก่อน แล้วจึงลงมาที่เซลล์ของแต่ละแถวในสดมภ์เดียวกัน

```
@prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix ex:   <http://example.org/sample#> .
@prefix tlp:  <http://purl.org/tulip/ns#> .

ex:TableExample
    tlp:member _:Table1 .
_:Table1 rdf:type tlp:Table ;
    tlp:index 1 ;
    tlp:member _:Col1, _:Col2, _:Col3 .
_:Col1 rdf:type tlp:Column ;
    tlp:index 1 ;
    tlp:member _:Cell11, _:Cell12, _:Cell13 .
_:Cell11 rdf:type tlp:Cell ;
    tlp:index 1 ;
    rdfs:label "Cell Content 1,1" .
_:Cell12 rdf:type tlp:Cell ;
    tlp:index 2 ;
    rdfs:label "Cell Content 1,2" .
_:Cell13 rdf:type tlp:Cell ;
    tlp:index 3 ;
    rdfs:label "Cell Content 1,3" .
_:Col2 rdf:type tlp:Column ;
    tlp:index 2 ;
    tlp:member _:Cell21, _:Cell22, _:Cell23 .
_:Cell21 rdf:type tlp:Cell ;
    tlp:index 1 ;
    rdfs:label "Cell Content 2,1" .
_:Cell22 rdf:type tlp:Cell ;
    tlp:index 2 ;
    rdfs:label "Cell Content 2,2" .
_:Cell23 rdf:type tlp:Cell ;
    tlp:index 3 ;
    rdfs:label "Cell Content 2,3" .
_:Col3 rdf:type tlp:Column ;
    tlp:index 3 ;
    tlp:member _:Cell31, _:Cell32, _:Cell33 .
_:Cell31 rdf:type tlp:Cell ;
    tlp:index 1 ;
    rdfs:label "Cell Content 3,1" .
_:Cell32 rdf:type tlp:Cell ;
    tlp:index 2 ;
    rdfs:label "Cell Content 3,2" .
_:Cell33 rdf:type tlp:Cell ;
    tlp:index 3 ;
    rdfs:label "Cell Content 3,3" .
```

รายการที่ 2 RDF ทริปเปิลของตารางตัวอย่างโดยใช้ RDF Schema

ลองนำ RDF ทริปเปิล ชุดนี้มาแสดงในรูปแบบ RDF กราฟ จะได้ดังภาพที่ 29 (ขอปรับตารางให้เป็นขนาด 2 column x 2 row เพื่อให้กราฟกระชับขึ้น)



ภาพที่ 29 RDF กราฟของตารางตัวอย่างโดยใช้ RDF Schema

จะสังเกตได้ว่าหากตารางมีขนาดใหญ่มาก และในบางกรณีอาจมีการซ้อนกันของข้อมูล เช่น มีรายการซ้อนอยู่ในเซลล์ของตาราง หรือมีตารางเป็นไอเท็มย่อยในรายการ โครงสร้างกราฟนี้ก็จะกว้างและลึกมาก ซึ่งแท้จริงแล้วตรงนี้ก็ไม่ใช่เรื่องแปลกสำหรับ RDF กราฟ เพราะว่าหากสังเกตดูจาก RDF กราฟของ LOD cloud ทั้งหมด จะเป็นระดับแสนล้าน RDF ทริปเปิล เราทราบดีอยู่แล้วว่าถึงแม้รีซอร์สจะมีขนาดไม่ใหญ่มาก แต่กราฟของ RDF ก็มีโอกาสที่จะลึกอยู่แล้ว ซึ่งก็เป็นเรื่องปกติของลิงก์เดต้า ประเด็นสำคัญคือการคิวรีกราฟที่มีความลึกจะมีความยุ่งยากซับซ้อน โครงสร้าง RDF Schema แบบ IM model แบบแรกนี้จึงมีทั้งข้อดีและข้อเสีย ข้อดีก็คือมันเป็นลำดับชั้นชัดเจน เราสามารถใช้แอปพลิเคชันของซีแมนติกเว็บทั่วไปที่รองรับโครงสร้างข้อมูล RDF ในการเข้าถึงข้อมูลแต่ละสดมภ์แต่ละแถว หรือแต่ละเซลล์ของตารางชุดนี้ในแบบปกติทั่วไปได้เลย แต่ด้วยข้อเสียดังที่กล่าวไปข้างต้น ผู้วิจัยจึงออกแบบ RDF Schema อีกโมเดลหนึ่ง โดยการกำหนดโครงสร้างให้เป็นเสมือน multi-dimensional array คือเป็นอาร์เรย์แบบหลายมิติ

การแบ่งเนื้อหาของเอกสารทั่วไปให้อยู่ในรูปแบบของอาร์เรย์แบบหลายมิติ

ก่อนจะกล่าวถึง RDF Schema ในโมเดลแบบที่ 2 ขอแสดงถึงแนวคิดในการแบ่งเนื้อหาของเอกสารโดยทั่วไปให้อยู่ในรูปแบบของอาร์เรย์แบบหลายมิติ และการกำหนด multi-dimensional array index ไปที่แต่ละส่วนย่อยของเนื้อหาเหล่านั้น จะขออธิบายด้วยเอกสารตัวอย่างตามภาพที่ 30

Article title / header

Section 1 header

Section 1 paragraph - Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam efficitur consectetur metus at consequat. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae.

Section 2 header

Section 2 paragraph 1 - Quisque hendrerit libero eu placerat ultricies. Mauris eros lectus, commodo ut mollis quis, tempor eu nibh.

Section 2 paragraph 2 - Donec venenatis euismod sem vestibulum pellentesque. Vestibulum feugiat lacus purus, eget malesuada urna.

Section 3 header

- Section 3 list item 1 - Proin luctus sodales nulla vel auctor.
- Section 3 list item 2 - Integer aliquam, turpis posuere porta.
  - Section 3 list item 2.1 - Risus erat auctor massa, ut commo.
  - Section 3 list item 2.2 - Nisl in nisi. Sed ultricies lorem Biben.
- Section 3 list item 3 - Lorem volutpat sollicitudin. Nam ac tempus elit.
- Section 3 list item 4 - Sit amet volutpat ex.

Section 4 header

	Lorem ipsum	**Dolor sitame**	**Consectetur**	**Adipiscing eli**
#01	Fusce tincidu	arcu vitae jus	Vestibulum	consectetur
#02	elit tempus	Suspendisse	luctus auctor	Egestas
#03	Pellentesque	habitant mor	tristique	senectus
#04	et netus	et malesuada	fames turpis	Cras nec

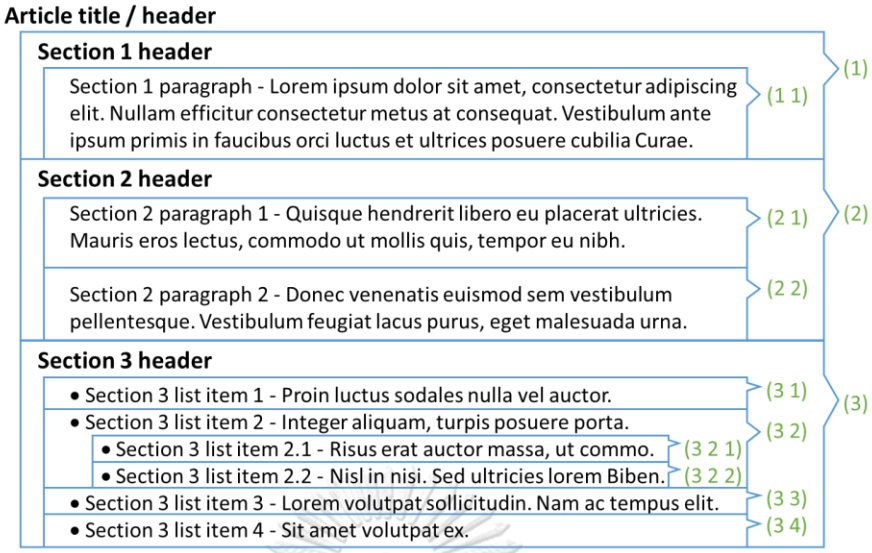
Table 4.1 Title - Praesent vehicula dolor ut tristique commodo.

First	Lorem ipsum	dolor sit ame	Consectetur
Second	Fusce tincidu	arcu vitae jus	Vestibulum
Third	elit tempus	Suspendisse	luctus auctor
Forth	Pellentesque	habitant mor	tristique

Table 4.2 Title - Praesent vehicula dolor ut tristique commodo.

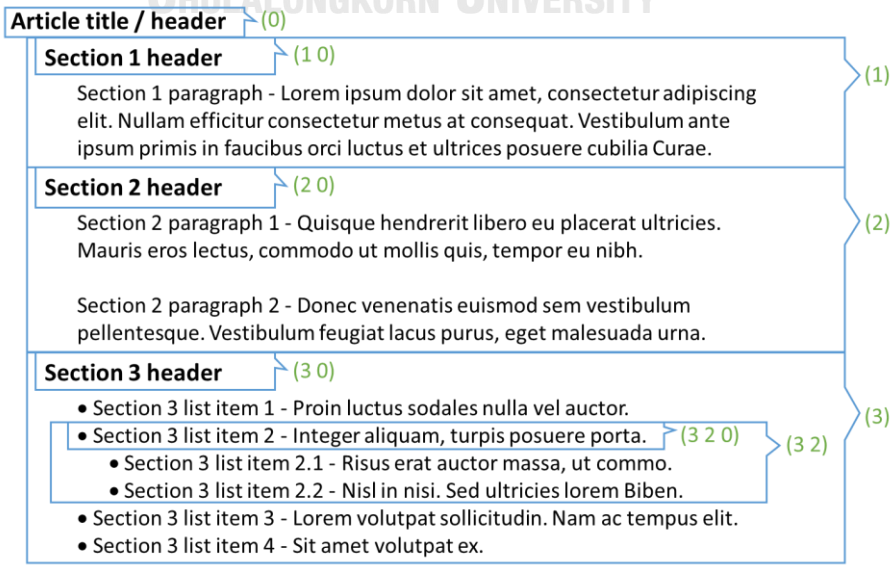
ภาพที่ 30 เอกสารตัวอย่างบทความที่ประกอบด้วยรายการและตาราง

เราทำการกำหนดลงไปว่าอาร์เรย์มิติแรกคือแบ่งตาม section และมิติที่สองย่อยลงมาอาจจะเป็นย่อหน้า, รายการ, ตาราง หรือเนื้อหารูปแบบอื่น ๆ แล้วถัดจากนั้นก็มิติที่สามย่อยลงไปอีกก็จะได้หมายเลขระบุ multi-dimensional array index เรียงลำดับลงไปตาม element ย่อย ตามในภาพที่ 31 ซึ่งหมายเลข index เหล่านี้แสดงให้เห็นในรูปแบบของ RDF list ในลักษณะของ S-expression คือครอบด้วยเครื่องหมายวงเล็บเปิดและวงเล็บปิด และแยกตัวเลข index แต่ละมิติออกจากกันด้วยช่องว่าง

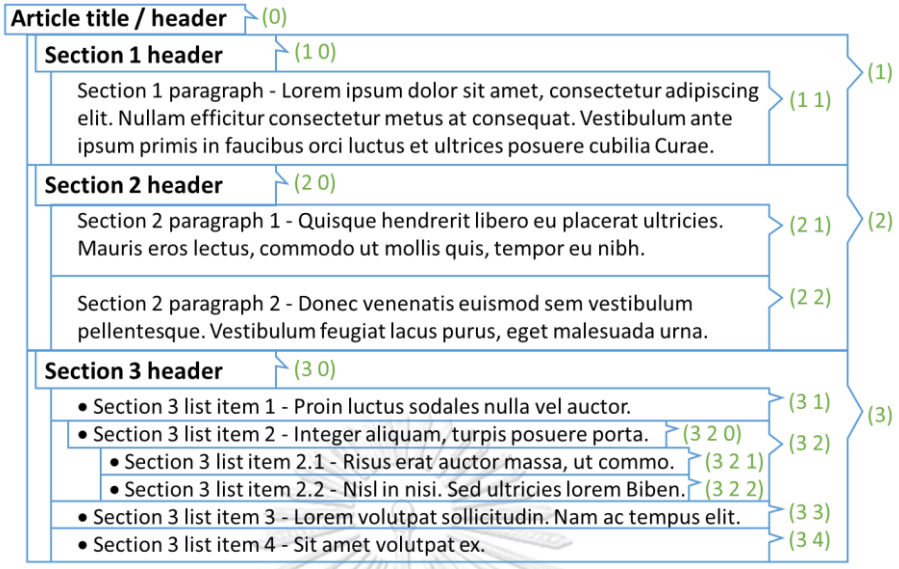


ภาพที่ 31 การกำหนด multi-dimensional array index ให้กับย่อหน้าและรายการ

ในขณะเดียวกันถ้าเราใส่เลข 0 ลงไปปิดท้าย RDF list จะเป็นการกรองแยกเฉพาะตัวที่เป็นหัวเรื่องหลักออกจากส่วนย่อย ซึ่งจะทำให้เราสามารถ aggregate ข้อมูลแต่ละมิติได้ ซึ่งเป็นคุณสมบัติที่สำคัญอย่างนึงของโมเดลที่ผู้วิจัยออกแบบ ยกตัวอย่างเช่น ตามในภาพที่ 32 หากเราระบุแค่ index (2) ตัวเดียว เราจะได้ทั้ง (2 0), (2 1) และ (2 2) หรืออย่างเช่น index (3) ก็จะได้ element ที่ทั้งหมดในมิติที่หนึ่งมีค่าเป็น 3 แต่ถ้าเราระบุ (3 0) จะได้แค่ “Section 3 header” หรือหากคิวรี (3 2) จะได้ “Section 3 list item 2” ทั้งหมดรวมทั้งรายการย่อย “item 2.1” และ “item 2.2” แต่หากคิวรีแค่ (3 2 0) จะได้เฉพาะ “item 2” รายการเดียว ส่วน header บนสุดของบทความผู้วิจัยกำหนดให้เป็น (0) ลงไปเลย ซึ่งเมื่อดูโดยรวมทั้งหมดจะเป็นดังภาพที่ 33

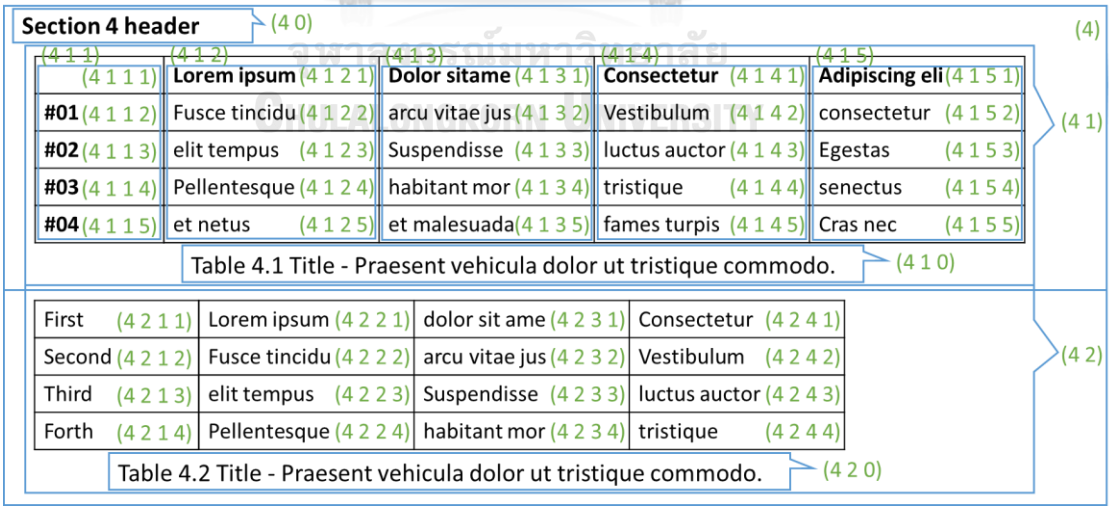


ภาพที่ 32 การเพิ่ม index 0 ให้กับ multi-dimensional array index เพื่อระบุเฉพาะหัวข้อหลัก



ภาพที่ 33 multi-dimensional array index ที่ใช้ระบุส่วนประกอบทั้งหมดของเอกสารตัวอย่าง

ในส่วนของตารางก็ใช้หลักการเดียวกัน ก็คือกำหนดลงไปอีกสองมิติตามในภาพที่ 34 ในกรณีนี้ยกตัวอย่างเป็นแบบ column-major เราก็จะ group มิติแรกที่สดมภ์ก่อน แล้วก็เซลล์ย่อยของสดมภ์นั้น ๆ ก็จะเป็นมิติถัดไป ซึ่งจะเป็นไปตามโครงสร้างที่เราเห็นในกราฟตัวอย่างที่ได้อธิบายแล้วในหัวข้อที่ผ่านมา หากมีตารางที่สองก็จะเป็นในทำนองเดียวกัน ก็จะมีตัวเลขเพิ่มขึ้นมาเป็นอีก index หนึ่งของมิติเดียวกัน



ภาพที่ 34 ตัวอย่างการกำหนด multi-dimensional array index สำหรับตาราง

สร้างการเข้าถึงข้อมูลแบบ direct

จะเห็นว่า RDF กราฟทั้งในภาพที่ 26 และในภาพที่ 29 เป็นโครงสร้างแบบ hierarchy และเรียงลำดับโดย tlp:index อยู่แล้ว (ขอเรียกว่า Index Member หรือ IM model) จึงสามารถใช้สร้างตารางและรายการเดิมกลับได้โดยไม่สูญเสียโครงสร้างต้นฉบับ โดยการทำ graph traversal แอปลิเคชันซีแมนติกเว็บต่าง ๆ สามารถเข้าถึงข้อมูลได้ตามลำดับชั้นด้วยวิธีการที่ไม่แตกต่างจากการเข้าถึง RDF container มาตรฐาน ซึ่งหากต้องการเข้าถึงข้อมูลแต่ละตัวแบบ arbitrary เราจำเป็นต้องเข้าถึงแบบ indirect โดยการทำ graph traversal ทีละชั้นไปเรื่อย ๆ จนกว่าจะถึงข้อมูลที่ต้องการ ซึ่งจะทำให้เขียนคิวรี SPARQL เป็นไปได้โดยยาก ดังนั้นผู้วิจัยจึงได้ออกแบบการเข้าถึงข้อมูลแบบ direct ขึ้นมาโดยการกำหนด “ตำแหน่ง” เพิ่มลงไปในข้อมูลแต่ละตัว คล้ายกับการสร้าง multi-dimension array index ซึ่งผู้วิจัยจะจำลองแนวคิดนี้โดยการใช้คุณสมบัติของ RDF ที่เรียกว่า collection หรือ RDF list เพื่อทำให้การคิวรีสามารถเข้าถึงตัวข้อมูลได้โดยตรง (ต่อไปจะเรียกว่า Index List หรือ IL model) โดยการสร้าง tlp:element เพื่อชี้ไปที่ blank node ของข้อมูลแต่ละตัวในชั้นเดียวกันหมดในโครงสร้างแบบแบนราบ (flat structure) และสร้างพร็อปเพอร์ตี้ tlp:indexList ให้แต่ละโหนด ซึ่งพร็อปเพอร์ตี้ tlp:indexList จะมี range เป็นคลาส rdf:List ของหมายเลขลำดับตาม tlp:index ในแต่ละชั้นของ IM model และปิดท้ายด้วย 0 โดยมีไว้เพื่อใช้ในการเลือกระบุว่าจะแยกเฉพาะ element ตัวเดียวโดด ๆ หรือรวมกลุ่ม element ในชั้นเดียวกันทั้งหมด (จะอธิบายรายละเอียดเรื่องนี้ในภายหลัง) ซึ่งการใช้โครงสร้างแบบแบนราบมีข้อดีคือ เราสามารถคิวรีเข้าถึงทุก element ในชั้นเดียว โดยไม่ต้องเข้าตาม hierarchy ทีละชั้นแบบ IM model หากเราลองนำตารางและรายการตัวอย่าง มาสร้าง RDF ทริปเปิลด้วย IL model จะเป็นดังต่อไปนี้

```
@prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix ex:   <http://example.org/sample#> .
@prefix tlp:  <http://purl.org/tulip/ns#> .

ex:TableExample
    tlp:element _:Table1,
                _:Col1, _:Cell11, _:Cell12, _:Cell13,
                _:Col2, _:Cell21, _:Cell22, _:Cell23,
                _:Col3, _:Cell31, _:Cell32, _:Cell33 .
_:Table1 rdf:type tlp:Table ;
    tlp:indexList ( 1 0 ) .
_:Col1 rdf:type tlp:Column ;
    tlp:indexList ( 1 1 0 ) .
_:Cell11 rdf:type tlp:Cell ;
    tlp:indexList ( 1 1 1 0 ) ;
```

```
    rdfs:label "Cell Content 1,1" .
_:Cell12 rdf:type tlp:Cell ;
    tlp:indexList ( 1 1 2 0 ) ;
    rdfs:label "Cell Content 1,2" .
_:Cell13 rdf:type tlp:Cell ;
    tlp:indexList ( 1 1 3 0 ) ;
    rdfs:label "Cell Content 1,3" .
_:Col2 rdf:type tlp:Column ;
    tlp:indexList ( 1 2 0 ) .
_:Cell21 rdf:type tlp:Cell ;
    tlp:indexList ( 1 2 1 0 ) ;
    rdfs:label "Cell Content 2,1" .
_:Cell22 rdf:type tlp:Cell ;
    tlp:indexList ( 1 2 2 0 ) ;
    rdfs:label "Cell Content 2,2" .
_:Cell23 rdf:type tlp:Cell ;
    tlp:indexList ( 1 2 3 0 ) ;
    rdfs:label "Cell Content 2,3" .
_:Col3 rdf:type tlp:Column ;
    tlp:indexList ( 1 3 0 ) .
_:Cell31 rdf:type tlp:Cell ;
    tlp:indexList ( 1 3 1 0 ) ;
    rdfs:label "Cell Content 3,1" .
_:Cell32 rdf:type tlp:Cell ;
    tlp:indexList ( 1 3 2 0 ) ;
    rdfs:label "Cell Content 3,2" .
_:Cell33 rdf:type tlp:Cell ;
    tlp:indexList ( 1 3 3 0 ) ;
    rdfs:label "Cell Content 3,3" .
```

รายการที่ 3 RDF ทริปเปิลของตารางตัวอย่าง สร้างโดยใช้ IL model

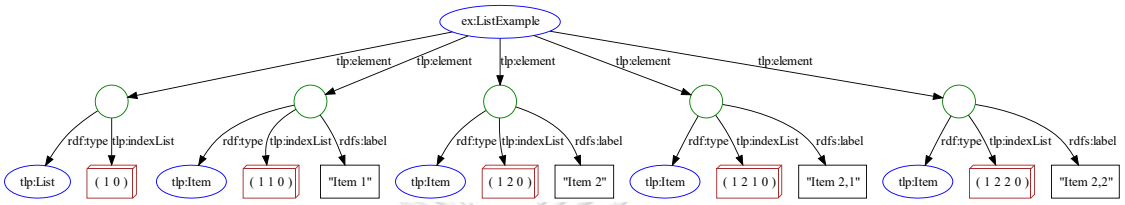
```
@prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix ex:   <http://example.org/sample#> .
@prefix tlp:  <http://purl.org/tulip/ns#> .

ex:ListExample
    tlp:element _:List1,
                _:Item1,
                _:Item2, _:Item21, _:Item22, _:Item221,
                _:Item3 .
_:List1 rdf:type tlp:List ;
    tlp:indexList ( 1 0 ) .
_:Item1 rdf:type tlp:Item ;
    tlp:indexList ( 1 1 0 ) ;
    rdfs:label "List Item 1" .
_:Item2 rdf:type tlp:Item ;
    tlp:indexList ( 1 2 0 ) ;
    rdfs:label "List Item 2" .
_:Item21 rdf:type tlp:Item ;
    tlp:indexList ( 1 2 1 0 ) ;
    rdfs:label "List Item 2,1" .
_:Item22 rdf:type tlp:Item ;
    tlp:indexList ( 1 2 2 0 ) ;
    rdfs:label "List Item 2,2" .
```

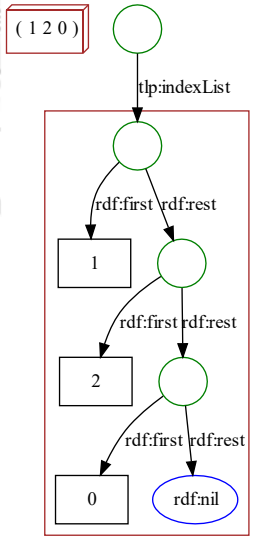
```
_:Item221 rdf:type tlp:Item ;
    tlp:indexList ( 1 2 2 1 0 ) ;
    rdfs:label "List Item 2,2,1" .
_:Item3 rdf:type tlp:Item ;
    tlp:indexList ( 1 3 0 ) ;
    rdfs:label "List Item 3" .
```

รายการที่ 4 RDF ทริปเปิลของรายการตัวอย่างสร้างโดยใช้ IL model

ขอยกตัวอย่าง RDF กราฟเฉพาะรายการที่ 4 เมื่อแสดงเพียง 5 โหนดแรกจะเป็นดังต่อไปนี้



โดยที่แต่ละ tlp:indexList แท้จริงแล้วมันคือ multi-dimensional array index ซึ่งเป็นโครงสร้างที่ใช้คุณสมบัติของ RDF อย่างหนึ่งที่เรียกว่า RDF collection หรือ RDF list คือเมื่อนำ RDF มาใช้เก็บ list จะเขียนในรูปแบบ S-expression ที่ประกอบด้วยวงเล็บซ้อนกัน ซึ่งแท้จริงแล้วมันเป็นกราฟย่อย ๆ ที่เมื่อนำมาขยายต่อจะได้เป็นโครงสร้างลักษณะคล้าย unbalanced binary tree โดยจะมี leaf node เป็นสมาชิกแต่ละลำดับของ RDF list ยกตัวอย่างเช่น tlp:indexList (1 2 0) จะมีโครงสร้างดังภาพที่ 35



ภาพที่ 35 กราฟตัวอย่างของ RDF list (1 2 0)

ภายใน tlp:indexList (1 2 0) คือ blank node ที่เชื่อมมาที่แต่ละ blank node ต่อเนื่องกันคล้ายกับ linked list ตัวเลขที่อยู่ในวงเล็บคือสมาชิกแต่ละตัวของ RDF list ซึ่งจะเป็น

ข้อมูลประเภทใดก็ได้เช่น ตัวเลขหรือสายอักขระ ในกรณีของ IL model เราเอาตัวเลข index มาเก็บเป็นสมาชิกของ RDF list ซึ่งมี rdf:first ชี้ไปที่ first member หรือสมาชิกตัวที่หนึ่งคือตัวเลข 1 ในขณะเดียวกันก็จะมี rdf:rest ชี้ไปที่ blank node ถัดไป ซึ่ง blank node ตัวต่อไปก็ชี้บอกว่า rdf:first เป็นตัวเลข 2 และก็มี rdf:rest ชี้ไปที่ blank node ที่มี rdf:first เป็นตัวเลข 0 ซึ่งจะเป็นสมาชิกตัวสุดท้าย แล้วก็ยังมีลิงก์ rdf:rest ชี้ไปที่ rdf:nil ซึ่งมีความหมายเหมือน null เพื่อเป็นตัวปิดบอกว่า RDF list นี้สิ้นสุดแล้ว ทั้งหมดนี้คือโครงสร้างที่แทนวงเล็บชุดนี้ กล่าวได้ว่า การใช้เครื่องหมายวงเล็บเป็นสัญลักษณ์ในการกำหนด RDF list แบบนี้เป็นเพียงแค่ syntactic sugar ซึ่งแท้จริงแล้วโครงสร้างที่เป็นกราฟ จะประกอบด้วย RDF ทริปเปิลถึง 7 บรรทัด ซึ่งถูกยุบอยู่ในบรรทัดเดียวใน Turtle ดังนั้นถ้าหากเรานำ Turtle บรรทัดนี้มาแปลงเป็นรูปแบบอื่น ยกตัวอย่างเช่น เอามาแปลงเป็น N-Triples ซึ่งจะเอาทริปเปิลแต่ละบรรทัดมาเรียงต่อกันเลย ก็จะกลายเป็น 7 บรรทัด ซึ่งการใช้เครื่องหมายวงเล็บแทน RDF list แบบนี้ถึงแม้ว่าโครงสร้างกราฟจะเป็นแบบเดียวกันไม่มีความแตกต่าง แต่จะทำให้เราใช้งานได้สะดวกขึ้นหากเราสร้าง RDF list ในรูปแบบ Turtle

ซึ่งนี่เป็นแนวคิดสำคัญอย่างหนึ่งของ IL model ในการนำ RDF list หรืออาจมองว่าคืออาร์เรย์ขนาด 1 มิติมาเก็บ subscript แต่ละชั้นของอาร์เรย์ขนาด n มิตินั่นเอง ทำให้การเข้าถึงแต่ละ element ของ IL model สามารถใช้ SPARQL ได้โดยการคิวรี element หา tlp:indexList ที่มี list item ตรงกับ subscript ที่ต้องการ ปัญหาอยู่ตรงข้อจำกัดของ SPARQL ที่ไม่มีความสามารถในการคิวรี RDF list หรือ RDF collection ได้โดยตรง (ทำให้เกิดงานวิจัยอีกหลายแนวทางในการสร้างสิ่งที่จะมาทดแทน RDF collection ตามมาตรฐาน เช่น Ordered List Ontology[99] และ Collections Ontology[100] หรือแนวคิดที่จะปรับปรุง SPARQL ให้เข้าถึง RDF collection ได้สะดวกขึ้น เช่น ข้อเสนอแนะของ Leigh และ Wood[101]) ทางแก้ไขปัญหาเฉพาะหน้าระหว่างรอ SPARQL เวอร์ชันถัดไปที่อาจจะมีการปรับปรุงในเรื่องนี้ คือต้องประยุกต์ใช้คุณสมบัติที่เรียกว่า property path ที่มีใน SPARQL 1.1 มาใช้เพื่อเข้าถึง collection item หรือ list item แต่ละตัว ยกตัวอย่างเช่น หากต้องการข้อมูลในรายการตัวอย่าง รายการที่ 1 ไอเท็มที่ 2 เราเพียงแค่คิวรีหา element ที่มี tlp:indexList แมทช์กับ (1 2) ด้วยคิวรี SPARQL ดังนี้

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX ex:   <http://example.org/sample#>
PREFIX tlp:  <http://purl.org/tulip/ns#>

SELECT ?label
WHERE {
    ex:ListExample tlp:element ?elem .
```

```
    ?elem tlp:indexList/rdf:first 1 .
    ?elem tlp:indexList/rdf:rest/rdf:first 2 .
    ?elem rdfs:label ?label .
}
```

จะได้ผลลัพธ์ดังนี้

```
"List Item 2"
"List Item 2,1"
"List Item 2,2"
"List Item 2,2,1"
```

ซึ่งจะเป็นทุกไอเท็มที่อยู่ในลำดับที่ 2 แต่หากต้องการเฉพาะไอเท็มหลักหมายเลข 2 ไอเท็มเดียวเท่านั้น ทำได้โดยแมทช์เพิ่ม subscript ตัวสุดท้ายด้วยตัวเลข 0 เพื่อเลือกเฉพาะโหนดที่ตรงกับที่กรองไว้เพียงโหนดเดียว ซึ่งทำได้โดยแค่เพิ่มการกรองไปในคิวรีอีก 1 บรรทัดเพื่อให้แมทช์กับ indexList (1 2 0) คือ

```
    ?elem tlp:indexList/rdf:rest{2}/rdf:first 0 .
```

แต่เป็นที่น่าเสียดายที่ property path รูปแบบนี้ใช้ได้เฉพาะใน SPARQL 1.1 Working Draft 05 January 2012[18] เท่านั้น หลังจากนั้นเมื่อ SPARQL 1.1 Recommendation Specification จริงออกมาก็ได้ตัดคุณสมบัตินี้ทิ้งไป (ซึ่ง SPARQL engine บางตัวก็ยังคงรองรับรูปแบบนอกมาตรฐานแบบนี้อยู่ เช่น Jena ARQ รวมทั้งยังมี extension เพิ่มเติมจาก SPARQL มาตรฐานเพื่อรองรับการใช้งาน RDF collection ได้สะดวกมากขึ้นด้วย[19]) ดังนั้นจึงต้องแก้ปัญหาโดยการใส่ property path ซ้อนต่อกันไปตามลำดับความลึกของ RDF collection หรือ subscript ที่เราต้องการ (ซึ่งความจริงแล้วเรื่องนี้ไม่ใช่ปัญหาใหญ่มากนัก เนื่องจากในทางปฏิบัติเรามักไม่ได้เขียนคิวรีด้วยมือ แต่เราสร้างคิวรีขึ้นมาผ่านทางการเขียนโปรแกรม ซึ่งเราจะสร้างอย่างซับซ้อนแค่ไหนก็ได้ เทียบเคียงได้อย่างเช่น ทุกวันนี้เราแทบไม่ได้สร้างคิวรี SQL ด้วยมือ แต่คำสั่ง SQL ที่ซับซ้อนในการใช้งานจริงในแอปพลิเคชันต่าง ๆ ถูกสร้างขึ้นมาผ่านทางโปรแกรมมิ่ง) ซึ่ง SPARQL บรรทัดนั้นก็จะถูกเขียนใหม่ให้สอดคล้องตามมาตรฐาน SPARQL 1.1 ได้แบบนี้

```
    ?elem tlp:indexList/rdf:rest/rdf:rest/rdf:first 0 .
```

ดังนั้นคิวรีที่เพิ่มบรรทัดแล้วจะเป็นดังนี้

[18] SPARQL 1.1 Query Language. Available from: https://www.w3.org/TR/2012/WD-sparql11-query-20120105/#propertypaths

[19] Apache Jena – ARQ – RDF Collections. Available from: https://jena.apache.org/documentation/query/rdf_lists.html

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX ex:   <http://example.org/sample#>
PREFIX tlp:  <http://purl.org/tulip/ns#>

SELECT ?label
WHERE {
    ex:ListExample tlp:element ?elem .
    ?elem tlp:indexList/rdf:first 1 .
    ?elem tlp:indexList/rdf:rest/rdf:first 2 .
    ?elem tlp:indexList/rdf:rest/rdf:rest/rdf:first 0 .
    ?elem rdfs:label ?label .
}
```

ซึ่งจะได้ผลลัพธ์ดังนี้

```
"List Item 2"
```

ทำนองเดียวกันกับตารางตัวอย่าง หากต้องการทั้งสดมภ์ที่ 3 ของตารางที่ 1 เราจะทำการแมทช์ tlp:indexList ด้วย (1 3) ดังนี้

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX ex:   <http://example.org/sample#>
PREFIX tlp:  <http://purl.org/tulip/ns#>

SELECT ?label
WHERE {
    ex:TableExample tlp:element ?elem .
    ?elem tlp:indexList/rdf:first 1 .
    ?elem tlp:indexList/rdf:rest/rdf:first 3 .
    ?elem rdfs:label ?label .
}
```


เนื่องจากตารางนี้ถูกกำหนดให้เป็นแบบ column-major ดังนั้นจะได้ผลลัพธ์ดังนี้ คือได้สมาชิกทั้ง 3 ตัวของ group ที่ 3 หรือสดมภ์ที่ 3 นั่นเอง

```
"Cell Content 3,1"
"Cell Content 3,2"
"Cell Content 3,3"
```

หรือหากต้องการทั้ง line ที่ 3 ซึ่งในกรณีนี้คือทั้ง row ที่ 3 จะทำได้โดยแมทช์ tlp:indexList ด้วย (1 ? 3) โดย ? ที่ตำแหน่ง subscript ลำดับที่ 2 คือชั้นของ group ซึ่งหมายถึงเราจะไม่กรองดังนั้นก็จะได้มาทุกสดมภ์

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX ex:   <http://example.org/sample#>
PREFIX tlp:  <http://purl.org/tulip/ns#>

SELECT ?label
```

```
WHERE {
    ex:TableExample tlp:element ?elem .
    ?elem tlp:indexList/rdf:first 1 .
    ?elem tlp:indexList/rdf:rest/rdf:rest/rdf:first 3 .
    ?elem rdfs:label ?label .
}
```

อย่างที่กล่าวในหัวข้อที่ผ่านมา ในกรณีของ column-major การรวม line จะได้มาเป็น virtual row ดังนั้นไม่ว่าเซลล์นั้นจะอยู่ใน group ใดที่ตรงกับ line 3 ก็จะมาทั้งหมด ซึ่งจะเทียบเท่ากับการดึงมาทั้งแถว จึงได้ผลลัพธ์ดังนี้

```
"Cell Content 1,3"
"Cell Content 2,3"
"Cell Content 3,3"
```

ในทำนองเดียวกัน หากเรากำหนดเพิ่มลงไปทั้ง group และ line เช่นบอกว่าจะเอาเฉพาะสดมภ์ที่ 2 และแถวที่ 3 ของตารางที่ 1 แบบนี้

```
    ?elem tlp:indexList/rdf:first 1 .
    ?elem tlp:indexList/rdf:rest/rdf:first 2 .
    ?elem tlp:indexList/rdf:rest/rdf:rest/rdf:first 3 .
```

ก็จะได้มาแค่เซลล์เดียวคือ

```
"Cell Content 2,3"
```

นำข้อดีของทั้งสองโมเดลมารวมกัน

RDF Schema model ทั้งสองรูปแบบคือ Index Member model และ Index List model มีข้อดีและข้อด้อยแตกต่างกัน ผู้ใช้สามารถเลือกใช้แบบใดแบบหนึ่งตามความเหมาะสมกับความต้องการ หรือจะสร้าง RDF ทริปเปิลโดยใช้ model แบบลูกผสมที่เรียกว่า “Hybrid model” ซึ่งจะรวมโครงสร้างของทั้งสองรูปแบบมาไว้ด้วยกันในโครงสร้างเดียว หลักการคือ นำพร็อปเพอร์ตี้ IM model มาตั้งต้นเป็นพื้นฐาน แล้วนำพร็อปเพอร์ตี้ของ IL model มาเสริมเข้าไป โดยเราจะสร้างพร็อปเพอร์ตี้ tlp:indexList เข้าไปในแต่ละ blank node ของ IM model และเพิ่มพร็อปเพอร์ตี้ tlp:element จากรีซอร์สหลักชี้ไปที่ blank node ของ IM model ทุกตัวของทุกชั้น ตัวอย่างเช่นเมื่อเพิ่ม tlp:indexList และ tlp:element เข้าไปในรายการที่ 2 แล้วจะเป็นดังนี้

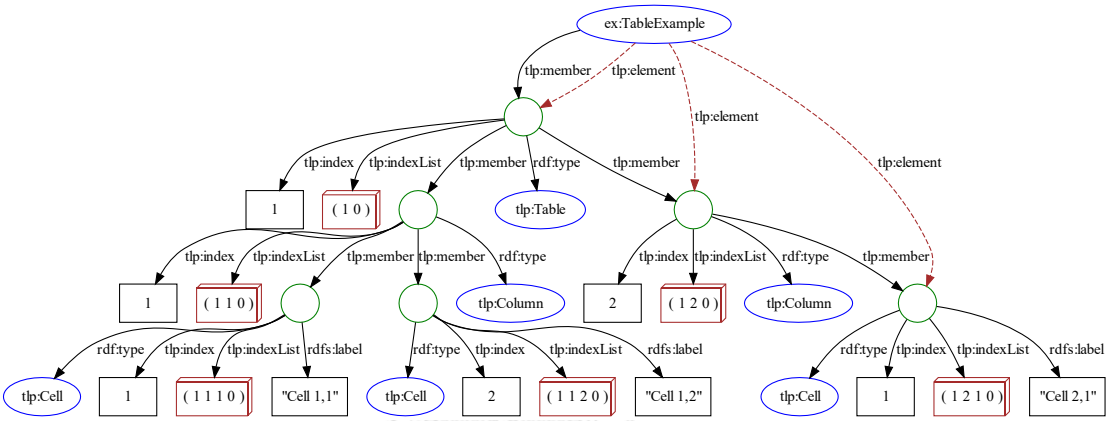
```
@prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix ex:   <http://example.org/sample#> .
```

```
@prefix tlp:  <http://purl.org/tulip/ns#> .

ex:TableExample
    tlp:element _:Table1,
                _:Col1, _:Cell11, _:Cell12, _:Cell13,
                _:Col2, _:Cell21, _:Cell22, _:Cell23,
                _:Col3, _:Cell31, _:Cell32, _:Cell33 ;
    tlp:member _:Table1 .
_:Table1 rdf:type tlp:Table ;
    tlp:index 1 ;
    tlp:indexList ( 1 0 ) ;
    tlp:member _:Col1, _:Col2, _:Col3 .
_:Col1 rdf:type tlp:Column ;
    tlp:index 1 ;
    tlp:indexList ( 1 1 0 ) ;
    tlp:member _:Cell11, _:Cell12, _:Cell13 .
_:Cell11 rdf:type tlp:Cell ;
    tlp:index 1 ;
    tlp:indexList ( 1 1 1 0 ) ;
    rdfs:label "Cell Content 1,1" .
_:Cell12 rdf:type tlp:Cell ;
    tlp:index 2 ;
    tlp:indexList ( 1 1 2 0 ) ;
    rdfs:label "Cell Content 1,2" .
_:Cell13 rdf:type tlp:Cell ;
    tlp:index 3 ;
    tlp:indexList ( 1 1 3 0 ) ;
    rdfs:label "Cell Content 1,3" .
_:Col2 rdf:type tlp:Column ;
    tlp:index 2 ;
    tlp:indexList ( 1 2 0 ) ;
    tlp:member _:Cell21, _:Cell22, _:Cell23 .
_:Cell21 rdf:type tlp:Cell ;
    tlp:index 1 ;
    tlp:indexList ( 1 2 1 0 ) ;
    rdfs:label "Cell Content 2,1" .
_:Cell22 rdf:type tlp:Cell ;
    tlp:index 2 ;
    tlp:indexList ( 1 2 2 0 ) ;
    rdfs:label "Cell Content 2,2" .
_:Cell23 rdf:type tlp:Cell ;
    tlp:index 3 ;
    tlp:indexList ( 1 2 3 0 ) ;
    rdfs:label "Cell Content 2,3" .
_:Col3 rdf:type tlp:Column ;
    tlp:index 3 ;
    tlp:indexList ( 1 3 0 ) ;
    tlp:member _:Cell31, _:Cell32, _:Cell33 .
_:Cell31 rdf:type tlp:Cell ;
    tlp:index 1 ;
    tlp:indexList ( 1 3 1 0 ) ;
    rdfs:label "Cell Content 3,1" .
_:Cell32 rdf:type tlp:Cell ;
    tlp:index 2 ;
    tlp:indexList ( 1 3 2 0 ) ;
```

```
    rdfs:label "Cell Content 3,2" .
_:Cell33 rdf:type tlp:Cell ;
    tlp:index 3 ;
    tlp:indexList ( 1 3 3 0 ) ;
    rdfs:label "Cell Content 3,3" .
```

ซึ่งเมื่อแสดงเป็น RDF กราฟแล้วจะเป็นดังนี้ (เพื่อให้กราฟกระชับขึ้น ขอปรับขนาดตารางเป็น 2 column x 2 row และตัดโหนดสุดท้ายทิ้งไป และในที่นี้จะขอแสดง edge tlp:element เฉพาะบางเส้นเพื่อให้กราฟดูได้สะดวกและชัดเจน ซึ่งในความเป็นจริง tlp:element จะชี้ไปยังทุก blank node)



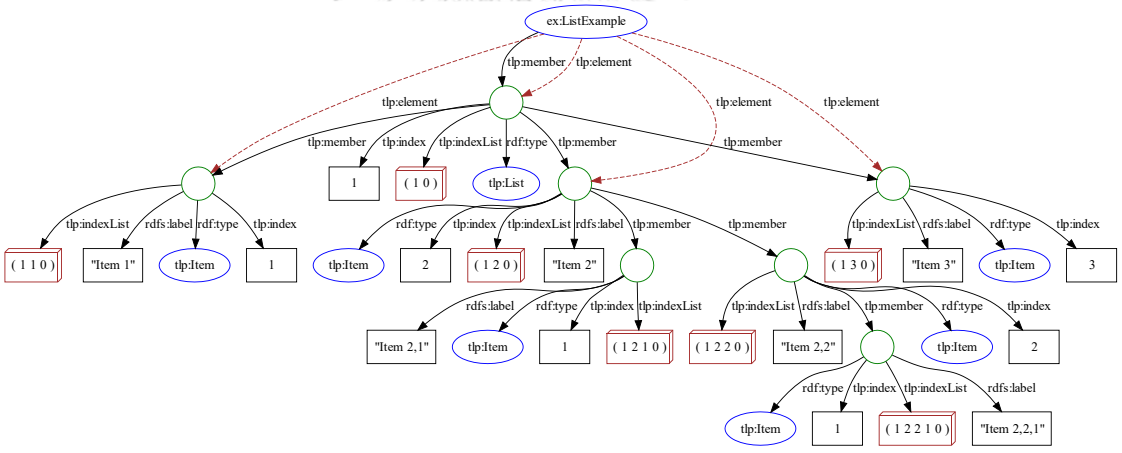
ในทำนองเดียวกัน Hybrid model ของรายการตัวอย่างจะเป็นดังนี้

```
@prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix ex:   <http://example.org/sample#> .
@prefix tlp:  <http://purl.org/tulip/ns#> .

ex:ListExample
    tlp:element _:List1,
                _:Item1,
                _:Item2, _:Item21, _:Item22, _:Item221,
                _:Item3 ;
    tlp:member _:List1 .
_:List1 rdf:type tlp:List ;
    tlp:index 1 ;
    tlp:indexList ( 1 0 ) ;
    tlp:member _:Item1, _:Item2, _:Item3 .
_:Item1 rdf:type tlp:Item ;
    tlp:index 1 ;
    tlp:indexList ( 1 1 0 ) ;
    rdfs:label "List Item 1" .
_:Item2 rdf:type tlp:Item ;
    tlp:index 2 ;
    tlp:indexList ( 1 2 0 ) ;
    rdfs:label "List Item 2" ;
```

```
    tlp:member _:Item21, _:Item22 .
_:Item21 rdf:type tlp:Item ;
    tlp:index 1 ;
    tlp:indexList ( 1 2 1 0 ) ;
    rdfs:label "List Item 2,1" .
_:Item22 rdf:type tlp:Item ;
    tlp:index 2 ;
    tlp:indexList ( 1 2 2 0 ) ;
    rdfs:label "List Item 2,2" ;
    tlp:member _:Item221 .
_:Item221 rdf:type tlp:Item ;
    tlp:index 1 ;
    tlp:indexList ( 1 2 2 1 0 ) ;
    rdfs:label "List Item 2,2,1" .
_:Item3 rdf:type tlp:Item ;
    tlp:index 3 ;
    tlp:indexList ( 1 3 0 ) ;
    rdfs:label "List Item 3" .
```

ซึ่งเมื่อนำมาแสดงเป็น RDF กราฟจะเป็นดังนี้ (ขอแสดง edge tlp:element เพียงบางเส้น เพื่อให้กราฟดูสะดวกและชัดเจน)



การคิวรีข้อมูล TULIP

หัวข้อนี้จะทำการสาธิตการคิวรีข้อมูล TULIP โดยจะยกตัวอย่างจากข้อมูลอันดับของจุฬาลงกรณ์มหาวิทยาลัยตามตารางที่ 1 เช่น ต้องการทราบว่าในปี ค.ศ. 2009 สาขา Engineering & IT อยู่ในอันดับที่เท่าไหร่ ดังตารางที่ 2 (เพื่อแสดงให้เห็นอย่างชัดเจน จะเน้นโดยใช้สีเหลืองที่คำถามและใช้สีเขียวที่คำตอบ)

ตารางที่ 2 ยกตัวอย่างกรณีที่ต้องการทราบข้อมูลอันดับของสาขา Engineering & IT ในปี ค.ศ. 2009

University World Ranking in 2005 – 2012 by QS World University Rankings

	2012	2011	2010	2009	2008	2007	2006	2005
Overall	201	171	180	138	166	223	161	161
Arts & Humanities	113	69	78	49	119	136	–	–
Natural Sciences	140	138	186	136	171	159	–	–
Engineering & IT	106	104	101	78	86	100	–	–
Social Sciences	80	68	78	51	72	83	–	–
Life Sciences	92	78	130	51	108	138	–	–

โดยทำการสอบถามด้วย SPARQL คิวรีดังนี้

```
SELECT ?ranking
WHERE {
   ?resource prov:wasDerivedFrom article:Chulalongkorn_University .
   ?resource tlp:member ?table .
   ?table rdfs:label "University World Ranking in 2005 - 2012 by QS
World University Rankings" .
   ?table tlp:member/tlp:member ?colHead .
   ?table tlp:member/tlp:member ?rowHead .
   ?table tlp:member/tlp:member ?cell .
   ?colHead tlp:indexList/rdf:rest/rdf:first ?col .
   ?colHead tlp:indexList/rdf:rest/rdf:rest/rdf:first 1 .
   ?colHead rdfs:label "2009" .
   ?rowHead tlp:indexList/rdf:rest/rdf:first 1 .
   ?rowHead tlp:indexList/rdf:rest/rdf:rest/rdf:first ?row .
   ?rowHead rdfs:label "Engineering & IT" .
   ?cell tlp:indexList/rdf:rest/rdf:first ?col .
   ?cell tlp:indexList/rdf:rest/rdf:rest/rdf:first ?row .
   ?cell rdfs:label ?ranking .
}
```

กรณีนี้เราไม่ได้เลือกตารางด้วยหมายเลขของตาราง แต่ทำการกำหนดด้วยชื่อตารางเลย แล้วกำหนดว่าเป็นปี ค.ศ. 2009 และสาขา Engineering & IT ถัดจากนั้นสอบถามด้วย ?ranking ซึ่งผลลัพธ์ก็จะแสดงขึ้นมาแค่จำนวนเดียวคือ 78

ในทางกลับกันถ้าเราทำการสอบถามว่า อันดับ 49 เป็นของสาขาอะไร เราจะทำการสอบถามด้วยคิวรีดังนี้

```
SELECT ?group ?year
WHERE {
   ?resource prov:wasDerivedFrom article:Chulalongkorn_University .
```

```
   ?resource tlp:member ?table .
   ?table rdfs:label "University World Ranking in 2005 - 2012 by QS
World University Rankings" .
   ?table tlp:member/tlp:member ?colHead .
   ?table tlp:member/tlp:member ?rowHead .
   ?table tlp:member/tlp:member ?cell .
   ?colHead tlp:indexList/rdf:rest/rdf:first ?col .
   ?colHead tlp:indexList/rdf:rest/rdf:rest/rdf:first 1 .
   ?colHead rdfs:label ?year .
   ?rowHead tlp:indexList/rdf:rest/rdf:first 1 .
   ?rowHead tlp:indexList/rdf:rest/rdf:rest/rdf:first ?row .
   ?rowHead rdfs:label ?group .
   ?cell tlp:indexList/rdf:rest/rdf:first ?col .
   ?cell tlp:indexList/rdf:rest/rdf:rest/rdf:first ?row .
   ?cell rdfs:label "49" .
}
```

โดยสอบถามว่า ?group และ ?year อะไร โดยกำหนดอันดับด้วย 49 ซึ่งผลลัพธ์ที่ได้คือสาขา Arts & Humanities และปี ค.ศ. 2009 ดังตารางที่ 3

ตารางที่ 3 ยกตัวอย่างกรณีที่ต้องการทราบข้อมูลสาขาและปีของอันดับที่ 49

University World Ranking in 2005 – 2012 by QS World University Rankings

	2012	2011	2010	2009	2008	2007	2006	2005
Overall	201	171	180	138	166	223	161	161
Arts & Humanities	113	69	78	49	119	136	–	–
Natural Sciences	140	138	186	136	171	159	–	–
Engineering & IT	106	104	101	78	86	100	–	–
Social Sciences	80	68	78	51	72	83	–	–
Life Sciences	92	78	130	51	108	138	–	–

ต่อไปเราจะทดลองสอบถามกลับด้วยอันดับที่ 78 ซึ่งเป็นอันดับผลลัพธ์ที่ได้ของสาขา Engineering & IT ในปี ค.ศ. 2009 จากที่ได้ทดลองสอบถามในตัวอย่างแรก ด้วยคิวรีเดียวกันกับในตัวอย่างที่แล้ว แต่เปลี่ยนอันดับจาก 49 เป็น 78 ดังนี้

```
SELECT ?group ?year
WHERE {
   ?resource prov:wasDerivedFrom article:Chulalongkorn_University .
   ?resource tlp:member ?table .
   ?table rdfs:label "University World Ranking in 2005 - 2012 by QS
World University Rankings" .
   ?table tlp:member/tlp:member ?colHead .
   ?table tlp:member/tlp:member ?rowHead .
   ?table tlp:member/tlp:member ?cell .
```

```
   ?colHead tlp:indexList/rdf:rest/rdf:first ?col .
   ?colHead tlp:indexList/rdf:rest/rdf:rest/rdf:first 1 .
   ?colHead rdfs:label ?year .
   ?rowHead tlp:indexList/rdf:rest/rdf:first 1 .
   ?rowHead tlp:indexList/rdf:rest/rdf:rest/rdf:first ?row .
   ?rowHead rdfs:label ?group .
   ?cell tlp:indexList/rdf:rest/rdf:first ?col .
   ?cell tlp:indexList/rdf:rest/rdf:rest/rdf:first ?row .
   ?cell rdfs:label "78" .
}
```

ซึ่งผลลัพธ์ในครั้งนี้จะได้ออกมา 4 คำตอบคือ

```
Arts & Humanities 2010
Engineering & IT  2009
Social Sciences   2010
Life Sciences     2011
```

เพราะว่าเครื่องไม่สามารถรู้ว่าอันดับ 78 ที่ได้ทำการสอบถามไปคือข้อมูล ณ ตำแหน่งไหน ดังนั้นจึงทำการไปสืบค้นดูอันดับ 78 มาทั้งหมดที่มี และตอบกลับมาทั้งหมดซึ่งตรงกับตารางที่ 4

ตารางที่ 4 ยกตัวอย่างกรณีที่ต้องการทราบข้อมูลสาขาและปีของอันดับที่ 78

University World Ranking in 2005 – 2012 by QS World University Rankings

	2012	2011	2010	2009	2008	2007	2006	2005
Overall	201	171	180	138	166	223	161	161
Arts & Humanities	113	69	78	49	119	136	–	–
Natural Sciences	140	138	186	136	171	159	–	–
Engineering & IT	106	104	101	78	86	100	–	–
Social Sciences	80	68	78	51	72	83	–	–
Life Sciences	92	78	130	51	108	138	–	–

หรือเราอาจทำการสอบถามเพียงแค่อย่างเดียว เช่น ถามแค่ปี ค.ศ. คำตอบที่ได้ก็จะมาทุกสาขาและอันดับในปีนั้น โดยคิวรีดังต่อไปนี้

```
SELECT ?group ?ranking
WHERE {
   ?resource prov:wasDerivedFrom article:Chulalongkorn_University .
   ?resource tlp:member ?table .
   ?table rdfs:label "University World Ranking in 2005 - 2012 by QS
World University Rankings" .
   ?table tlp:member/tlp:member ?colHead .
   ?table tlp:member/tlp:member ?rowHead .
   ?table tlp:member/tlp:member ?cell .
```

```
   ?colHead tlp:indexList/rdf:rest/rdf:first ?col .
   ?colHead tlp:indexList/rdf:rest/rdf:rest/rdf:first 1 .
   ?colHead rdfs:label "2009" .
   ?rowHead tlp:indexList/rdf:rest/rdf:first 1 .
   ?rowHead tlp:indexList/rdf:rest/rdf:rest/rdf:first ?row .
   ?rowHead rdfs:label ?group .
   ?cell tlp:indexList/rdf:rest/rdf:first ?col .
   ?cell tlp:indexList/rdf:rest/rdf:rest/rdf:first ?row .
   ?cell rdfs:label ?ranking .
}
```

ก็จะได้ผลลัพธ์ตามตารางที่ตารางที่ 5

ตารางที่ 5 ยกตัวอย่างกรณีที่ต้องการทราบข้อมูลทุกสาขาและทุกอันดับของปี ค.ศ. 2009

University World Ranking in 2005 – 2012 by QS World University Rankings

	2012	2011	2010	2009	2008	2007	2006	2005
Overall	201	171	180	138	166	223	161	161
Arts & Humanities	113	69	78	49	119	136	–	–
Natural Sciences	140	138	186	136	171	159	–	–
Engineering & IT	106	104	101	78	86	100	–	–
Social Sciences	80	68	78	51	72	83	–	–
Life Sciences	92	78	130	51	108	138	–	–

หรือหากสอบถามว่าในสาขา Arts & Humanities ในปี ค.ศ. ไหนได้อันดับที่เท่าไหร่ ก็สามารถสอบถามด้วยคิวรีดังนี้


```
SELECT ?year ?ranking
WHERE {
   ?resource prov:wasDerivedFrom article:Chulalongkorn_University .
   ?resource tlp:member ?table .
   ?table rdfs:label "University World Ranking in 2005 - 2012 by QS
World University Rankings" .
   ?table tlp:member/tlp:member ?colHead .
   ?table tlp:member/tlp:member ?rowHead .
   ?table tlp:member/tlp:member ?cell .
   ?colHead tlp:indexList/rdf:rest/rdf:first ?col .
   ?colHead tlp:indexList/rdf:rest/rdf:rest/rdf:first 1 .
   ?colHead rdfs:label ?year .
   ?rowHead tlp:indexList/rdf:rest/rdf:first 1 .
   ?rowHead tlp:indexList/rdf:rest/rdf:rest/rdf:first ?row .
   ?rowHead rdfs:label "Arts & Humanities" .
   ?cell tlp:indexList/rdf:rest/rdf:first ?col .
   ?cell tlp:indexList/rdf:rest/rdf:rest/rdf:first ?row .
   ?cell rdfs:label ?ranking .
}
```

ก็จะได้ผลลัพธ์ออกมาตามตารางที่ 6

ตารางที่ 6 ยกตัวอย่างกรณีที่ต้องการทราบข้อมูลทุกอันดับในแต่ละปีของสาขา Arts & Humanities

University World Ranking in 2005 – 2012 by QS World University Rankings

	2012	2011	2010	2009	2008	2007	2006	2005
Overall	201	171	180	138	166	223	161	161
Arts & Humanities	113	69	78	49	119	136	–	–
Natural Sciences	140	138	186	136	171	159	–	–
Engineering & IT	106	104	101	78	86	100	–	–
Social Sciences	80	68	78	51	72	83	–	–
Life Sciences	92	78	130	51	108	138	–	–

จากตัวอย่างทั้งหมดดังกล่าว ทำให้เราสามารถที่จะ roll-up, drill-down หรือ slicing/dicing ได้อย่างอิสระเพราะว่าข้อมูลถูกเก็บในลักษณะ multi-dimensional array ดังนั้นจึงสามารถเข้าถึงข้อมูลได้ทุกตำแหน่งโดยการกำหนดมิติ โดยเฉพาะอย่างยิ่งเนื่องจากข้อมูล TULIP เป็น 5 ดาวอย่างสมบูรณ์ จึงสามารถเชื่อมข้อมูลข้ามตาราง หรือข้ามเอกสารได้อีกด้วย จะขอยกตัวอย่างโดยการใช้คิวรีดังนี้เพื่อสืบค้นข้อมูลที่เกี่ยวข้องกับภาพยนตร์เรื่อง Shrek

```
SELECT ?resource ?section ?group ?value ?link_text ?link_url
WHERE {
    ?link tlp:url article:Shrek .
    ?cell tlp:link ?link .
    ?cell tlp:index ?row .
    ?cell_group tlp:member ?cell .
    ?section_node tlp:member ?cell_group .
    ?section_node tlp:member ?all_group .
    ?section_node rdfs:label ?section .
    ?all_group tlp:member ?value_node .
    ?all_group tlp:member ?first_cell .
    ?value_node tlp:index ?row .
    ?value_node rdfs:label ?value .
    ?first_cell tlp:index 1 .
    ?first_cell rdfs:label ?group .
    ?resource_node tlp:element ?cell .
    ?resource_node rdfs:label ?resource .
    OPTIONAL {
        ?value_node tlp:link ?value_link .
        ?value_link tlp:text ?link_text .
        ?value_link tlp:url ?link_url .
    }
}
```

ซึ่งจะได้ผลลัพธ์เป็นชุดของข้อมูลดังภาพที่ 36 (ตัดมาเฉพาะบางส่วน)

Resource	Section	Group	Value	Link_text	Link_url
"Academy Award for Best Animated Feature"	"Academy Award for Best Animated Feature"	"Academy Award for Best Animated Feature"	"2001 (for Shrek)"	"2001"	wikipedia:74th_Academy_Awards
"Academy Award for Best Animated Feature"	"Academy Award for Best Animated Feature"	"Academy Award for Best Animated Feature"	"2001 (for Shrek)"	"Shrek"	wikipedia:Shrek
"Academy Award for Best Animated Feature"	"Academy Award for Best Animated Feature"	"Academy Award for Best Animated Feature"	"First awarded"		
"Academy Award for Best Animated Feature"	"Winners and nominees"	"Film"	"Shrek"	"Shrek"	wikipedia:Shrek
"Academy Award for Best Animated Feature"	"Winners and nominees"	"Nominees"	"Aron Warner"	"Aron Warner"	wikipedia:Aron_Warner
"Academy Award for Best Animated Feature"	"Winners and nominees"	"Year"	2001 (74th)	"2001"	wikipedia:2001_in_animation
"Academy Award for Best Animated Feature"	"Winners and nominees"	"Year"	2001 (74th)	"74th"	wikipedia:74th_Academy_Awards
"Academy Award for Best Animated Feature"	"Studios with multiple nominations"	"Studio"	"DreamWorks Animation"	"DreamWorks Animation"	wikipedia:DreamWorks_Animation
"Academy Award for Best Animated Feature"	"Studios with multiple nominations"	"Wins"	"1"		
"Academy Award for Best Animated Feature"	"Studios with multiple nominations"	"Nominations"	"12"		

ภาพที่ 36 ผลลัพธ์ที่ได้จากการสืบค้นข้อมูลที่เกี่ยวกับภาพยนตร์เรื่อง Shrek

โดยเราสามารถนำผลลัพธ์ที่ได้กลับมาแสดงผลในรูปแบบตารางซึ่งจะแสดงเฉพาะสดมภ์และแถวที่เกี่ยวกับภาพยนตร์เรื่อง Shrek โดยที่ไม่มีส่วนอื่นของตารางต้นฉบับที่ไม่เกี่ยวข้องออกมาปะปนด้วยดังภาพที่ 37 (ตัดมาเฉพาะบางส่วน)

Academy Award for Best Animated Feature

- Academy Award for Best Animated Feature

Academy Award for Best Animated Feature
2001 (for Shrek)
First awarded

- Winners and nominees

Film	Nominees	Year
Shrek	Aron Warner	2001 (74th)

- Studios with multiple nominations

Studio	Wins	Nominations	Films
DreamWorks Animation	1	12	Shrek, Spirit: Stallion of the Cimarron, Shrek 2, Shark Tale, Kung Fu Panda, How to 2, Puss in Boots, The Croods, How to Train Your Dragon 2, The Boss Baby, How to World

List of 3D films (2005 onwards)

- 2010

Title	Release date	Prod. country	Camera system	Aspect ratio	Runtime min	Notes
Shrek	December 1, 2010	United States	Rendered in 2D	1.78:1	93	Originally released in 2D in 2001, convert re-release.

Annie Award for Best Animated Feature

- 2000s

Year	Film	Studios
2001	Shrek	DreamWorks Animation / Pacific Data Images

ภาพที่ 37 การนำผลลัพธ์ที่ได้จาการสืบค้นข้อมูลภาพยนตร์เรื่อง Shrek มาแสดงในรูปแบบตาราง

ซึ่งจะแสดงข้อมูลที่เกี่ยวข้องกับภาพยนตร์เรื่อง Shrek ในรูปแบบของลิงก์อยู่ในหลายบทความ เช่น บทความที่เกี่ยวกับ Academy Award เพราะภาพยนตร์เรื่องนี้ได้รางวัลภาพยนตร์แอนิเมชันยอดเยี่ยมในปี ค.ศ. 2001 และได้รางวัล Annie Award สาขาเดียวกันด้วยในปีเดียวกัน หลังจากนั้นเมื่อสืบค้นต่อไปที่ Academy Award จะพบว่าผู้ที่ได้รับรางวัลคือ Aaron Werner และในทำนองเดียวกันข้อมูลที่พูดถึงใน Annie Award ก็จะมีชื่อบริษัทผู้สร้างคือ DreamWorks Animation และ Pacific Data Images เป็นต้น

TULIP - Table and List with Interchangeable, Unified and Pivotal

TULIP[102] เป็นชุดของ RDF Schema vocabulary ที่ประกอบไปด้วย ชุดของพร็อปเพอร์ตี้และคลาสต่าง ๆ จำนวนหนึ่ง ซึ่งใช้ในการกำหนดโครงสร้างสำหรับการทำ data representation เพื่อเก็บข้อมูลตารางและรายการให้ครบถ้วนตามต้นฉบับ และคุณสมบัติพื้นฐานที่จำเป็นสำหรับ 5-star Open Data รวมไปถึงการทำให้เกิดคุณสมบัติเฉพาะทั้ง 3 รูปแบบของ TULIP คือ Interchangeable, Unified และ Pivotal ซึ่งจะอธิบายรายละเอียดทีละคุณสมบัติในลำดับต่อไป

แนวคิดสำคัญของ TULIP คือทำให้ข้อมูลที่อยู่ในรูปแบบตารางและรายการที่เป็น semi-structured data ไม่ว่าจะมาจากแหล่งไหนก็ตาม เช่น กระดาษ, Office document (Excel/Calc, Word/Writer), HTML table/list ฯลฯ ให้มาเป็น structured data ที่อยู่ในรูปแบบ 5-star Open Data หมายความว่า ข้อมูลทั้งหมดจะต้องกลายสภาพเป็นชุดประโยค RDF ทริปเปิลซึ่งแต่ละประโยคมีเพียงแค่ ซับเจ็กต์-เพรดิเคต-อ็อบเจ็กต์ แต่สามารถเชื่อมต่อกันเป็นโครงสร้าง directed graph ที่เรียกว่า RDF กราฟ ซึ่งมีความสัมพันธ์เชื่อมโยงกันแบบมีความหมายตามหลักการของซีแมนติกเว็บ และลิงก์เดต้า ทำให้แอปพลิเคชันทางด้านซีแมนติกเว็บต่าง ๆ สามารถบริโภคข้อมูลระดับ 5 ดาวของ TULIP ได้เช่นเดียวกับเดต้าเซ็ตอื่น ๆ

แต่ที่พิเศษนอกเหลือไปจากนั้น ข้อมูลตารางและรายการที่ใช้ TULIP RDF Schema จะมีคุณสมบัติสำคัญเพิ่มอีก 3 ข้อ ข้อแรกคือสามารถแปลงสภาพกลับมาเป็นตารางและรายการดั้งเดิมโดยที่ยังคงรักษาโครงสร้างเชิง semantic ของตารางและรายการเดิมไว้ได้อย่างครบถ้วน ส่วนข้อถัดมาค่อนข้างจะดูขัดแย้งกันกับข้อแรก คือข้อมูล TULIP สามารถจะถูก “มอง” ให้เป็นโครงสร้างในรูปแบบเดียวกันแบบกลมกลืนโดยแยกไม่ออกว่าข้อมูลนั้นมาจากตารางหรือรายการนั่นหมายความว่ามันถูก unified ให้อยู่ในโครงสร้างรูปแบบเดียวกันไปแล้ว ประโยชน์ที่ได้คือ เราสามารถมองข้อมูลโดยที่ไม่สนใจที่มาว่าเดิมมันมาจากรูปแบบชนิดไหน แต่เราสามารถเลือกที่จะมองมันใหม่ได้ในแบบที่เรา

ต้องการ ยกตัวอย่างเช่น เราสามารถมองข้อมูลที่ดั้งเดิมมันมีที่มาจากตารางแล้วนำมันมาแสดงใหม่ในรูปแบบของรายการ หรือในทางกลับกัน เราสามารถนำข้อมูลที่ดั้งเดิมมันมีที่มาจากรายการแต่เราจะนำมันมาแสดงในรูปแบบตาราง หรือไม่เราก็อาจจะนำมาแสดงผลผสมผสานกันในทั้งสองรูปแบบ หรือแม้กระทั่งนำข้อมูลทั้งหมดไปแสดงในรูปแบบอื่น ๆ ที่แตกต่างออกไปโดยสิ้นเชิง เช่น ชาร์ตหรือไดอะแกรมรูปแบบต่าง ๆ ดังนั้น หากเราต้องการสร้าง infographics จากข้อมูล TULIP เราสามารถที่จะสร้างมันในแบบ dynamic และเปลี่ยนรูปแบบการแสดงผลข้อมูลได้อย่างอิสระในหน้าตาที่เราต้องการแบบสด ๆ หรือ on the fly

คุณสมบัติพิเศษข้อสุดท้ายสืบเนื่องมาจากข้อที่แล้ว ข้อมูล TULIP เป็นข้อมูลที่อยู่ในรูปแบบ multi-dimensional array และสามารถจัดเก็บข้อมูลได้ทั้งในรูปแบบ column-major (columnar) หรือ row-major (row-based) โดยที่เราสามารถใช้หลักการของ Data Warehouse และ Online Analytical Processing (OLAP) มาบริหารจัดการมันได้ เช่นใช้ operation พื้นฐานทั้ง 3 อย่างของ Data Warehouse คือ roll-up, drill-down และ slicing/dicing ซึ่งหมายความว่าเราสามารถทำ pivot คัดกรอง หรือพลิกมุมมองของข้อมูลที่อยู่ในรูปแบบ TULIP แบบไหนก็ได้ตามที่เราต้องการ โดยที่ยังคงคุณสมบัติของ 5-star Open Data ไว้ได้อย่างครบถ้วน คือข้อมูลเหล่านั้นถูกเชื่ยมโยงกันแบบกระจาย หรือ distributed data โดยหลักการของ Linked Open Data

1. <u>TULIP Interchangeable property</u>

คุณสมบัติ Interchangeable หรือการสลับไปกลับได้อย่างครบถ้วนสมบูรณ์ของ TULIP ในการที่จะเก็บรักษาโครงสร้างเดิมของตารางและรายการต้นทางไว้ได้อย่างครบถ้วนในเชิง semantic structure เช่น ข้อความ, โครงสร้างของตารางและหัวตาราง ลำดับชั้นของรายการ (โดยจะรวมถึงสิ่งที่เกี่ยวกับรูปแบบบางประเภท เช่น spanning cells แต่ไม่รวมสิ่งที่เกี่ยวกับการตกแต่ง ตัวอย่างเช่น สไตล์รูปแบบของตัวอักษร เช่น ตัวหนา, ตัวเอียง, สี, ขอบ, พื้นหลัง ฯลฯ) ดังนั้นเราจึงสามารถสร้างตารางและรายการต้นฉบับได้จากชุดของ RDF ทริปเปิลที่อยู่ในรูปแบบของ TULIP RDF Schema

การทำแบบนี้ได้เนื่องจาก TULIP vocabulary มีชุดของพร็อปเพอร์ตี้และคลาสจำนวนหนึ่งในการบันทึกตัวเนื้อหาข้อมูล และรูปแบบของเนื้อหา เช่นแยกเป็นประเภท ตาราง, สดมภ์, แถว, เซลล์, รายการ, ไอเท็มของรายการ ฯลฯ รวมถึงลักษณะเฉพาะต่าง ๆ เช่น หัวตาราง, spanning cells, รายการชนิดมีลำดับขั้น เป็นต้น

2. TULIP Unified property

คุณสมบัติ Unified หรือการเป็นหนึ่งเดียว หมายถึง TULIP เก็บโครงสร้างของทั้งตารางและรายการไว้ในรูปแบบเดียวกัน เป็นโครงสร้างต้นไม้ hierarchical tree-like structure ซึ่งการเก็บโครงสร้างแบบนี้ด้วยชุดของ RDF ทริปเปิล ซึ่งมีข้อจำกัดคือ สามารถ represent ได้เพียงแค่ directed graph ที่ไม่มี order หรือ precedence ระหว่าง sibling nodes จะต้องมีการสร้างพร็อปเพอร์ตี้ชุดหนึ่งขึ้นมา เพื่อจำลองหรือเลียนแบบโครงสร้างที่มีลำดับชั้นและอันดับก่อนหลังของโหนดในลำดับชั้นเดียวกันขึ้นมา ซึ่งจะมีลักษณะคล้ายกับคุณสมบัติอย่างหนึ่งของ RDF ที่เรียกว่า RDF container แต่จะมีความเฉพาะเจาะจงมากกว่า

ดังนั้นแม้ว่าข้อมูลใน TULIP จะถูกระบุแยกแยะไว้อย่างละเอียดว่า เนื้อหาตัวไหนเป็นประเภทอะไรเพื่อให้สามารถทำการสร้างย้อนกลับตารางและรายการต้นฉบับให้เหมือนเดิมได้ในเชิง semantic ตามคุณสมบัติ Interchangeable แต่ในทางกลับกันเนื่องจาก TULIP เก็บโครงสร้างทุกอย่างไม่ว่าจะเป็นตารางหรือรายการ ในแบบต้นไม้ลักษณะเดียวกันตามคุณสมบัติ Unified หากคิวรีตามโครงสร้างนี้เราก็จะสามารถปรับเปลี่ยนรูปแบบผลลัพธ์ด้วยการนำเสนอข้อมูลในโครงสร้างใหม่ให้มีความแตกต่างไปจากรูปแบบดั้งเดิมของต้นฉบับได้ หมายความว่าเราสามารถจะแปลงรายการเป็นตาราง ปรับมุมมองเปลี่ยนไปเปลี่ยนมาได้แบบสด ๆ หรือ on the fly โดยขึ้นอยู่กับแอปพลิเคชันในการนำเสนอข้อมูลที่เข้าใจโครงสร้างของ TULIP แต่ถึงแม้หากเป็นแอปพลิเคชันซีแมนติกพื้นฐานที่ไม่เข้าใจ TULIP RDF Schema ก็ยังสามารถนำข้อมูลจาก TULIP ไปใช้ได้ในระดับทั่วไป เนื่องจากเนื้อหาของ TULIP เก็บด้วยพร็อปเพอร์ตี้มาตรฐาน เช่น rdf:type และ rdfs:label และถึงแม้ว่าจะมีพร็อปเพอร์ตี้พิเศษเพื่อจำลองโครงสร้างต้นไม้ แต่โครงสร้างนั้นก็ซ้อนอยู่บน RDF กราฟมาตรฐานแอปพลิเคชันซีแมนติกเว็บทั่วไปจึงสามารถทำ graph traversal และได้เนื้อหากลับมาครบถ้วนเพียงแต่จะขาดคุณสมบัติเฉพาะของ TULIP ไปบางอย่าง

3. TULIP Pivotal property

คุณสมบัติ Pivotal หรือการพลิกแพลงเปลี่ยนแปลงมุมมอง ทำได้เนื่องจาก TULIP มีโมเดลโครงสร้างอีกประเภทหนึ่งในการจำลอง multi-dimensional array บน RDF กราฟซึ่งทำให้เราสามารถคิวรีเข้าถึงเนื้อหาได้อย่างเฉพาะเจาะจงได้ทุกขนาดและมิติในคิวรีเดียว ยิ่งไปกว่านั้น การสร้างโมเดลแบบนี้เราสามารถประยุกต์ใช้ operation ต่าง ๆ ของ OLAP ในลักษณะ Data Warehouse ได้ เช่นการ roll-up ข้อมูลทั้งกลุ่มในชุดเดียวกัน drill-down ข้อมูลลึกลงไปในแต่ละ

ชั้น หรือแม้การทำ slicing เพื่อตัดเฉพาะแกนของเนื้อหา multi-dimensional มาทำ dicing เพื่อหมุนปรับเปลี่ยนมุมมอง ซึ่งเราสามารถทำ operation เหล่านี้ได้โดยที่ข้อมูลต้นฉบับไม่เสียหาย

ส่วนที่สำคัญคือ ผู้วิจัยได้ทำการจำลอง multi-dimensional array โดยการนำ RDF list มาใช้เป็นอาร์เรย์ขนาด 1 มิติ เพื่อเก็บ subscript แต่ละมิติของ multi-dimensional array ขนาด n มิตินั่นเอง ซึ่งนี่ถือเป็นแนวคิดสำคัญอย่างหนึ่งของ TULIP ในการจำลองโมเดลรูปแบบนี้บน RDF ซึ่งผู้วิจัยได้ประยุกต์ใช้คุณสมบัติหนึ่งของ RDF ที่เรียกว่า RDF collection หรือ RDF list โดยการนำ subscript ของแต่ละมิติ มาเป็นสมาชิกแต่ละลำดับของ collection แล้วนำ collection นี้ไปฝังไว้กับทุกโหนดของ TULIP ทำให้สามารถเข้าถึงได้ทุกโหนดโดยใช้ SPARQL คิวรี subscript ที่ต้องการโดยการแมทช์กับ collection ที่ตรงกัน ปัญหาอยู่ตรงข้อจำกัดของ SPARQL ที่ไม่มีความสามารถในการคิวรี RDF collection ได้โดยตรง (ทำให้เกิดงานวิจัยอีกหลายแนวทางในการสร้างสิ่งที่จะมาทดแทน RDF collection ตามมาตรฐาน เช่น Ordered List Ontology[99] และ Collections Ontology[100] หรือแนวคิดที่จะปรับปรุง SPARQL ให้เข้าถึง RDF collection ได้สะดวกขึ้น เช่น ข้อเสนอแนะของ Leigh และ Wood[101]) ทางแก้ไขปัญหาเฉพาะหน้าระหว่างรอ SPARQL เวอร์ชันถัดไปที่อาจจะมีการปรับปรุงในเรื่องนี้ คือต้องประยุกต์ใช้คุณสมบัติที่เรียกว่า property path ที่มีใน SPARQL 1.1 มาใช้เพื่อเข้าถึง collection item แต่ละตัว

TULIP RDFS Vocabulary Specification

หัวข้อนี้จะอธิบายถึงไวยากรณ์ของ TULIP ภาพที่ 38 แสดงให้เห็นบางส่วนของ TULIP vocabulary เขียนด้วย RDF Schema ซึ่งแท้จริงแล้วตัว RDFS ก็คือชุดของ RDF ทริปเปิลเช่นกัน โดยถูกเขียนเป็น Turtle ภายใต้ namespace http://purl.org/tulip/ns# แต่ในภาพนี้นำมาแสดงให้ดูในรูปแบบ railroad model หรือแผนภาพรางรถไฟไวยากรณ์ในลักษณะที่คล้ายกับภาษาโปรแกรมมิ่งเพื่อให้เห็นลักษณะของ TULIP vocabulary ได้โดยง่าย รายละเอียดทั้งหมดอยู่ในเว็บเพจของ TULIP vocabulary specification ที่ http://purl.org/tulip/ ซึ่งจะมีการอัพเดทเวอร์ชันที่พัฒนาเพิ่มเติมขึ้นในอนาคต

1. ส่วนหัวของ TULIP RDF Schema vocabulary

   ประกอบไปด้วยส่วน namespace prefixes ดังนี้

```
@prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl:  <http://www.w3.org/2002/07/owl#> .
```

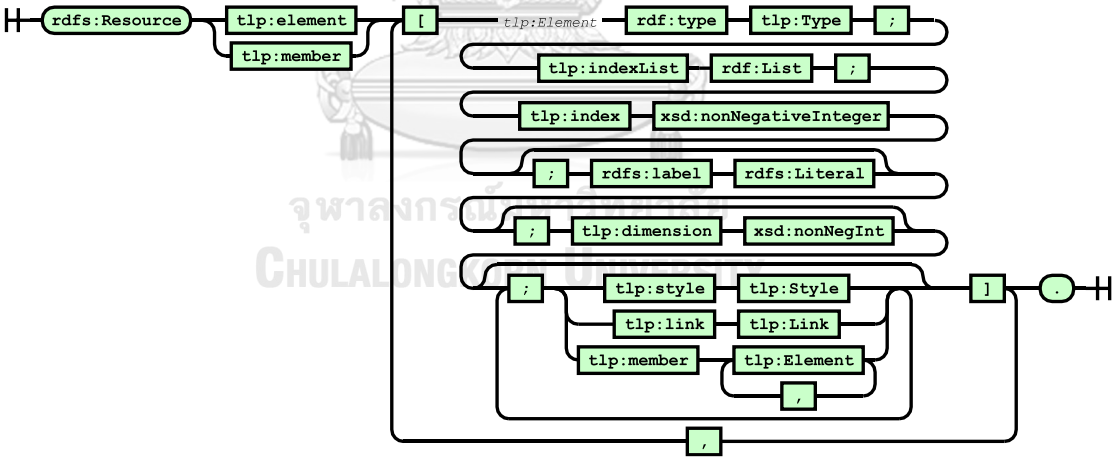
```
@prefix dc:   <http://purl.org/dc/elements/1.1/> .
@prefix xsd:  <http://www.w3.org/2001/XMLSchema#> .
@prefix tlp:  <http://purl.org/tulip/ns#> .
```

ซึ่งทั้งหมดนี้คือ namespace prefixes ที่จำเป็นต้องใช้ในชุดข้อมูลของ TULIP ทั้งในส่วนของ ซับเจ็กต์, เพรดิเคต และอ็อบเจ็กต์ ถัดมาเป็นส่วนอธิบายรายละเอียดของตัว vocabulary ดังนี้

```
<http://purl.org/tulip/ns#> a owl:Ontology ;
    dc:title       """TULIP - Table/List Interchangeable, Unified,
                      Pivotal vocabulary""" ;
    dc:date          "2019-12-20" ;
    owl:versionInfo  "1.0" ;
    dc:creator       "Julthep Nandakwang" ;
    dc:description """This is the RDF Schema for the TULIP vocabulary
                      terms in the TULIP namespace.""" .
```

2. พร็อปเพอร์ตี้ของ TULIP RDF Schema vocabulary

ถัดมาจะเป็นส่วนอธิบายพร็อปเพอร์ตี้แต่ละตัวที่กำหนดขึ้นใน TULIP vocabulary ที่ใช้ในการขยายความโหนดรากของ TULIP ที่เป็นชนิด rdfs:Resource และมีโหนดลูกหลานเป็น blank node ชนิด tlp:Element เชื่อมต่อกันดังที่ได้อธิบายไว้ในหัวข้อก่อนหน้านี้ ซึ่งในภาพที่ 38 เป็นแผนภาพไวยากรณ์ที่แสดงถึงพร็อปเพอร์ตี้ทั้งหมดที่มีให้ใช้ได้ โดยจะอธิบายแต่ละตัวดังนี้



ภาพที่ 38 แผนภาพไวยากรณ์ RDFS ของพร็อปเพอร์ตี้ tlp:element และ tlp:member

2.1 พร็อปเพอร์ตี้ tlp:indexList ของโดเมน rdfs:Resource

เป็นพร็อปเพอร์ตี้ที่ใช้ในการกำหนด indexList ของโหนดแต่ละ element ใน IL model โดยมีเรนจ์เป็น rdf:List (คือคลาสของ RDF collection หรือ RDF list) ในกรณีของ tlp:indexList จะเริ่มต้นด้วยเครื่องหมายวงเล็บเปิด ตามด้วยชุดของตัวเลขจำนวนเต็มบวกที่ใช้กำหนดตำแหน่งของโหนดในรูปแบบ multi-dimensional array index และปิดท้ายด้วยตัวเลข 0 ตามด้วยเครื่องหมาย

วงเล็บปิด พร็อปเพอร์ตี้ tlp:indexList แสดงเป็นแผนภาพไวยากรณ์ได้ดังภาพที่ 39 ซึ่งทั้งหมดกำหนดไว้ในรูปแบบ RDFS ดังนี้

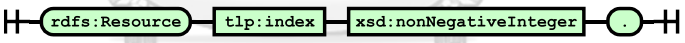
```
tlp:indexList  a rdf:Property ;
    rdfs:isDefinedBy <http://purl.org/tulip/ns#> ;
    rdfs:label   "indexList" ;
    rdfs:domain  rdfs:Resource ;
    rdfs:range   rdf:List ;
    rdfs:comment "RDF List represented position of the subject." .
```

ภาพที่ 39 แผนภาพไวยากรณ์ RDFS ของพร็อปเพอร์ตี้ tlp:indexList ของโดเมน rdfs:Resource

2.2 พร็อปเพอร์ตี้ tlp:index ของโดเมน rdfs:Resource

เป็นพร็อปเพอร์ตี้ที่ใช้ในการกำหนด index ของโหนดแต่ละ member ใน IM model โดยมีเรนจ์เป็นตัวเลขจำนวนเต็มบวก พร็อปเพอร์ตี้ tlp:index แสดงเป็นแผนภาพไวยากรณ์ได้ดังภาพที่ 40 ซึ่งทั้งหมดกำหนดในรูปแบบ RDFS ดังนี้

```
tlp:index      a rdf:Property ;
    rdfs:isDefinedBy <http://purl.org/tulip/ns#> ;
    rdfs:label   "index" ;
    rdfs:domain  rdfs:Resource ;
    rdfs:range   xsd:nonNegativeInteger ;
    rdfs:comment "Order number of the subject." .
```

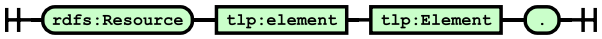


ภาพที่ 40 แผนภาพไวยากรณ์ RDFS ของพร็อปเพอร์ตี้ tlp:index ของโดเมน rdfs:Resource

2.3 พร็อปเพอร์ตี้ tlp:element ของโดเมน rdfs:Resource

เป็นพร็อปเพอร์ตี้ที่ใช้ในการระบุ element แต่ละตัวใน IL model โดยมีเรนจ์เป็น tlp:Element (ซึ่งเป็นชนิด rdfs:Class เป็นคนละตัวกันกับ tlp:element ที่เป็นชนิด rdf:Property) พร็อปเพอร์ตี้ tlp:element แสดงเป็นแผนภาพไวยากรณ์ได้ดังภาพที่ 41 ซึ่งทั้งหมดกำหนดในรูปแบบ RDFS ไว้ดังนี้

```
tlp:element    a rdf:Property ;
    rdfs:isDefinedBy <http://purl.org/tulip/ns#> ;
    rdfs:label   "element" ;
    rdfs:domain  rdfs:Resource ;
    rdfs:range   tlp:Element ;
    rdfs:comment "An element of the subject resource in IL model." .
```



ภาพที่ 41 แผนภาพไวยากรณ์ RDFS ของพร็อปเพอร์ตี้ tlp:element ของโดเมน rdfs:Resource

2.4 พร็อปเพอร์ตี้ tlp:member ของโดเมน rdfs:Resource

เป็นพร็อปเพอร์ตี้ที่ใช้ในการระบุ member แต่ละตัวใน IM model โดยมีเรนจ์เป็น tlp:Element พร็อปเพอร์ตี้ tlp:member แสดงเป็นแผนภาพไวยากรณ์ได้ดังภาพที่ 42 ซึ่งทั้งหมดกำหนดในรูปแบบ RDFS ไว้ดังนี้

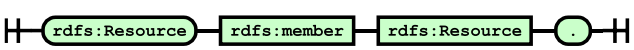
```
tlp:member     a rdf:Property ;
    rdfs:isDefinedBy <http://purl.org/tulip/ns#> ;
    rdfs:label   "member" ;
    rdfs:domain  rdfs:Resource ;
    rdfs:range   tlp:Element ;
    rdfs:comment "A member of the subject resource in IM model." .
```

ภาพที่ 42 แผนภาพไวยากรณ์ RDFS ของพร็อปเพอร์ตี้ tlp:member ของโดเมน rdfs:Resource

2.5 พร็อปเพอร์ตี้ rdfs:member ของโดเมน rdfs:Resource

เป็นพร็อปเพอร์ตี้ที่ใช้ในการระบุ member แต่ละตัวใน IM model โดยมีเรนจ์เป็น rdfs:Resource นอกเหนือจากพร็อปเพอร์ตี้ tlp:member ที่สร้างขึ้นใหม่ใน TULIP RDFS vocabulary เราสามารถเลือกใช้ rdfs:member ที่เป็นพร็อปเพอร์ตี้มาตรฐานที่แอปพลิเคชันของซีแมนติกเว็บส่วนใหญ่เลือกใช้ในการระบุสมาชิกโหนดลูก ดังนั้นแอปพลิเคชันต่าง ๆ ถึงแม้จะไม่รู้จัก TULIP RDF Schema ก็สามารถค้นหาสมาชิกโหนดลูก และนำข้อมูลไปใช้ได้ทันทีตามโครงสร้างแบบ hierarchical ตามที่ได้กล่าวไว้ในหัวข้อที่ผ่านมา พร็อปเพอร์ตี้ rdfs:member แสดงเป็นแผนภาพไวยากรณ์ได้ดังภาพที่ 43 ซึ่งทั้งหมดกำหนดในรูปแบบ RDFS ไว้แล้วที่ namespace http://www.w3.org/1999/02/22-rdf-syntax-ns# ดังนี้


```
rdfs:member    a rdf:Property ;
    rdfs:isDefinedBy <http://www.w3.org/2000/01/rdf-schema#> ;
    rdfs:label   "member" ;
    rdfs:domain  rdfs:Resource ;
    rdfs:range   rdfs:Resource ;
    rdfs:comment "A member of the subject resource." .
```

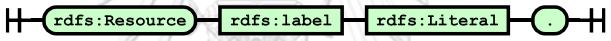


ภาพที่ 43 แผนภาพไวยากรณ์ RDFS ของพร็อปเพอร์ตี้ rdfs:member ของโดเมน rdfs:Resource

2.6 พร็อปเพอร์ตี้ rdfs:label ของโดเมน rdfs:Resource

เป็นพร็อปเพอร์ตี้ที่ใช้เก็บเนื้อความตัวอักษรของโหนด มีเรนจ์เป็น rdfs:Literal ในกรณีนี้แทนที่จะสร้างพร็อปเพอร์ตี้ชนิดใหม่ขึ้นมาใน TULIP RDFS vocabulary ผู้วิจัยเลือกใช้ rdfs:label ที่เป็นพร็อปเพอร์ตี้มาตรฐานที่แอปพลิเคชันของซีแมนติกเว็บส่วนใหญ่เลือกใช้ในการเก็บและแสดงผลเนื้อความตัวอักษรหลักของโหนด ดังนั้นแอปพลิเคชันต่าง ๆ ถึงแม้จะไม่รู้จัก TULIP RDF Schema ก็สามารถทำ graph traversal ไปตามแต่ละโหนดและนำข้อมูลไปใช้ได้ทันทีตามโครงสร้างแบบ hierarchical ตามที่ได้กล่าวไว้ในหัวข้อที่ผ่านมา พร็อปเพอร์ตี้ rdfs:label แสดงเป็นแผนภาพไวยากรณ์ได้ดังภาพที่ 44 พร็อปเพอร์ตี้ rdfs:label ถูกกำหนดในรูปแบบ RDFS ไว้แล้วที่ namespace http://www.w3.org/2000/01/rdf-schema# ดังนี้

```
rdfs:label     a rdf:Property ;
    rdfs:isDefinedBy <http://www.w3.org/2000/01/rdf-schema#> ;
    rdfs:label   "label" ;
    rdfs:domain  rdfs:Resource ;
    rdfs:range   rdfs:Literal ;
    rdfs:comment "A human-readable name for the subject." .
```

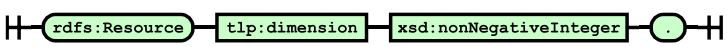


ภาพที่ 44 แผนภาพไวยากรณ์ RDFS ของพร็อปเพอร์ตี้ rdfs:label ของโดเมน rdfs:Resource

2.7 พร็อปเพอร์ตี้ tlp:dimension ของโดเมน rdfs:Resource

เป็นพร็อปเพอร์ตี้ที่ใช้ระบุมิติระดับชั้นความลึกของลูกหลาน มีเรนจ์เป็นตัวเลขจำนวนเต็มบวก ตัวนี้เป็น optional property ของ TULIP มีไว้เพื่ออำนวยความสะดวกในการใช้งาน ซึ่งจะกำหนดหรือไม่ก็ได้ เพราะเราสามารถใช้งาน TULIP ได้โดยไม่ต้องรู้ระดับความลึกของโหนดอย่างชัดเจนก่อนล่วงหน้า พร็อปเพอร์ตี้ tlp:dimension แสดงเป็นแผนภาพไวยากรณ์ได้ดังภาพที่ 45 ซึ่งทั้งหมดกำหนดในรูปแบบ RDFS ไว้ดังนี้


```
tlp:dimension  a rdf:Property ;
    rdfs:isDefinedBy <http://purl.org/tulip/ns#> ;
    rdfs:label   "dimension" ;
    rdfs:domain  rdfs:Resource ;
    rdfs:range   xsd:nonNegativeInteger ;
    rdfs:comment "(Optional) depth of the subject’s element(s)." .
```

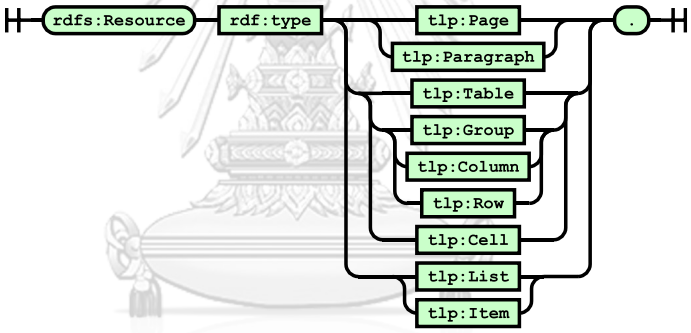


ภาพที่ 45 แผนภาพไวยากรณ์ RDFS ของพร็อปเพอร์ตี้ tlp:dimension ของโดเมน rdfs:Resource

2.8 พร็อปเพอร์ตี้ rdf:type ของโดเมน rdfs:Resource

เป็นพร็อปเพอร์ตี้ที่ใช้ระบุชนิดของโหนด มีเรนจ์เป็น rdfs:Class ในกรณีนี้แทนที่จะสร้างพร็อปเพอร์ตี้ชนิดใหม่ขึ้นมาใน TULIP RDFS vocabulary ผู้วิจัยเลือกใช้ rdf:type ที่เป็นพร็อปเพอร์ตี้ที่ใช้กันเป็นมาตรฐาน และยังได้ความสะดวกในการใช้ syntactic sugar โดยการระบุเพียงแค่ shorthand หรือตัวอักษรย่อ a แทนพร็อปเพอร์ตี้ rdf:type ใน Turtle serialization[20] พร็อปเพอร์ตี้ rdf:type และ rdfs:Class ของเรนจ์ที่เป็นไปได้ทั้งหมดใน TULIP element type แสดงเป็นแผนภาพไวยากรณ์ได้ดังภาพที่ 46 ซึ่งทั้งหมดกำหนดในรูปแบบ RDFS ไว้แล้วที่ namespace http://www.w3.org/1999/02/22-rdf-syntax-ns# ดังนี้

```
rdf:type       a rdf:Property ;
    rdfs:isDefinedBy <http://www.w3.org/1999/02/22-rdf-syntax-ns#> ;
    rdfs:label   "type" ;
    rdfs:domain  rdfs:Resource ;
    rdfs:range   rdfs:Class ;
    rdfs:comment "The subject is an instance of a class." .
```



ภาพที่ 46 แผนภาพไวยากรณ์ RDFS ของพรอบเพอร์ตี้ rdf:type ของโดเมน rdfs:Resource

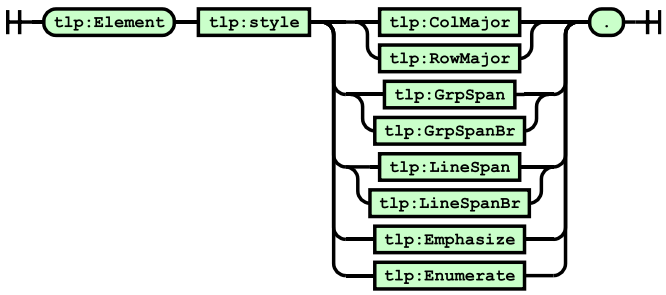
2.9 พร็อปเพอร์ตี้ tlp:style ของโดเมน tlp:Element

เป็นพร็อปเพอร์ตี้ที่ใช้ระบุรูปแบบของโหนด มีเรนจ์เป็น tlp:Style (ซึ่งเป็นชนิด rdfs:Class เป็นคนละตัวกันกับ tlp:style ที่เป็นชนิด rdf:Property) พร็อปเพอร์ตี้ tlp:style และ rdfs:Class ของเรนจ์ที่เป็นไปได้ทั้งหมดใน TULIP element style แสดงเป็นแผนภาพไวยากรณ์ได้ดังภาพที่ 47 ซึ่งทั้งหมดกำหนดในรูปแบบ RDFS ไว้ดังนี้

```
tlp:style      a rdf:Property ;
    rdfs:isDefinedBy <http://purl.org/tulip/ns#> ;
    rdfs:label   "style" ;
    rdfs:domain  tlp:Element ;
    rdfs:range   tlp:Style ;
```

[20] Proposed Turtle Specification – compared to N-Triples. W3C. Available from: https://www.w3.org/2010/01/Turtle/#sec-diff-ntriples

```
    rdfs:comment "(Optional) TULIP style of the subject." .
```

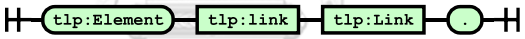


ภาพที่ 47 แผนภาพไวยากรณ์ RDFS ของพร็อปเพอร์ตี้ tlp:style ของโดเมน tlp:Element

2.10 พร็อปเพอร์ตี้ tlp:link ของโดเมน tlp:Element

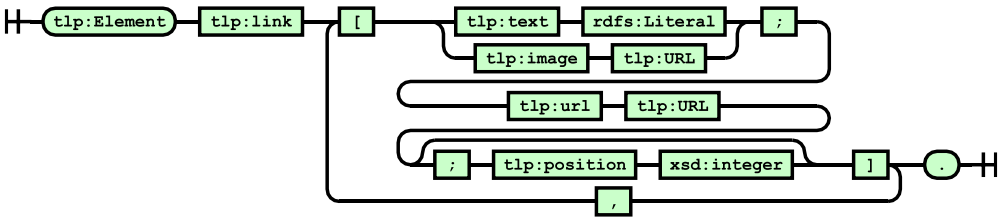
เป็นพร็อปเพอร์ตี้ที่ใช้ระบุลิงก์ของแต่ละ tlp:Element มีเรนจ์เป็น tlp:Link พร็อปเพอร์ตี้ tlp:link แสดงเป็นแผนภาพไวยากรณ์ได้ดังภาพที่ 48 ซึ่งทั้งหมดกำหนดในรูปแบบ RDFS ไว้ดังนี้

```
tlp:link       a rdf:Property ;
    rdfs:isDefinedBy <http://purl.org/tulip/ns#> ;
    rdfs:label   "link" ;
    rdfs:domain  tlp:Element ;
    rdfs:range   tlp:Link ;
    rdfs:comment "All link(s) of the subject." .
```



ภาพที่ 48 แผนภาพไวยากรณ์ RDFS ของพร็อปเพอร์ตี้ tlp:link ของโดเมน tlp:Element

ซึ่ง tlp:Link ผู้วิจัยจะกำหนดเป็น blank node ที่มีพร็อปเพอร์ตี้ย่อยอีก 4 ตัว แสดงเป็นแผนภาพไวยากรณ์ได้ดังภาพที่ 49 คืออันดับแรกเลือกระหว่าง tlp:text หรือ tlp:image ตัวใดตัวหนึ่ง ถัดมาให้ระบุ tlp:url ส่วน tlp:position จะมีหรือไม่มีก็ได้ ซึ่งรายละเอียดของแต่ละพร็อปเพอร์ตี้จะเป็นดังนี้



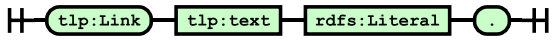
ภาพที่ 49 แผนภาพไวยากรณ์ RDFS ของส่วนประกอบพร็อปเพอร์ตี้ย่อยในเรนจ์ tlp:Link

2.11 พร็อปเพอร์ตี้ tlp:text ของโดเมน tlp:Link

เป็นพร็อปเพอร์ตี้ที่ใช้กำหนดส่วนของเนื้อความตัวอักษรของลิงก์ มีเรนจ์เป็น rdfs:Literal ซึ่งเมื่อนำข้อความของลิงก์ไปเทียบกับข้อความทั้งหมดของโหนดที่ระบุไว้ที่ rdfs:label จะทำให้เรา

สามารถนำไปใช้แสดงผลเป็นไฮเปอร์ลิงก์บนข้อความของโหนดได้ทันที พร็อปเพอร์ตี้ tlp:text แสดงเป็นแผนภาพไวยากรณ์ได้ดังภาพที่ 50 ซึ่งทั้งหมดกำหนดในรูปแบบ RDFS ไว้ดังนี้

```
tlp:text       a rdf:Property ;
    rdfs:isDefinedBy <http://purl.org/tulip/ns#> ;
    rdfs:label   "text" ;
    rdfs:domain  tlp:Link ;
    rdfs:range   rdfs:Literal ;
    rdfs:comment "Text of a link." .
```



ภาพที่ 50 แผนภาพไวยากรณ์ RDFS ของพร็อปเพอร์ตี้ tlp:text ของโดเมน tlp:Link

2.12 พร็อปเพอร์ตี้ tlp:image ของโดเมน tlp:Link

เป็นพร็อปเพอร์ตี้ที่ใช้กำหนดรูปภาพของลิงก์ มีเรนจ์เป็น tlp:URL ที่เป็นที่อยู่ของตัวรูปภาพ (ไม่ใช่ที่อยู่ปลายทางของลิงก์) พร็อปเพอร์ตี้ tlp:image แสดงเป็นแผนภาพไวยากรณ์ได้ดังภาพที่ 51 ซึ่งทั้งหมดกำหนดในรูปแบบ RDFS ไว้ดังนี้

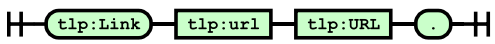
```
tlp:image      a rdf:Property ;
    rdfs:isDefinedBy <http://purl.org/tulip/ns#> ;
    rdfs:label   "image" ;
    rdfs:domain  tlp:Link ;
    rdfs:range   tlp:URL ;
    rdfs:comment "Image URL of a link." .
```

ภาพที่ 51 แผนภาพไวยากรณ์ RDFS ของพร็อปเพอร์ตี้ tlp:image ของโดเมน tlp:Link

2.13 พร็อปเพอร์ตี้ tlp:url ของโดเมน tlp:Link

เป็นพร็อปเพอร์ตี้ที่ใช้ระบุที่อยู่ปลายทางของลิงก์ มีเรนจ์เป็น tlp:URL (ซึ่งเป็นชนิด rdfs:Class เป็นคนละตัวกันกับ tlp:url ที่เป็นชนิด rdf:Property) พร็อปเพอร์ตี้ tlp:url แสดงเป็นแผนภาพไวยากรณ์ได้ดังภาพที่ 52 ซึ่งทั้งหมดกำหนดในรูปแบบ RDFS ไว้ดังนี้

```
tlp:url        a rdf:Property ;
    rdfs:isDefinedBy <http://purl.org/tulip/ns#> ;
    rdfs:label   "url" ;
    rdfs:domain  tlp:Link ;
    rdfs:range   tlp:URL ;
    rdfs:comment "Target URL of a link." .
```

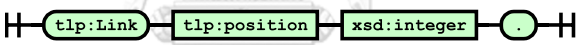


ภาพที่ 52 แผนภาพไวยากรณ์ RDFS ของพร็อปเพอร์ตี้ tlp:url ของโดเมน tlp:Link

2.14 พร็อปเพอร์ตี้ tlp:position ของโดเมน tlp:Link

เป็นพร็อปเพอร์ตี้ที่ใช้ระบุตำแหน่งของข้อความที่จะแสดงไฮเปอร์ลิงก์ ในกรณีที่มีข้อความเหมือนกันอยู่หลายตำแหน่งแต่ต้องการระบุตำแหน่งที่จะให้ขึ้นไฮเปอร์ลิงก์ตรงตำแหน่งที่เฉพาะเจาะจง มีเรนจ์เป็นเลขจำนวนเต็ม หากไม่กำหนดพร็อปเพอร์ตี้นี้ หรือระบุเป็น 0 จะแสดงผลไฮเปอร์ลิงก์ในทุก ๆ ข้อความที่แมทช์กับ tlp:text หรือหากระบุเป็นเลขติดลบ จะเริ่มนับตำแหน่งจากด้านหลังของข้อความ เป็น optional feature ที่เผื่อไว้ในกรณีที่มีความจำเป็นต้องใช้ พร็อปเพอร์ตี้ tlp:position แสดงเป็นแผนภาพไวยากรณ์ได้ดังภาพที่ 53 ซึ่งทั้งหมดกำหนดในรูปแบบ RDFS ไว้ดังนี้

```
tlp:position   a rdf:Property ;
    rdfs:isDefinedBy <http://purl.org/tulip/ns#> ;
    rdfs:label   "position" ;
    rdfs:domain  tlp:Link ;
    rdfs:range   xsd:integer ;
    rdfs:comment """(Optional) position of the link text in subject
                 resource, as number of occurrences in subject
                 resource starting from 1.
                 Omission of this property, or explicit 0, indicates
                 no fixed position will hyperlink all occurrences.
                 (Allow negative integer for starting from the end,
                 in case of any special need.)""" .
```



ภาพที่ 53 แผนภาพไวยากรณ์ RDFS ของพร็อปเพอร์ตี้ tlp:position ของโดเมน tlp:Link

3. คลาสของ TULIP RDF Schema vocabulary

ถัดมาเป็นส่วนของคลาสต่าง ๆ ที่ถูกกำหนดขึ้นใน TULIP RDFS vocabulary เพื่อนำไปใช้ร่วมกับพร็อปเพอร์ตี้ที่กล่าวมาในหัวข้อที่แล้ว

3.1 คลาส tlp:Element ที่ใช้สำหรับพร็อปเพอร์ตี้ tlp:element

เป็นคลาสที่ใช้กำหนดว่าโหนดนั้นเป็น blank node ที่เป็นโหนดลูกในโครงสร้างของ TULIP ทั้ง IL และ IM model คลาส tlp:Element ถูกกำหนดในรูปแบบ RDFS ไว้ดังนี้

```
tlp:Element       a rdfs:Class ;
    rdfs:isDefinedBy <http://purl.org/tulip/ns#> ;
    rdfs:label     "Element" ;
    rdfs:subClassOf rdfs:Resource ;
    rdfs:comment   "The class of TULIP element." .
```

3.2 คลาส tlp:Link ที่ใช้สำหรับพร็อปเพอร์ตี้ tlp:link

เป็นคลาสที่ใช้กำหนดว่าโหนดนั้นเป็น blank node ที่เก็บชุดพร็อปเพอร์ตี้ของแต่ละลิงก์ ถูกกำหนดในรูปแบบ RDFS ไว้ดังนี้

```
tlp:Link          a rdfs:Class ;
    rdfs:isDefinedBy <http://purl.org/tulip/ns#> ;
    rdfs:label     "Link" ;
    rdfs:subClassOf rdfs:Resource ;
    rdfs:comment   "The class of TULIP link." .
```

3.3 คลาส tlp:URL ที่ใช้สำหรับพร็อปเพอร์ตี้ tlp:image และ tlp:url

เป็นคลาสที่ใช้กำหนดว่าโหนดนั้นเป็น URL ซึ่งเป็นการระบุเฉพาะเจาะจงว่าเป็น URL ของรูปภาพและลิงก์ในรูปแบบเดียวกันกับ HTML (ถึงแม้ว่าทุกส่วนประกอบของโครงสร้าง RDF จะเป็น URI ทั้งหมดอยู่แล้วก็ตาม) คลาส tlp:URL ถูกกำหนดในรูปแบบ RDFS ไว้ดังนี้

```
tlp:URL           a rdfs:Class ;
    rdfs:isDefinedBy <http://purl.org/tulip/ns#> ;
    rdfs:label     "URL" ;
    rdfs:subClassOf rdfs:Resource ;
    rdfs:comment   "The class of resource URL." .
```

3.4 ซุปเปอร์คลาส tlp:Type ที่ใช้สำหรับคลาสกำหนดชนิดของ tlp:Element

เป็นซุปเปอร์คลาสที่ใช้กำหนดว่าโหนดนั้นเป็นคลาสชนิดหนึ่งของ TULIP element type ถูกกำหนดในรูปแบบ RDFS ไว้ดังนี้

```
tlp:Type          a rdfs:Class ;
    rdfs:isDefinedBy <http://purl.org/tulip/ns#> ;
    rdfs:label     "Type" ;
    rdfs:subClassOf rdfs:Class ;
    rdfs:comment   "The class of TULIP type." .
```

3.5 คลาส tlp:Page ที่ใช้สำหรับพร็อปเพอร์ตี้ rdf:type

เป็นคลาสที่ใช้กำหนดว่าโหนดนั้นเป็นอ็อบเจ็กต์หนึ่งของเอกสารทั้งหน้า เช่น หน้าของเว็บเพจ ถูกกำหนดในรูปแบบ RDFS ไว้ดังนี้

```
tlp:Page          a rdfs:Class ;
    rdfs:isDefinedBy <http://purl.org/tulip/ns#> ;
    rdfs:label     "Page" ;
    rdfs:subClassOf tlp:Type ;
    rdfs:comment """The TULIP class of document page,
                    e.g., from HTML <head> with <body>.""" .
```

3.6 คลาส tlp:Paragraph ที่ใช้สำหรับพร็อปเพอร์ตี้ rdf:type

เป็นคลาสที่ใช้กำหนดว่าโหนดนั้นเป็นอ็อบเจ็กต์หนึ่งของย่อหน้าเอกสาร ถูกกำหนดในรูปแบบ RDFS ไว้ดังนี้

```
tlp:Paragraph     a rdfs:Class ;
    rdfs:isDefinedBy <http://purl.org/tulip/ns#> ;
    rdfs:label     "Paragraph" ;
    rdfs:subClassOf tlp:Type ;
    rdfs:comment """The TULIP class of paragraph or text block,
                    e.g., from HTML <p> or <div>.""" .
```

3.7 คลาส tlp:Table ที่ใช้สำหรับพร็อปเพอร์ตี้ rdf:type

เป็นคลาสที่ใช้กำหนดว่าโหนดนั้นเป็นอ็อบเจ็กต์หนึ่งของตาราง ถูกกำหนดในรูปแบบ RDFS ไว้ดังนี้

```
tlp:Table         a rdfs:Class ;
    rdfs:isDefinedBy <http://purl.org/tulip/ns#> ;
    rdfs:label     "Table" ;
    rdfs:subClassOf tlp:Type ;
    rdfs:comment   "The TULIP class of table, e.g., HTML <table>." .
```

3.8 คลาส tlp:Group ที่ใช้สำหรับพร็อปเพอร์ตี้ rdf:type

เป็นคลาสที่ใช้กำหนดว่าโหนดนั้นเป็นอ็อบเจ็กต์หนึ่งของสดมภ์หรือแถวของตารางขึ้นอยู่กับว่าเก็บเป็นรูปแบบ column-major หรือ row-major เช่น หากเป็น column-major สดมภ์ของตาราง HTML จะมาจากการรวมกันของข้อมูลใน <th> หรือ <td> แท็ก ที่อยู่ในตำแหน่งเดียวกันของทุก ๆ <tr> แท็ก หรือหากเป็น row-major แถวของตาราง HTML จะมาจากข้อมูลใน <tr> แท็กทั้งแถวรวมกัน เป็นต้น

tlp:Group มีซับคลาสคือ tlp:Column และ tlp:Row เป็น optional คลาสที่มีเผื่อไว้ในกรณีที่ต้องการระบุเฉพาะเจาะจงว่าตารางเป็น column-major หรือ row-major อย่างชัดเจน อย่างไรก็ตามโครงสร้างของ TULIP ก็ยังเป็นโครงสร้างแบบเดียวกันคือเป็นลักษณะ group และ cell ซึ่งสามารถจะพลิกมุมมองเป็น row-major หรือ column-major ได้ทุกเมื่อขึ้นอยู่กับระบบที่นำข้อมูลไปใช้ คลาส tlp:Group ถูกกำหนดในรูปแบบ RDFS ไว้ดังนี้

```
tlp:Group         a rdfs:Class ;
    rdfs:isDefinedBy <http://purl.org/tulip/ns#> ;
    rdfs:label     "Group" ;
    rdfs:subClassOf tlp:Type ;
    rdfs:comment """The TULIP class of table column/row group
                    e.g. for column - combination of all HTML <th>
```

```
                or <td> in same position of each <tr>,
                and for row - from HTML <tr>.""" .
```

3.9 คลาส tlp:Column ที่ใช้สำหรับพร็อปเพอร์ตี้ rdf:type

เป็นซับคลาสของ tlp:Group ที่ใช้กำหนดว่าโหนดนั้นเป็นอ็อบเจ็กต์หนึ่งของสดมภ์ของตาราง เช่น หากเป็นสดมภ์ของตาราง HTML จะมาจากการรวมกันของข้อมูลใน <th> หรือ <td> แท็ก ที่อยู่ในตำแหน่งเดียวกันของทุก ๆ <tr> แท็ก เป็นต้น ซับคลาสนี้เป็น optional คลาสที่มีเผื่อไว้ในกรณีที่ต้องการใช้ tlp:Column แทน tlp:Group เพื่อระบุเฉพาะเจาะจงว่าตารางเป็น column-major อย่างชัดเจน อย่างไรก็ตามโครงสร้างของ TULIP ก็ยังเป็นโครงสร้างแบบเดียวกันคือเป็นลักษณะ group และ cell ซึ่งสามารถจะพลิกมุมมองเป็น row-major ได้ทุกเมื่อขึ้นอยู่กับระบบที่นำข้อมูลไปใช้ ซับคลาส tlp:Column กำหนดในรูปแบบ RDFS ไว้ดังนี้

```
tlp:Column        a rdfs:Class ;
    rdfs:isDefinedBy <http://purl.org/tulip/ns#> ;
    rdfs:label     "Column" ;
    rdfs:subClassOf tlp:Group ;
    rdfs:comment """The TULIP class of table column,
                    e.g., combination of all HTML <th> or <td>
                    in same position of each <tr>.""" .
```

3.10 คลาส tlp:Row ที่ใช้สำหรับพร็อปเพอร์ตี้ rdf:type

เป็นซับคลาสของ tlp:Group ที่ใช้กำหนดว่าโหนดนั้นเป็นอ็อบเจ็กต์ของแถวของตาราง เช่น หากเป็นแถวของตาราง HTML จะมาจากข้อมูลใน <tr> แท็กทั้งแถวรวมกัน เป็นต้น ซับคลาสนี้เป็น optional คลาสที่มีเผื่อไว้ในกรณีที่ต้องการใช้ tlp:Row แทน tlp:Group เพื่อระบุเฉพาะเจาะจงว่าตารางเป็น row-major อย่างชัดเจน อย่างไรก็ตามโครงสร้างของ TULIP ก็ยังเป็นโครงสร้างแบบเดียวกันคือเป็นลักษณะ group และ cell ซึ่งสามารถจะพลิกมุมมองเป็น column-major ได้ทุกเมื่อขึ้นอยู่กับระบบที่นำข้อมูลไปใช้ ซับคลาส tlp:Row กำหนดในรูปแบบ RDFS ไว้ดังนี้

```
tlp:Row           a rdfs:Class ;
    rdfs:isDefinedBy <http://purl.org/tulip/ns#> ;
    rdfs:label     "Row" ;
    rdfs:subClassOf tlp:Group ;
    rdfs:comment """The TULIP class of table row,
                    e.g., from HTML <tr>.""" .
```

3.11 คลาส tlp:Cell ที่ใช้สำหรับพร็อปเพอร์ตี้ rdf:type

เป็นคลาสที่ใช้กำหนดว่าโหนดนั้นเป็นอ็อบเจ็กต์ของเซลล์เดี่ยวของตาราง ถูกกำหนดในรูปแบบ RDFS ไว้ดังนี้

```
tlp:Cell          a rdfs:Class ;
    rdfs:isDefinedBy <http://purl.org/tulip/ns#> ;
    rdfs:label     "Cell" ;
    rdfs:subClassOf tlp:Type ;
    rdfs:comment """The TULIP class of table cell,
                    e.g., a single HTML <th> or <td>.""" .
```

3.12 คลาส tlp:List ที่ใช้สำหรับพร็อปเพอร์ตี้ rdf:type

เป็นคลาสที่ใช้กำหนดว่าโหนดนั้นเป็นอ็อบเจ็กต์ของรายการ ถูกกำหนดในรูปแบบ RDFS ไว้ดังนี้

```
tlp:List          a rdfs:Class ;
    rdfs:isDefinedBy <http://purl.org/tulip/ns#> ;
    rdfs:label     "List" ;
    rdfs:subClassOf tlp:Type ;
    rdfs:comment """The TULIP class of unordered/ordered list,
                    e.g., from HTML <ul> or <ol>.""" .
```

3.13 คลาส tlp:Item ที่ใช้สำหรับพร็อปเพอร์ตี้ rdf:type

เป็นคลาสที่ใช้กำหนดว่าโหนดนั้นเป็นอ็อบเจ็กต์ของรายการเดี่ยวหรือไอเท็มของรายการ ถูกกำหนดในรูปแบบ RDFS ไว้ดังนี้

```
tlp:Item          a rdfs:Class ;
    rdfs:isDefinedBy <http://purl.org/tulip/ns#> ;
    rdfs:label     "Item" ;
    rdfs:subClassOf tlp:Type ;
    rdfs:comment """The TULIP class of list item,
                    e.g., a single HTML <li>.""" .
```


3.14 ซุปเปอร์คลาส tlp:Style ที่ใช้สำหรับคลาสกำหนดรูปแบบของ tlp:Element

เป็นซุปเปอร์คลาสที่ใช้กำหนดว่าโหนดนั้นเป็นคลาสชนิดหนึ่งของ TULIP element style ถูกกำหนดในรูปแบบ RDFS ไว้ดังนี้

```
tlp:Style         a rdfs:Class ;
    rdfs:isDefinedBy <http://purl.org/tulip/ns#> ;
    rdfs:label     "Style" ;
    rdfs:subClassOf rdfs:Class ;
    rdfs:comment   "The class of TULIP style." .
```

3.15 คลาส tlp:Emphasize ที่ใช้สำหรับพร็อปเพอร์ตี้ tlp:style

เป็นคลาสที่ใช้กำหนดว่าโหนดนี้เป็นอ็อบเจ็กต์ของข้อมูลที่มีความเด่น เช่น หัวของตาราง ไม่ว่าจะเป็นในแนวตั้งหรือแนวนอน หรือเพื่อระบุว่าโหนดนี้ดั้งเดิมมาจาก HTML <th> แท็ก คลาส tlp:Emphasize กำหนดในรูปแบบ RDFS ไว้ดังนี้

```
tlp:Emphasize     a rdfs:Class ;
    rdfs:isDefinedBy <http://purl.org/tulip/ns#> ;
    rdfs:label     "Emphasize" ;
    rdfs:subClassOf tlp:Style ;
    rdfs:comment """The TULIP cell is emphasized style,
                    e.g., column or row header, HTML <th> cell.""" .
```

3.16 คลาส tlp:ColMajor ที่ใช้สำหรับพร็อปเพอร์ตี้ tlp:style

เป็นคลาสที่ใช้กำหนดว่าโหนดนั้นเป็นอ็อบเจ็กต์ของตารางที่เป็นแบบ column-major คือเริ่มแบ่ง group ด้วยสดมภ์ก่อนแล้วจึงค่อยลงไปในแต่ละ cell ตามแถวของสดมภ์ (ตามที่ได้อธิบายไว้ในหัวข้อก่อนหน้านี้ดังภาพที่ 27) ซึ่งทั้ง tlp:ColMajor และ tlp:RowMajor หากไม่ได้กำหนดมาหรือกำหนดมาทั้งคู่ จะขึ้นอยู่การโปรแกรมที่นำข้อมูลไปใช้ว่าได้มีการตั้งค่าปริยายไว้ในลักษณะไหนเป็นรูปแบบหลัก คลาส tlp:ColMajor ถูกกำหนดในรูปแบบ RDFS ไว้ดังนี้

```
tlp:ColMajor      a rdfs:Class ;
    rdfs:isDefinedBy <http://purl.org/tulip/ns#> ;
    rdfs:label     "ColMajor" ;
    rdfs:subClassOf tlp:Style ;
    rdfs:comment   "The TULIP table is in Column-Major." .
```

3.17 คลาส tlp:RowMajor ที่ใช้สำหรับพร็อปเพอร์ตี้ tlp:style

เป็นคลาสที่ใช้กำหนดว่าโหนดนั้นเป็นอ็อบเจ็กต์ของตารางที่เป็นแบบ row-major คือเริ่มแบ่ง group ด้วยแถวก่อนแล้วจึงค่อยกวาด cell ในแต่ละสดมภ์ทีละเซลล์ (ตามที่ได้อธิบายไว้ในหัวข้อก่อนหน้านี้ดังภาพที่ 28) ซึ่งทั้ง tlp:ColMajor และ tlp:RowMajor หากไม่ได้กำหนดมาหรือกำหนดมาทั้งคู่ จะขึ้นอยู่การโปรแกรมที่นำข้อมูลไปใช้ว่าได้มีการตั้งค่าปริยายไว้ในลักษณะไหนเป็นรูปแบบหลัก คลาส tlp:RowMajor กำหนดในรูปแบบ RDFS ไว้ดังนี้


```
tlp:RowMajor      a rdfs:Class ;
    rdfs:isDefinedBy <http://purl.org/tulip/ns#> ;
    rdfs:label     "RowMajor" ;
    rdfs:subClassOf tlp:Style ;
    rdfs:comment   "The TULIP table is in Row-Major." .
```

3.18 คลาส tlp:GrpSpan ที่ใช้สำหรับพร็อปเพอร์ตี้ tlp:style

เป็นคลาสที่ใช้กำหนดว่าโหนดนั้นเป็นอ็อบเจ็กต์ของเซลล์ที่มีการเชื่อม (หรือเรียกว่า spanned cell) มากกว่า 1 เซลล์ขึ้นไปในแนว group ดังข้อมูลในตารางตัวอย่างนี้

ราคาซื้อ		ราคาขาย	
ก่อน VAT	หลัง VAT	ก่อน VAT	หลัง VAT

จะถูกเก็บไว้ในตารางรูปแบบ column-major ในลักษณะนี้ (→ หมายถึงมีการระบุ tlp:GrpSpan เพิ่มไว้)

ราคาซื้อ →	ราคาซื้อ →	ราคาขาย →	ราคาขาย →
ก่อน VAT	หลัง VAT	ก่อน VAT	หลัง VAT

คลาส tlp:GrpSpan ถูกกำหนดในรูปแบบ RDFS ไว้ดังนี้

```
tlp:GrpSpan       a rdfs:Class ;
    rdfs:isDefinedBy <http://purl.org/tulip/ns#> ;
    rdfs:label     "GrpSpan" ;
    rdfs:subClassOf tlp:Style ;
    rdfs:comment """The TULIP cell is spanned in group,
                    e.g., HTML <th> or <td> with colspan attribute,
                    if the table is column-major.""" .
```

3.19 คลาส tlp:LineSpan ที่ใช้สำหรับพร็อปเพอร์ตี้ tlp:style

เป็นคลาสที่ใช้กำหนดว่าโหนดนั้นเป็นอ็อบเจ็กต์ของเซลล์ที่มีการเชื่อม (หรือเรียกว่า spanned cell) มากกว่า 1 เซลล์ขึ้นไปในแนว line คลาส tlp:LineSpan ถูกกำหนดในรูปแบบ RDFS ไว้ดังนี้

```
tlp:LineSpan      a rdfs:Class ;
    rdfs:isDefinedBy <http://purl.org/tulip/ns#> ;
    rdfs:label     "LineSpan" ;
    rdfs:subClassOf tlp:Style ;
    rdfs:comment """The TULIP cell is spanned in line,
                    e.g., HTML <th> or <td> with rowspan attribute,
                    if the table is column-major.""" .
```


3.20 คลาส tlp:GrpSpanBr ที่ใช้สำหรับพร็อปเพอร์ตี้ tlp:style

เป็นคลาสที่ใช้กำหนดว่าโหนดนั้นเป็นอ็อบเจ็กต์ของเซลล์ที่เป็นจุดสิ้นสุดของแต่ละ spanned cell ใช้ในกรณีที่มี spanned cell มากกว่า 1 spanned cell ที่มีข้อมูลเหมือนกันและอยู่ติดกันในแนว group ดังข้อมูลในตารางตัวอย่างนี้

ราคาขาย		ราคาขาย	
ก่อน S.charge	หลัง S,charge	ก่อน VAT	หลัง VAT

จะถูกเก็บไว้ในตารางรูปแบบ column-major ในลักษณะนี้ (→ หมายถึงมีการระบุ tlp:GrpSpan และ Br หมายถึง tlp:GrpSpanBr เพิ่มไว้)

ราคาขาย →	ราคาขาย→Br	ราคาขาย →	ราคาขาย→Br
ก่อน S.charge	หลัง S.charge	ก่อน VAT	หลัง VAT

คลาส tlp:GrpSpanBr ถูกกำหนดในรูปแบบ RDFS ไว้ดังนี้

```
tlp:GrpSpanBr     a rdfs:Class ;
    rdfs:isDefinedBy <http://purl.org/tulip/ns#> ;
    rdfs:label     "GrpSpanBr" ;
    rdfs:subClassOf tlp:Style ;
    rdfs:comment   "(Optional) To break cell in same GrpSpan." .
```

3.21 คลาส tlp:LineSpanBr ที่ใช้สำหรับพร็อปเพอร์ตี้ tlp:style

เป็นคลาสที่ใช้กำหนดว่าโหนดนั้นเป็นอ็อบเจ็กต์ของเซลล์ที่เป็นจุดสิ้นสุดของแต่ละ spanned cell ใช้ในกรณีที่มี spanned cell มากกว่า 1 spanned cell ที่มีข้อมูลเหมือนกันและอยู่ติดกันในแนว line ซึ่งทั้ง tlp:GrpSpanBr และ tlp:LineSpanBr ไม่มีความจำเป็นต้องระบุหากข้อมูลในเซลล์ถัดไปมีข้อมูลที่แตกต่างกันถึงแม้ว่าจะยังมีลักษณะเป็น spanned cell ก็ตาม ดังตัวอย่างที่ได้แสดงไว้ในคลาส tlp:GrpSpan คลาส tlp:LineSpanBr ถูกกำหนดในรูปแบบ RDFS ไว้ดังนี้

```
tlp:LineSpanBr    a rdfs:Class ;
    rdfs:isDefinedBy <http://purl.org/tulip/ns#> ;
    rdfs:label     "LineSpanBr" ;
    rdfs:subClassOf tlp:Style ;
    rdfs:comment   "(Optional) To break cell in same LineSpan." .
```

3.22 คลาส tlp:Enumerate ที่ใช้สำหรับพร็อปเพอร์ตี้ tlp:style

เป็นคลาสที่ใช้กำหนดว่าโหนดนี้เป็นอ็อบเจ็กต์ที่มีลำดับกำกับ เช่น เป็นรายการแบบมีตัวเลขหรือตัวอักษรกำกับลำดับ เป็นต้น หรือเพื่อระบุว่าที่โหนดนี้ดั้งเดิมเป็นรายการที่มาจาก HTML <ol> แท็ก คลาส tlp:Enumerate กำหนดในรูปแบบ RDFS ไว้ดังนี้

```
tlp:Enumerate     a rdfs:Class ;
    rdfs:isDefinedBy <http://purl.org/tulip/ns#> ;
    rdfs:label     "Enumerate" ;
    rdfs:subClassOf tlp:Style ;
    rdfs:comment """The TULIP list is enumerated style,
                    e.g., HTML <ol> ordered list.""" .
```

เปรียบเทียบ DOM กับ TULIP RDF

อย่างที่กล่าวไปแล้ว การเก็บโครงสร้างข้อมูลใน DOM จะเป็น hierarchy ที่อยู่ในโครงสร้างต้นไม้ ส่วน TULIP จะเป็นชิ้น ๆ ของ RDF ทริปเปิลที่เชื่อมกันเป็น RDF กราฟ

DOM เป็น document based คือเป็นข้อมูลรวมทั้งแผ่นหรือเอกสารมาทั้งชุด เช่น เปิดไฟล์ HTML มาจะเป็น <head> และ <body> เป็นอันดับแรก และใน <body> ก็จะมีแต่ละชั้นของโครงสร้างตาม hierarchy ของ HTML แท็กอ็อบเจ็กต์ แต่ TULIP จะเป็น element based คือเป็นชิ้นส่วนย่อย ๆ ซึ่งเป็นธรรมชาติของ RDF คือเป็นกราฟที่เชื่อมระหว่าง RDF ทริปเปิลย่อย ๆ เชื่อมถึงกัน

โครงสร้างการรีพรีเซนต์ข้อมูล 2 มิติอย่างเช่นตารางของ DOM จะเป็น static คือกำหนดเป็น row based ได้อย่างเดียวตามรูปแบบมาตรฐานของ HTML แต่ TULIP จะเป็น dynamic คือเรากำหนดเองได้ว่าจะใช้เป็น row-major หรือ column-major เนื่องจากมันเป็นโครงสร้างที่เป็นกราฟ เราจึงจัดการมันได้อย่างอิสระ แม้กระทั่งการรองรับข้อมูลที่มากกว่า 2 มิติขึ้นไป เช่น ข้อมูลในลักษณะ multi-dimensional arrays

การคิวรี DOM เราอาจใช้ ID แบบดั้งเดิมของ HTML เช่นการใช้ฟังก์ชัน getElementById() ใน DOM Level 2 ซึ่งจะทำการวิ่งค้นหาข้อมูลเป็นรายตัว หรือการใช้ XPath ใน DOM Level 3 ทำให้สามารถเข้าถึงข้อมูลได้รวดเร็วขึ้นโดยการระบุตำแหน่งตามลำดับชั้น ส่วน TULIP มีสิ่งที่คล้ายกับเป็น array indexed ทำให้เราสามารถเข้าถึงข้อมูลใด ๆ ได้ในแบบ pointer หรือแบบ directed

การ aggregation หรือการยุบรวมข้อมูล รวมทั้งการ summarize ข้อมูล ถ้าเป็น DOM จะใช้วิธี top-down คือข้อมูลรวมกันมาอยู่แล้ว หลังจากนั้นจึงค่อยแยกไปดูชิ้นส่วนย่อยไปตามลำดับชั้น แต่ TULIP ข้อมูลจะเป็นชิ้น ๆ จึงใช้วิธี bottom-up นำข้อมูลจากชิ้นส่วนย่อยมารวมกันให้ใหญ่ขึ้น

ประสิทธิภาพของ DOM ขึ้นอยู่กับ XML engine หรือ JavaScript engine ที่อยู่ในเครื่องที่ใช้แสดงผล ในขณะที่ TULIP อยู่บน RDF กราฟ เราสามารถใช้ภาษา SPARQL หรือใช้คุณสมบัติของส่วนประกอบเสริมต่าง ๆ ของ SPARQL engine ในการคิวรีหรือการจัดการกับข้อมูลแบบ online ผ่านเซิร์ฟเวอร์ SPARQL endpoint หรือแบบ offline โดยใช้ JavaScript ไลบรารี

ตารางที่ 7 เป็นการสรุปเปรียบเทียบข้อแตกต่างระหว่าง DOM และ TULIP RDF Schema

ตารางที่ 7 สรุปข้อแตกต่างระหว่าง DOM และ TULIP

	DOM	TULIP
Storage	Hierarchical cascading objects	Array indexed individual objects
Data Structure	Tree	Graph
Manageability	Document based	Element based
Representation	Static (row-based only)	Dynamic (column หรือ row-orient)
Query	Name with Index (ID, XPath)	Multi-dimensional array index
Aggregation	Top-down	Bottom-up
Performance	ขึ้นอยู่กับ XML/JS engine และ library	ขึ้นอยู่กับ SPARQL engine และ extension

ภาพรวมคุณสมบัติของ TULIP RDF Schema

TULIP สามารถนำไปประยุกต์ใช้ได้หลากหลาย เนื่องจากมันถูกออกแบบมาให้ยืดหยุ่น ผู้นำไปใช้สามารถเลือกใช้ Schema ได้หลายโมเดลขึ้นอยู่กับการใช้งาน รองรับประเภทของข้อมูลได้หลายแบบ ซึ่งแท้จริงแล้ว TULIP สามารถรองรับส่วนประกอบหลายประเภทของ Document Object Model (DOM) โดยสามารถย้ายข้อมูล paragraph หรือ text block มา Unified อยู่ในโครงสร้างรูปแบบเดียวกันของ TULIP (ถึงแม้ว่า RDF Schema ที่ออกแบบจะรองรับข้อมูลหลายประเภท แต่ในขอบเขตงานวิจัยนี้ เบื้องต้นตัวไลบรารีต้นแบบอิมพลีเมนต์เพียงแค่การแปลงตารางและรายการ) และถูกนำไปใช้ในการแปลงข้อมูลบทความของวิกิพีเดียให้อยู่ในรูปแบบ TULIP เพื่อเป็นลิงก์โอเพ่นเดต้าเซ็ตระดับ 5 ดาวอีกชุดหนึ่ง ทำให้แอปพลิเคชันซีแมนติกเว็บสามารถบริโภคข้อมูลไปใช้ได้อย่างสะดวกมากขึ้น ทั้งหมดนี้เพื่อที่จะทำให้เกิดโครงสร้างข้อมูลที่ไม่ใช่เพียงแค่ machine-readable แต่จะใกล้เคียง machine-understandable มากยิ่งขึ้น

บทที่ 5

การทดลองและผลการวิจัย

บทนี้จะกล่าวถึงการนำเดต้าโมเดล TULIP มาอิมพลีเมนต์เป็นไลบรารีต้นแบบ ปัญหาที่พบในการทดลองและวิธีแก้ไข รวมถึงวิธีการนำไลบรารีที่พัฒนาขึ้น มาใช้สำหรับระบบตัวอย่างเพื่อทดลองนำไปใช้งาน

แนวคิดและวิธีวิจัย

เดต้าโมเดลและ RDF Schema vocabulary ที่พัฒนาขึ้นในงานวิจัยมีชื่อว่า TULIP – Table, List, Paragraph and Unstructured Information vocabulary (ตัวย่อสลับตำแหน่งแบบเดียวกับ OWL Web Ontology Language) มี namespace prefix คือ http://purl.org/tulip/ns# และ specification อยู่ที่ http://purl.org/tulip/spec

ผู้วิจัยได้ออกแบบและสร้างไลบรารีต้นแบบเพื่อเป็นไลบรารีอ้างอิง (reference libraries) ไว้ 2 ตัวคือ ไลบรารีส่วนที่ใช้สร้าง TULIP เดต้าเซ็ต พัฒนาด้วยภาษา Python ส่วนไลบรารีที่นำ TULIP เดต้าเซ็ตไปใช้ เป็น JavaScript ไลบรารี และได้วางซอร์สโค้ดไว้ที่ GitHub repository สารารถเข้าถึงได้ที่ https://github.com/julthep/tulip และ https://github.com/julthep/tulip.js ตามลำดับ

โดยระบบตัวอย่างที่ทำการทดลองนำ TULIP vocabulary และไลบรารีนี้ไปใช้งานในการสกัดข้อมูลจากวิกิพีเดียคือ TLPedia (TULIP from Wikipedia) และได้ทำการแปลงบทความจำนวนหนึ่งให้อยู่ในรูปแบบ TULIP โดยเดต้าเซ็ตที่ได้จากการทดลองสามารถเข้าถึงได้ผ่าน SPARQL endpoint ที่ http://www.tlpedia.org/sparql/ หรือสามารถดาวน์โหลดเดต้าเซ็ตในรูปแบบไฟล์ได้ที่ http://www.tlpedia.org/datasets/ โดยจะทำการอัพเดทเป็นระยะ และเพิ่มปริมาณบทความให้มากขึ้นตามช่วงเวลา

ภาษาที่ใช้ในการพัฒนาระบบต้นแบบ

ผู้วิจัยได้เลือกใช้ภาษาโปรแกรมดังนี้เพื่อใช้ในการพัฒนาระบบต้นแบบ

- Python ใช้เป็นภาษามาตรฐานหลักที่ใช้พัฒนาระบบทดสอบฝั่งเซิร์ฟเวอร์ เนื่องจากมีไลบรารีสนับสนุนจำนวนมาก โดยเฉพาะอย่างยิ่งในเรื่องการวิเคราะห์และประมวลผลข้อมูลในรูปแบบต่าง ๆ โดยผู้วิจัยได้เลือก Python 3 เนื่องจากเป็นเวอร์ชันล่าสุด ถึงแม้ว่าจะมีไลบรารีบางตัวที่น่าสนใจแต่รองรับเพียงแค่ Python 2 และยังไม่ได้ถูกพอร์ทขึ้นมาที่ Python 3 ผู้วิจัยจึงตัดสินใจที่จะไม่เลือกใช้ไลบรารีเหล่านั้น เพื่อให้ระบบต้นแบบสามารถพัฒนาต่อยอดได้อย่างสะดวก เนื่องจาก Python 2 เวอร์ชันสุดท้ายคือ Python 2.7.17 ได้หมดอายุลงในวันที่ 1 มกราคม 2020[1]
- JavaScript ใช้เป็นภาษามาตรฐานหลักที่ใช้พัฒนาระบบทดสอบฝั่งไคลเอนท์ เนื่องจากมีไลบรารีสนับสนุนจำนวนมาก โดยเฉพาะอย่างยิ่งในเรื่องการนำเสนอข้อมูลที่หลากหลาย

เครื่องมือเสริมและไลบรารีเพิ่มเติม

ผู้วิจัยได้เลือกใช้เครื่องมือดังนี้เพิ่มเติมเพื่อเสริมการทำงานของระบบต้นแบบ โดยยึดหลักการของ DRY หรือ Don't Repeat Yourself เพื่อที่จะไม่ต้องทำงานซ้ำซ้อนกับไลบรารีเดิมที่มีให้เลือกใช้อยู่แล้วในทั้งสองภาษาที่ผู้วิจัยใช้พัฒนา

- Beautiful Soup[2] หรือ BS4 เป็น Python ไลบรารีที่ใช้เป็นเครื่องมือในการสกัดข้อมูลจากเว็บ สาเหตุที่เลือกใช้เครื่องมือนี้ในขณะที่ยังมีเครื่องมือตัวอื่นให้เลือกใช้ เช่น Scrapy[3] เนื่องจาก BS4 เป็นเครื่องมือที่แพร่หลายมากที่สุด มีชุมชนผู้ใช้ขนาดใหญ่ที่สามารถช่วยในการค้นคว้าในกรณีที่ต้องการข้อมูลเพิ่มเติมในการพัฒนาระบบทดสอบ ส่วน Scrapy เป็นเครื่องมือสกัดข้อมูลจากเว็บอีกตัวหนึ่งที่มีฐานผู้ใช้จำนวนมากเช่นกัน ทำให้มีข้อมูลสนับสนุนทางเทคนิคมากมาย และมีคำสั่งสำเร็จรูปที่สะดวกทำให้ใช้งานง่าย แต่ด้วยความที่เป็นเครื่องมือใช้งานง่ายนี้เองจึงทำให้ Scrapy มีความยืดหยุ่นน้อยกว่า
- RDFLib[4] เป็นไลบรารี Python ในการอ่านและเขียนข้อมูลรูปแบบ RDF ในฟอร์แมทต่าง ๆ

---

[1] Sunsetting Python 2. Available from: https://www.python.org/doc/sunset-python-2/

[2] Beautiful Soup. Cummy. Available from: https://www.crummy.com/software/BeautifulSoup/

[3] Scrapy – A Fast and Powerful Scraping and Web Crawling Framework. Available from: https://scrapy.org/

[4] The RDFlib community. Available from: https://rdflib.github.io/

- Transcrypt[5] และ RapydScript[6] ทั้งคู่เป็น source code translator หรือ source to source compiler ที่ใช้ในการแปลงซอร์สโค้ด Python เป็น JavaScript จากการทดลองผู้วิจัยพบว่า Transcrypt มีประสิทธิภาพและความครอบคลุมครบถ้วนในการแปลงมากกว่า RapydScript
- Chart.js[7] เป็นไลบรารีสำหรับสร้างชาร์ต สาเหตุที่เลือกใช้ไลบรารีตัวนี้ในขณะที่ยังมีตัวอื่นให้เลือกใช้ เช่น D3[8], Highcharts[9] หรือ Google Charts[10] เป็นต้น เนื่องจาก Chart.js เป็นไลบรารีโอเพ่นซอร์สที่เป็นที่นิยม มีคนใช้แพร่หลาย มีกลุ่มสังคมออนไลน์ขนาดใหญ่ที่สามารถเข้าไปสืบค้นข้อมูลได้ในกรณีที่ติดปัญหาในการนำไปใช้งาน ส่วน D3 ซึ่งเป็นไลบรารีโอเพ่นซอร์สเช่นกัน มีความนิยมและมีกลุ่มผู้ใช้กว้างขวางมากกว่า แต่ก็เป็นไลบรารีสร้างไดอะแกรมได้อย่างหลากหลายที่มีความยืดหยุ่นและซับซ้อน ในขณะที่ความต้องการชาร์ตไลบรารีสำหรับงานวิจัยนี้เป็นเพียงแค่ต้องการไลบรารีสำเร็จรูปเพื่อนำข้อมูลในรูปแบบ TULIP ไปแสดงผลในรูปแบบชาร์ตเพื่อการสาธิตเท่านั้น ส่วน Highcharts ถึงแม้ว่าจะเป็นไลบรารีสร้างชาร์ตสำเร็จรูปที่ได้รับความนิยม แต่ก็ไม่ใช่ไลบรารีโอเพ่นซอร์สอย่างแท้จริง (เป็นเพียงแค่เปิดเผยซอร์สและแจกจ่ายให้ใช้ฟรีหากไม่นำไปใช้ในเชิงการค้า) ในขณะที่ Google Charts เป็นเว็บเซอร์วิสที่เรียกใช้งานผ่าน API จึงไม่สะดวกในการใช้งานสำหรับระบบสาธิต นอกจากนี้ผู้วิจัยยังได้ทดลองใช้ jqPlot[11] และ Plotly.js[12] อีกด้วย แต่ทั้งคู่ใช้งานไม่สะดวกเท่า Chart.js เนื่องจาก jqPlot จำเป็นต้องใช้ร่วมกับ jQuery ส่วน Plotly.js จำเป็นต้องใช้ร่วมกับ D3 และถึงแม้จะเป็นโอเพ่นซอร์สไลบรารีแต่ตัวมันเองก็เป็นส่วนหนึ่งของ Plotly ซึ่งเป็นผลิตภัณฑ์ทางการค้า
- CherryPy[13] เป็นเว็บเฟรมเวิร์ค ผู้วิจัยเลือกตัวนี้แทนที่จะเป็น Django[14] หรือ Flask[15] ที่ได้รับความนิยมมากกว่า เพราะความกะทัดรัดและคล่องตัว รวมทั้งมีคุณสมบัติเป็นเว็บ

[5] Transcrypt – Python in the browser – Lean, fast, open! Available from: https://www.transcrypt.org/

[6] RapydScript. Available from: http://atsepkov.github.io/RapydScript/

[7] Chart.js – Open source HTML5 Charts for your website. Available from: https://www.chartjs.org/

[8] D3.js – Data-Driven Documents. Available from: https://d3js.org/

[9] Interactive JavaScript charts for your webpage. Highcharts. Available from: https://www.highcharts.com/

[10] Google Charts. Google Developers. Available from: https://developers.google.com/chart

[11] jqPlot Charts and Graphs for jQuery. Available from: http://www.jqplot.com/

[12] Plotly JavaScript Graphing Library. Plotly. Available from: https://plotly.com/javascript/

[13] CherryPy — A Minimalist Python Web Framework. Available from: https://cherrypy.org/

[14] The Web framework for perfectionists with deadlines. Django. Available from: https://www.djangoproject.com/

[15] Flask Documentation (1.1.x). Available from: https://flask.palletsprojects.com/

เซิร์ฟเวอร์ในตัว ซึ่งมีความเสถียรและปลอดภัย[16]กว่าเว็บเซิร์ฟเวอร์ตัวที่เป็นไลบรารีมาตรฐานที่ติดมากับ Python[17] หรือแม้แต่ Flask[18] ในขณะที่ Django ถึงแม้จะเป็นเฟรมเวิร์คขนาดใหญ่ที่มีทุกอย่างครบถ้วน แต่ก็ออกแบบมาเพื่อให้ใช้งานร่วมกันกับเว็บเซิร์ฟเวอร์โดยเฉพาะเท่านั้น[19] นอกจากนี้ยังได้ทดลองตัวเลือกอื่นคือ Bottle[20] อีกด้วย

- RDF4J[21] (เดิมชื่อว่า OpenRDF Sesame) เป็น RDF เฟรมเวิร์คสำหรับ Java แต่ตัวมันเองมีคุณสมบัติเป็น triplestore ด้วย ซึ่งผู้วิจัยได้ใช้คุณสมบัตินี้ในการทดลองสร้างเดต้าเซ็ตและคิวรีด้วย SPARQL เพื่อทดสอบเดต้าโมเดล TULIP

ข้อมูลที่นำมาทดลอง

1. ส่วนประกอบของวิกิพีเดียที่นำมาใช้ในระบบตัวอย่าง TLPedia

ระบบตัวอย่าง TLPedia นำข้อมูลทุกส่วนจากแต่ละบทความของวิกิพีเดีย ยกเว้นเฉพาะส่วน categories ซึ่ง DBpedia ได้ทำการแปลงข้อมูลส่วนนี้ให้อยู่ในรูปแบบ RDF ครบถ้วนอยู่แล้ว (ซึ่งแตกต่างจาก Infobox ซึ่ง DBpedia แปลงข้อมูลเป็น RDF เพียงบางส่วนเฉพาะที่สอดคล้องกับสกีมาที่ออกแบบไว้ ไม่ได้เก็บข้อมูลทั้งหมดของ Infobox ในรูปแบบตารางเหมือนกับ TULIP)

2. ทำไมถึงใช้วิกิพีเดียแทนที่จะใช้ข้อมูลในเว็บ

เมื่อเปรียบเทียบกับวิกิพีเดียที่ปัจจุบันนี้มีจำนวนบทความเฉพาะภาษาอังกฤษประมาณ 6.2 ล้านบทความ[22] การสกัดข้อมูลจากเว็บทั้งโลกที่มีปริมาณมากมายมหาศาล[103] ซึ่งปัจจุบันนี้มีประมาณ 54 พันล้านเพจ[23] (นับเฉพาะ surface web หรือหน้าเว็บที่เปิดให้เข้าถึงโดยสาธารณะและสามารถทำ index โดย search engines เอาไว้ เราไม่สามารถนับรวม deep web หรือเว็บที่ต้องใช้รหัสผ่านหรือหน้าเว็บที่ไม่ได้ถูก search engines ทำ index ไว้ได้) เป็นงานที่มีความท้าทายหลายอย่าง[104-106] ดังนั้นผู้วิจัยจึงเลือกที่จะใช้ข้อมูลจากวิกิพีเดีย ด้วยเหตุผลดังต่อไปนี้

[16] Why CherryPy. Bitbucket. Available from: https://bitbucket.org/cherrypy/cherrypy/wiki/WhyCherryPy

[17] HTTP servers — Python 3.8.2 documentation. Available from: https://docs.python.org/3/library/http.server.html

[18] Deployment Options — Flask Documentation (1.1.x). Available from: https://flask.palletsprojects.com/en/1.1.x/deploying/

[19] Deploying Django. Django. Available from: https://docs.djangoproject.com/en/3.0/howto/deployment/

[20] Bottle – Python Web Framework — Bottle 0.13-dev documentation. Available from: http://bottlepy.org/

[21] Eclipse RDF4J. The Eclipse Foundation. Available from: https://rdf4j.org/

[22] Statistics. Wikipedia. Available from: https://en.wikipedia.org/wiki/Special:Statistics

[23] The size of the World Wide Web (The Internet). WorldWideWebSize. Available from: https://worldwidewebsize.com/

ในไตรมาสที่สองและสามของปี ค.ศ. 2019 วิกิพีเดียเป็นเว็บไซต์ที่มีผู้เข้าชมมากเป็นอันดับ 5 ของโลก[24] การที่มีผู้เข้าเยี่ยมชมเป็นจำนวนมากย่อมแสดงถึงคุณภาพและความน่าสนใจของเนื้อหาบนวิกิพีเดีย ในขณะที่ข้อมูลในเว็บยังมีความไม่เที่ยงตรงอยู่มาก เนื้อหาในวิกิพีเดียถูกตรวจสอบและคัดกรองอยู่ตลอดเวลา (ในอัตรามากกว่า 1.9 ครั้งต่อวินาที[25]) โดยชุมชนที่มีความรู้ความชำนาญในเรื่องนั้น ๆ[26] และถึงแม้ว่าจะไม่เที่ยงตรง 100 เปอร์เซ็นต์ แต่ก็ยังมีคุณภาพมากกว่าเมื่อเปรียบเทียบกับข้อมูลบนเว็บทั้งหมด หรือแม้แต่สารานุกรมแบบปิดที่ไม่ให้ผู้ใช้เข้าไปปรับปรุงแก้ไขความถูกต้องได้[27]

เนื้อหาบนเว็บเต็มไปด้วยความเปลี่ยนแปลง ทั้งเกิดขึ้นใหม่และบางส่วนถูกลบหายไปอยู่ตลอดเวลา ในขณะที่บนวิกิพีเดียมีความเสถียร เนื้อหาส่วนใหญ่จะไม่ถูกลบออกมีแต่จะพอกพูนรายละเอียดเพิ่มมากขึ้นทุกวัน ด้วยอัตราเฉลี่ย 600 บทความต่อวัน[25]

เนื้อหาในแต่ละเว็บไม่สอดคล้องสัมพันธ์กันและมีความซ้ำซ้อนกันเป็นจำนวนมาก นอกจากนั้นยังมีข้อมูลขยะที่ใช้งานไม่ได้อยู่อีกมากมาย มีเพียงแค่ส่วนน้อยนิดในเว็บเท่านั้นที่ตรงกับความต้องการของผู้ใช้ในแต่ละคน จนมีการกล่าวกันว่า “99% ของข้อมูลในเว็บ ไร้ประโยชน์กับ 99% ของผู้ใช้เว็บ”[105] ในขณะที่วิกิพีเดียพยายามจัดหมวดหมู่ และสร้างสภาวะความเป็นหนึ่งเดียว ลดความซ้ำซ้อน และมีการอ้างอิงที่มาของข้อมูลอย่างชัดเจน

รูปแบบและสไตล์ในการนำเสนอข้อมูลบนเว็บมีความหลากหลาย ในขณะที่วิกิพีเดียพยายามที่จะควบคุมรูปแบบให้ค่อนข้างตายตัวและมีความสอดคล้องกันในแต่ละบทความ อีกทั้งยังกำหนดรูปแบบในการเชื่อมโยงถึงกันในแนวทางที่ชัดเจน เพื่อให้ดูเหมือนเป็นสารานุกรม “เล่ม” เดียวกันอยู่บนระบบเดียว

ตารางที่ 8 เป็นสรุปคีย์เวิร์ดเปรียบเทียบความแตกต่างระหว่างเวิร์ลไวด์เว็บกับวิกิพีเดีย ซึ่งข้อได้เปรียบทั้งหมดนี้มีเป้าหมายเพื่อให้วิกิพีเดียเป็นเสมือนสารานุกรมที่มีความเที่ยงตรงใช้อ้างอิงได้อย่างแท้จริง

---

[24] wikipedia.org Competitive Analysis. Alexa. Available from: https://web.archive.org/web/20190817082636/https://www.alexa.com/siteinfo/wikipedia.org

[25] Wikipedia Statistics. Wikipedia. Available from: https://en.wikipedia.org/wiki/Wikipedia:Statistics

[26] Reliability of Wikipedia. Wikipedia. Available from: https://en.wikipedia.org/wiki/Reliability_of_Wikipedia

[27] How Accurate Is Wikipedia? Live Science. Available from: https://www.livescience.com/32950-how-accurate-is-wikipedia.html

ตารางที่ 8 สรุปข้อเปรียบเทียบเวิร์ลไวด์เว็บกับวิกิพีเดีย

Web	Wikipedia
Much larger	Smaller
Inaccuracy	Accuracy
Very dynamic	Less dynamic
Diversity	Unity
Irrelevant	Relevant
Redundancy	Consistency

3. ทำไมถึงใช้เพียงแค่วิกิพีเดียภาษาอังกฤษ

งานวิจัยนี้กำหนดขอบเขตไว้เพียงแค่วิกิพีเดียภาษาอังกฤษ แทนที่จะใช้ภาษาอื่น ๆ ด้วย เช่น ภาษาไทย เนื่องจากวิกิพีเดียภาษาอังกฤษเป็นวิกิพีเดียที่เกิดขึ้นก่อนวิกิพีเดียภาษาอื่น และภาษาอังกฤษก็ยังเป็นภาษาหลักในการสร้างบทความขึ้นครั้งแรกก่อนที่จะมีบทความที่มีเนื้อหาเดียวกันนั้นในภาษาอื่น แม้ว่าจะมีหลายบทความที่มีความเป็นท้องถิ่น เช่น บทความที่เกี่ยวข้องกับวัฒนธรรมของแต่ละชนชาติหรือภาษา ซึ่งหมายความว่าจะมีบทความที่เกี่ยวกับสิ่งนั้นในภาษาท้องถิ่นถูกสร้างขึ้นมาก่อน อย่างไรก็ตามโดยส่วนใหญ่แล้วบทความเหล่านั้นก็จะถูกแปลเป็นภาษาอังกฤษเพิ่มเติมขึ้นมาในภายหลัง

อีกทั้งบทความในภาษาอังกฤษโดยรวมแล้วยังมีทั้งคุณภาพและปริมาณมากกว่าบทความในภาษาอื่น ทั้งในแง่ของจำนวนและขนาดของบทความหรือความละเอียดของเนื้อหา จากการที่มีจำนวนผู้อ่านมากที่สุดเนื่องจากเป็นภาษากลาง ในทางเดียวกันทำให้มีผู้เขียนจำนวนมากที่สุดในการเข้ามาสร้าง ตรวจสอบ และแก้ไข ทำให้สามารถรักษาคุณภาพและความถูกต้องของเนื้อหาให้ได้อยู่ในระดับสูงตลอดเวลา และในเชิงปริมาณและคุณภาพนี้เองที่มีผลต่องานวิจัยนี้โดยตรง เนื่องจากจะพบว่าบทความส่วนใหญ่ที่มีตารางและรายการจะพบได้ในภาษาอังกฤษ ในขณะที่บทความเดียวกันนั้นในภาษาอื่นยังไม่มีความครบถ้วน โดยหลายบทความถูกสร้างไว้เป็นเพียงแค่ stub[28] หรือโครงเท่านั้น

ในกรณีของวิกิพีเดียภาษาไทย หากจะเปรียบเทียบกันแล้ว นอกจากจะมีปริมาณน้อยกว่าภาษาอังกฤษเกินกว่า 40 เท่า (ประมาณ 140,000 ต่อ 6,200,000)[29] ยังมีคุณภาพด้อยกว่ามาก เนื่องจากผู้ใช้ยังมีจำนวนน้อยทำให้มีผู้เข้ามาตรวจสอบแก้ไขได้ไม่เพียงพอต่อจำนวนข้อผิดพลาด อีก

[28] Stub. Wikipedia. Available from: https://en.wikipedia.org/wiki/Wikipedia:Stub

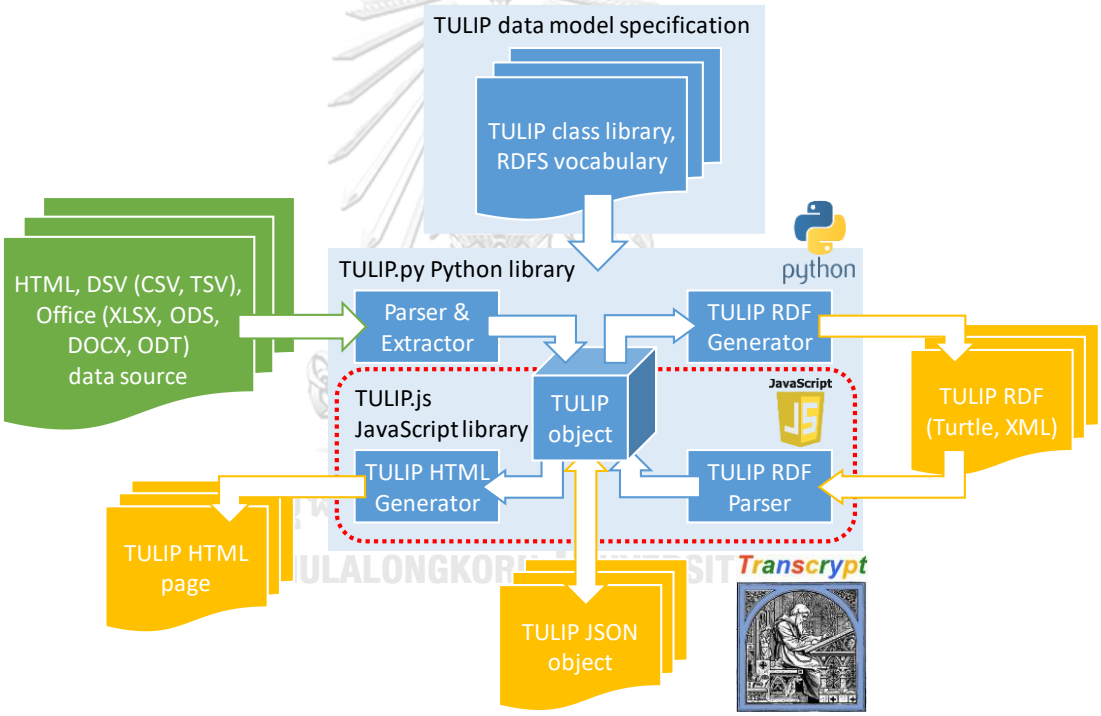
[29] List of Wikipedias. Wikipedia. Available from: https://en.wikipedia.org/wiki/List_of_Wikipedias#Detailed_list

ทั้งยังไม่สามารถจัดการกับการก่อกวน (vandalism[30]) ได้ทันท่วงที ทำให้ผลของการก่อกวนนั้นยังคงค้างอยู่ในบทความต่าง ๆ เป็นจำนวนมาก

ส่วนข้อจำกัดในแง่ของตัวอักษรในภาษาอื่นจะไม่พบในงานวิจัยนี้ เนื่องจากตามมาตรฐานของซีแมนติกเว็บเทคโนโลยีสแต็ก ทั้ง URI และ RDF ซึ่งเป็นหน่วยย่อยหลักของงานวิจัยนี้สามารถรองรับ Unicode ได้จึงทำให้สามารถเก็บข้อมูลได้ทุกภาษาแม้ว่าบทความในวิกิพีเดียภาษาอังกฤษนั้นจะมีตัวอักษรภาษาอื่นปะปนอยู่

ไลบรารีต้นแบบ TULIP.py และ TULIP.js

1. สถาปัตยกรรมภายในของไลบรารีต้นแบบ



ภาพที่ 54 สถาปัตยกรรมของไลบรารีต้นแบบ TULIP.py และ TULIP.js

ภาพที่ 54 แสดงสถาปัตยกรรมของไลบรารีต้นแบบ TULIP ซึ่งมีส่วนประกอบต่าง ๆ ภายในหลายส่วน ส่วนแรกคือเดต้าโมเดลเป็น RDF Schema ที่ออกแบบไว้ตามที่ได้อธิบายไว้ในบทที่แล้ว โดยกำหนดเป็น specification เอาไว้เพื่อใช้ในการแปลงข้อมูล ซึ่งในระบบตัวอย่างจะใช้ HTML เป็นหลัก แต่ในทางปฏิบัติสามารถปรับปรุงเพื่อแปลงจากตารางหรือรายการประเภทอื่นได้ทุกรูปแบบ ผู้วิจัยได้พัฒนา Python ไลบรารีขึ้นมา โดยมีโมดูล HTML Parser และ Extractor เพื่อคัดกรอง

[30] Vandalism. Wikipedia. Available from: https://en.wikipedia.org/wiki/Wikipedia:Vandalism

ข้อมูลแล้วจึงทำการสกัดข้อมูล ในขอบเขตของระบบต้นแบบนี้จะสกัดมาเพียงแค่ตารางและรายการมาก่อน เพื่อนำลงไปใน TULIP อ็อบเจ็กต์ซึ่งเป็น Python คลาสเนื่องจาก TULIP เดต้าโมเดลจะมี 2 ระดับ ระดับ RDF Schema เอาไว้สำหรับเก็บ RDF ทริปเปิลแบบ 5 ดาว แต่เมื่อนำเข้ามาใช้ในโปรแกรมก็จะเก็บเป็น TULIP อ็อบเจ็กต์ที่อยู่ในรูปแบบ TULIP คลาสที่สะดวกต่อการเข้าถึงในเชิงโปรแกรมมิ่ง ทำนองเดียวกันกับ HTML เมื่ออยู่ภายนอกเป็น markup language format แต่เมื่อนำเข้ามาในโปรแกรม เช่น เว็บเบราเซอร์ ก็จะอยู่ในรูปแบบ DOM นั่นเอง

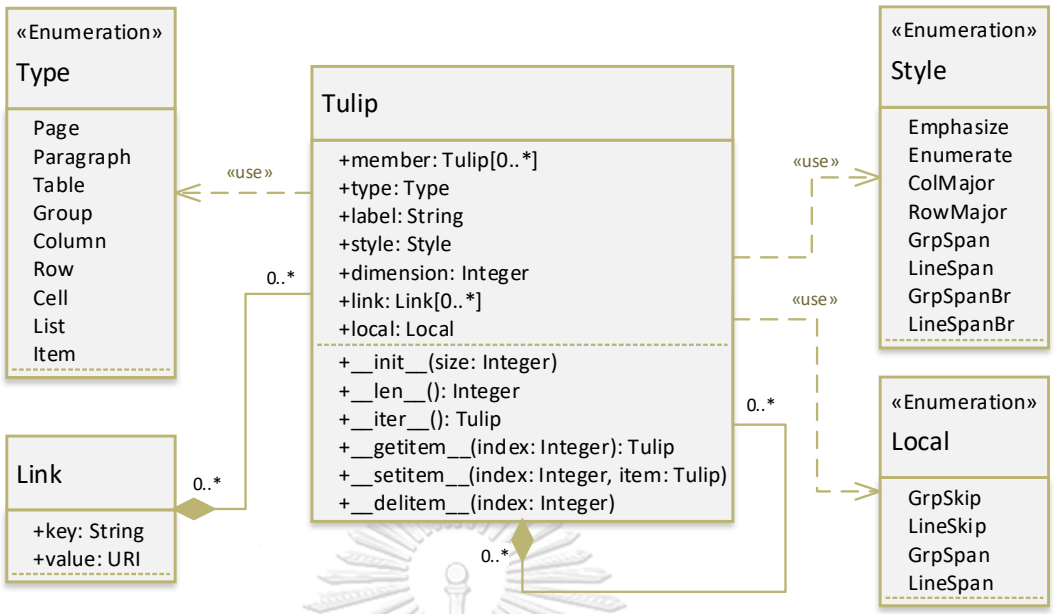
ซึ่ง TULIP อ็อบเจ็กต์นั้นสามารถนำเอาไปแปลงต่อโดยใช้โมดูล TULIP RDF Generator ไปสร้างเป็น RDF ไฟล์เก็บไว้เป็น Turtle format หรือรูปแบบ RDF serialization มาตรฐานอื่น ๆ ในขณะเดียวกัน เราอาจจะมีความจำเป็นต้อง parse ข้อมูลในรูปแบบ RDF serialization มาตรฐานกลับมา เพื่อเอาไปแสดงผลในรูปแบบอื่น ก็จะมีโมดูล TULIP RDF Parser คืออ่าน TULIP ในไฟล์ RDF รูปแบบต่าง ๆ แล้วนำมาแปลงเป็น TULIP อ็อบเจ็กต์ในรูปแบบ TULIP คลาสคือสามารถใช้ทั้ง 2 โมดูลนี้ในการแปลงไปแปลงกลับระหว่าง TULIP คลาสและ TULIP RDF serialization ได้ นอกจากนี้ก็ยังมีโมดูล HTML Generator ที่ใช้สร้าง HTML เพจกลับออกมาได้อีกด้วย ซึ่งโดยหลักการแล้วหน้าตาของไฟล์ HTML ที่ได้จะเหมือนต้นฉบับในเชิงซีแมนติก เนื่องจากเราเก็บแค่เฉพาะบางส่วนของ HTML เพจต้นฉบับ เช่น ตารางและรายการ ในเชิงความหมาย ข้อมูลจำพวกข้อความ, ตัวเลข และโครงสร้างตารางและรายการจะถูกเก็บไว้อย่างครบถ้วน แต่ในเบื้องต้นเรายังไม่ได้เก็บพวกรูปแบบหรือสไตล์ต่าง ๆ เช่น สี หรือลักษณะรูปแบบอักขระ รูปแบบเหล่านี้จะยังไม่ถูกเก็บไว้เนื่องจากยังไม่มีผลเชิงความหมายในเบื้องต้นอย่างมีนัยสำคัญ แต่ต่อไปในอนาคตผู้วิจัยสามารถปรับปรุงระบบต้นแบบให้เก็บรายละเอียดเหล่านี้เพิ่มเติมได้ ทำให้สามารถเก็บได้เสมือนกับว่า เราสามารถเลียนแบบหรือสร้าง HTML เพจปลายทางให้มีหน้าตาย้อนกลับมาได้เหมือนต้นฉบับเลย ทั้งสีทั้งรูปแบบตัวอักษรหรือแม้กระทั่งการเก็บ CSS หรือ Cascading Style Sheet ลงไปใน TULIP เลยก็มีความเป็นไปได้ แต่เบื้องต้นในขอบเขตของงานวิจัยนี้ TULIP เดต้าโมเดลจะรองรับแค่ข้อมูลในเชิงความหมายดังกล่าวก่อน นอกจากนี้ TULIP อ็อบเจ็กต์ในรูปแบบ TULIP คลาสยังสามารถบันทึกเป็น JSON เก็บไว้ได้ เพื่อนำมาโหลดกลับเข้าไปเป็น TULIP อ็อบเจ็กต์ในเครื่องได้โดยตรงและเอาไปใช้แปลงต่อในรูปแบบอื่นได้ทันทีแบบไม่ต้อง parse ก่อน

โมดูลทั้งหมดพัฒนาด้วยภาษา Python เพื่อเป็นไลบรารีสำหรับใช้ในการอ่านหรือบันทึกข้อมูลในรูปแบบ TULIP ซึ่งสามารถอิมพอร์ตใช้งานได้เลยโดยโหลดจาก PyPI repository เพื่อใช้ในการพัฒนาแอปพลิเคชัน นอกจากนี้ยังมี TULIP ไลบรารีในเวอร์ชัน JavaScript อีกด้วยเพื่อนำข้อมูล

ในรูปแบบ TULIP ไปแสดงผลในเว็บเบราเซอร์ ซึ่งโดยทั่วไปจะต้องพัฒนาเพิ่มต่างหากโดยการเขียนไลบรารีเพิ่มโดยภาษา JavaScript แต่เนื่องจากผู้วิจัยเห็นว่า แท้จริงแล้ว JavaScript ที่ต้องถูกพัฒนาขึ้นก็จำเป็นจะต้องมี 3 โมดูลนี้เช่นเดียวกัน คือ TULIP คลาส, TULIP RDF Parser และ TULIP HTML Generator โดยที่ 3 โมดูลนี้ ผู้วิจัยได้พัฒนาขึ้นมาแล้วโดยภาษา Python ดังนั้นจึงเห็นว่าไม่สมควรที่จะต้องเขียนขึ้นมาใหม่ โดยผู้วิจัยได้ใช้เทคนิคทางโปรแกรมมิ่งอย่างหนึ่งที่เรียกว่าเป็น source code translator หรือ source to source compiler เพื่อใช้ในการแปลงไลบรารีต้นฉบับที่พัฒนาด้วยภาษา Python มาเป็นภาษา JavaScript โดยเครื่องมือที่ผู้วิจัยใช้คือ Transcrypt ซึ่งถึงแม้ว่าจะแปลงได้ไม่สมบูรณ์ 100% เนื่องจากลักษณะของทั้งสองภาษานี้มีความแตกต่างกันในรายละเอียดอยู่พอสมควร เช่น engine ในการรันโปรแกรมก็มีความยืดหยุ่นได้ไม่เท่ากัน JavaScript engine ที่อยู่บนเว็บเบราเซอร์ทำงานอยู่ใน sandbox ก็จะมีข้อจำกัด ไม่สามารถเปิดสิทธิ์ในการรันโปรแกรมได้เท่า Python interpreter ที่รันบนโฮสต์โดยตรง อย่างไรก็ตาม ผู้วิจัยใช้ Transcrypt แปลง syntax ของ Python ได้เป็นส่วนใหญ่ ซึ่งส่วนของโปรแกรมที่ใช้ไลบรารีนอกเหนือจากไลบรารีมาตรฐานจะไม่สามารถแปลงได้เลย ผู้วิจัยจะต้องเขียนขึ้นมาใหม่ทั้งหมด แต่เนื่องจากไลบรารีของระบบ TULIP ส่วนใหญ่ผู้วิจัยพัฒนาขึ้นมาเองเกือบทั้งหมดโดยแทบไม่ได้ใช้ไลบรารีสำเร็จรูปเพิ่มเติม จึงมีส่วนที่ต้องแก้ไขหรือสร้างเพิ่มเติมเพียงบางส่วน ดังนั้น TULIP.js ที่พัฒนามาจาก TULIP.py จึงมีความแตกต่างกันในระดับหนึ่ง แต่ผลลัพธ์ที่ได้จากการทำงานของไลบรารีทั้งสองภาษาจะออกมาเหมือนกัน

2. <u>TULIP Object ในรูปแบบ Python Class และ JavaScript Class</u>

โครงสร้างของ TULIP คลาสแสดงในรูปแบบ UML Class Diagram เป็นดังภาพที่ 55 ซึ่งจะมีคลาสหลักอยู่เพียงแค่คลาสเดียวคือคลาสที่มีชื่อว่า Tulip มีรายละเอียดดังต่อไปนี้



ภาพที่ 55 แผนภูมิคลาสของ TULIP ไลบรารี

2.1 TULIP Class Attributes

- member จะเป็นลิสต์ที่ประกอบด้วยสมาชิกที่เป็นอ็อบเจ็กต์ของคลาส Tulip จำนวนตั้งแต่ 0 ตัวขึ้นไป
- type จะบ่งบอกถึงชนิดของ Tulip โหนดนี้ว่าเป็นโหนดประเภทใด ซึ่งจะมีค่าได้ดังต่อไปนี้
    - Page หมายถึงโหนดนี้เป็นหน้าเอกสาร เช่น หน้าเว็บเพจ หรือบทความวิกิ
    - Paragraph หมายถึงโหนดนี้เป็นย่อหน้าข้อความ
    - Table หมายถึงโหนดนี้เป็นตาราง
    - Group หมายถึงโหนดนี้เป็นกลุ่มของเซลล์ เช่น สดมภ์หรือแถวของตาราง ขึ้นอยู่กับว่าเป็น column-major หรือ row-major
    - Column หมายถึงโหนดนี้เป็นสดมภ์ทั้งสดมภ์ โดยทั่วไปจะใช้ Group เป็นหลัก นอกจากกรณีที่ข้อมูลต้นทางระบุมาอย่างเฉพาะเจาะจง
    - Row หมายถึงโหนดนี้เป็นแถวทั้งแถว โดยทั่วไปจะใช้ Group เป็นหลัก นอกจากกรณีที่ข้อมูลต้นทางระบุมาอย่างเฉพาะเจาะจง

- Cell หมายถึงโหนดนี้เป็นเซลล์ของตารางเพียงหนึ่งเซลล์
- List หมายถึงโหนดนี้เป็นรายการ
- Item หมายถึงโหนดนี้เป็นไอเท็มหนึ่งของรายการ

- label คือข้อมูลประจำโหนดนี้ อยู่ในรูปแบบของตัวอักษร Unicode
- style คือลักษณะรูปแบบของโหนดนี้ เก็บเป็นชนิดข้อมูลดิกชันนารีหรือตารางแฮชที่เก็บ flag ที่มีค่าได้เป็น True, None หรือ (explicit) False ซึ่งหมายความว่าโหนดแต่ละโหนดสามารถมีลักษณะได้ตั้งแต่ 0 ลักษณะขึ้นไป ซึ่งหากกำหนดค่าเป็น True จะมีลักษณะได้ดังนี้
  - Emphasize หมายความว่าโหนดนี้มีลักษณะเด่นกว่าปกติ เช่น เป็นหัวตาราง
  - Enumerate หมายความว่าโหนดนี้มีลักษณะที่แสดงเป็นตัวเลขกำกับลำดับได้ เช่น เป็นรายการแบบมีเลขลำดับ
  - ColMajor หมายความว่าโหนดนี้เป็นส่วนหนึ่งของตารางที่เก็บบันทึกแบบ column-major คือเริ่มแบ่ง group ด้วยสดมภ์ก่อนแล้วจึงค่อยลงไปในแต่ละ line ตามแถวของสดมภ์ (ตามที่ได้อธิบายไว้ในบทที่แล้วดังภาพที่ 27)
  - RowMajor หมายความว่าโหนดนี้เป็นส่วนหนึ่งของตารางที่เก็บบันทึกแบบ row-major คือเริ่มแบ่ง group ด้วยแถวก่อนแล้วจึงค่อยกวาด line ในแต่ละสดมภ์ทีละเซลล์ (ตามที่ได้อธิบายไว้ในบทที่แล้วดังภาพที่ 28) ซึ่งทั้ง ColMajor และ RowMajor หากไม่ได้กำหนดมาหรือกำหนดเป็น True ทั้งคู่ จะขึ้นอยู่การโปรแกรมที่นำข้อมูลไปใช้ ว่าได้มีการตั้งค่าปริยายไว้ในลักษณะไหนเป็นรูปแบบหลัก
  - GrpSpan หมายความว่าโหนดนี้เป็นส่วนหนึ่งของตารางที่มีการเชื่อม (หรือเรียกว่า spanned cell) มากกว่า 1 เซลล์ขึ้นไปในแนว group เดียวกัน ดังข้อมูลในตารางตัวอย่าง column-major นี้ (→ หมายถึง GrpSpan)

ราคาซื้อ →	ราคาซื้อ →	ราคาขาย →	ราคาขาย →
ก่อน VAT	หลัง VAT	ก่อน VAT	หลัง VAT

จะแสดงผลเป็นดังนี้

ราคาซื้อ		ราคาขาย	
ก่อน VAT	หลัง VAT	ก่อน VAT	หลัง VAT

- LineSpan หมายความว่าโหนดนี้เป็นส่วนหนึ่งของตารางที่มีการเชื่อม (หรือเรียกว่า spanned cell) มากกว่า 1 เซลล์ขึ้นไปในแนว line เดียวกัน
- GrpSpanBr เป็นตัวกำหนดจุดสิ้นสุดของแต่ละ spanned cell ใช้ในกรณีที่มี spanned cell มากกว่า 1 spanned cell ที่มีข้อมูลเหมือนกันและอยู่ติดกันในแนว group เดียวกัน ดังข้อมูลในตารางตัวอย่าง column-major นี้ (→ หมายถึง GrpSpan และ Br หมายถึง GrpSpanBr)

ราคาขาย →	ราคาขาย→Br	ราคาขาย →	ราคาขาย→Br
ก่อน S.charge	หลัง S.charge	ก่อน VAT	หลัง VAT

จะแสดงผลเป็นดังนี้

ราคาขาย		ราคาขาย	
ก่อน S.charge	หลัง S,charge	ก่อน VAT	หลัง VAT

ซึ่งหากข้อมูลไม่มีการกำหนด GrpSpanBr มา จะถูกแสดงผลเป็นดังนี้

ราคาขาย			
ก่อน S.charge	หลัง S,charge	ก่อน VAT	หลัง VAT

- LineSpanBr เป็นตัวกำหนดจุดสิ้นสุดของแต่ละ spanned cell ใช้ในกรณีที่มี spanned cell มากกว่า 1 spanned cell ที่มีข้อมูลเหมือนกันและอยู่ติดกันในแนว line เดียวกัน ซึ่งทั้ง GrpSpanBr และ LineSpanBr ไม่มีความจำเป็นต้องใช้หากข้อมูลในเซลล์ถัดไปมีข้อมูลที่แตกต่างกันถึงแม้ว่าจะยังมีลักษณะเป็น spanned cell ก็ตาม ดังตัวอย่างที่ได้แสดงไว้ใน GrpSpan

- dimension คือตัวเลขที่บ่งบอกถึงความลึกของโหนดนั้นว่ามีสมาชิกลูกหลานลงไปกี่ชั้น

- link เป็นข้อมูลเกี่ยวกับลิงก์ของโหนด เก็บอยู่ในรูปแบบของดิกชันนารีหรือตารางแฮชที่มี key เป็นตัวอักษร Unicode และ value เป็นข้อมูลในรูปแบบ URI การเก็บลิงก์ด้วย key-value ทำให้ attribute link สามารถเก็บได้มากกว่าหนึ่งลิงก์

- local เป็นที่เก็บข้อมูลชั่วคราวที่ใช้ในขณะที่ไลบรารีทำงานในเครื่อง เก็บในรูปแบบดักชันนารีที่มี key ดังนี้ GrpSkip, LineSkip, GrpSpan และ LineSpan (ใช้ชั่วคราวในช่วงจังหวะรันไทม์)

2.2 TULIP Class Methods

TULIP class methods ทั้งหมดจะเป็น magic method ซึ่งเป็น special method ในรูปแบบหนึ่งที่ทำให้เราสามารถเรียกใช้ methods เหล่านี้โดยวิธีพื้นฐานของภาษาโปรแกรมมิ่ง จะแสดงตัวอย่างการเรียกใช้ magic method ในแต่ละ method ที่สร้างไว้ใน TULIP Class Library

- __init__(size) สำหรับสร้าง TULIP object โดยระบุขนาดเริ่มต้นหรือจำนวนโหนดสมาชิกในชั้นแรกเพื่อ ตัวอย่างเช่น

```
tlp_obj1 = Tulip()
```

เป็นการสร้าง TULIP object ชื่อว่า tlp_obj1 จำนวน 1 ตัว โดยที่ไม่มีสมาชิกลูกอยู่เลย

```
tlp_obj1 = Tulip(10)
```

เป็นการสร้าง TULIP object ชื่อว่า tlp_obj1 จำนวน 1 ตัว ที่มีสมาชิกลูกชั้นแรก 10 ตัว

- __setitem__(index, item) ใช้ในการกำหนดค่าให้กับโหนดสมาชิกแบบระบุตำแหน่งโดยใช้เครื่องหมาย subscription หรือ [ ] ตัวอย่างเช่น

```
tlp_obj1[5] = Tulip(20)
```

เป็นการสร้าง TULIP object ที่มีสมาชิกในชั้นที่ 2 จำนวน 20 ตัว ภายใต้ tlp_obj1[5] หรือที่ตำแหน่งที่ 6 ของ tlp_obj1

```
tlp_obj1[5][15] = Tulip(7)
```

เป็นการสร้าง TULIP object ที่มีสมาชิกในชั้นที่ 3 จำนวน 7 ตัว ภายใต้ tlp_obj1[5][15]

- __getitem__(index) สำหรับการเข้าถึงสมาชิกลูกของ TULIP object แบบระบุตำแหน่งโดยใช้เครื่องหมาย subscription หรือ [ ] โดยจะตอบกลับออกมาเป็น TULIP object ตัวอย่างเช่น

```
tlp_obj2 = tlp_obj1[3]
```

tlp_obj2 จะเป็นตัวเดียวกันกับโหนดลูกสมาชิกตัวที่ 4 ของ tlp_obj1

- __delitem__(index) สำหรับลบสมาชิกลูกแบบระบุตำแหน่งโดยใช้เครื่องหมาย subscription หรือ [ ] ตัวอย่างเช่น

```
del tlp_obj1[5]
```

จะลบโหนดสมาชิกตัวที่ 6 ออกจาก tlp_obj1

- __iter__() สำหรับการเข้าถึงโหนดสมาชิกลูกโดยใช้ iteration โดยจะตอบกลับออกมาเป็น TULIP object ตัวอย่างเช่น

```
for mem_obj in tlp_obj1:
    print(mem_obj.label)
```

จะวนลูปแสดงข้อมูลของโหนดสมาชิกตั้งแต่ตัวแรกจนถึงตัวสุดท้าย

- __len__() สำหรับสอบถามขนาดจำนวนสมาชิกลูกชั้นแรกของ TULIP object นั้น โดยจะตอบกลับค่าออกมาเป็นจำนวนเต็ม ตัวอย่างเช่น

```
var1 = len(tlp_obj1)
```

ต่อเนื่องจากตัวอย่างข้างบน var1 จะมีค่าเป็น 10

3. ฟังก์ชันที่มีใน TULIP library

ฟังก์ชันหลักที่มีให้ใช้ใน TULIP ไลบรารีมีดังนี้ (ไม่ได้แสดงฟังก์ชันย่อยที่ใช้เฉพาะภายในตัวไลบรารีเอง)

- read_file(filename: String): String
- write_file(text: String, filename: String)
- read_url(URL: String): String
- parse_html(HTML: String): Tulip
- parse_article(Wikipedia_article_name: String): Tulip
- parse_rdf(RDF: String, RDF_serialized_format: String): Tulip
- gen_turtle(TULIP: Tulip): String

```
• gen_html(TULIP: Tulip): String
• dump_tulip(TULIP: Tulip): String
• ttl2nt(Turtle_filename: String, NTriples_filename: String)
• tulip2json(TULIP: Tulip): String
• json2tulip(JSON: String): Tulip
```

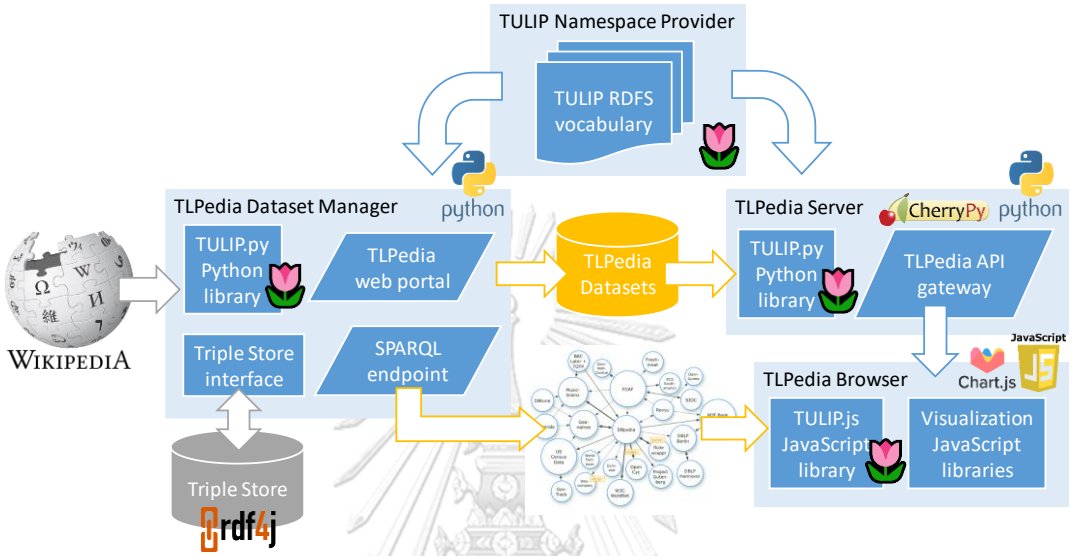
- read_file(filename) อ่านไฟล์อักขระ ส่งกลับมาเป็นสายอักขระ
- write_file(string, filename) นำสายอักขระ มาเขียนลงไฟล์อักขระ
- read_url(URL) อ่านหน้าเว็บเพจ ส่งกลับมาเป็นสายอักขระ
- parse_html(HTML) พาร์สสายอักขระ HTML ส่งกลับมาเป็น TULIP อ็อบเจ็กต์
- parse_article(Wikipedia_article_name) พาร์สบทความวิกิพีเดีย ส่งกลับมาเป็น TULIP อ็อบเจ็กต์
- parse_rdf(RDF, RDF_serialized_format) พาร์สสายอักขระ RDF ระบุเลือกรูปแบบ RDF ส่งกลับมาเป็น TULIP อ็อบเจ็กต์
- gen_turtle(TULIP) อ่าน TULIP อ็อบเจ็กต์ ส่งค่ากลับมาเป็นสายอักขระ Turtle
- gen_html(TULIP) อ่าน TULIP อ็อบเจ็กต์ ส่งค่ากลับมาเป็นสายอักขระ HTML
- dump_tulip(TULIP) อ่าน TULIP อ็อบเจ็กต์ ส่งค่ากลับมาเป็นสายอักขระ TULIP ในรูปแบบต้นไม้โครงสร้างแบบเยื้อง indented tree structure
- ttl2nt(str:Turtle_filename, str:NTriples_filename) แปลงไฟล์อักขระ Turtle เป็นไฟล์อักขระ N-Triples
- tulip2json(TULIP) อ่าน TULIP อ็อบเจ็กต์ ส่งกลับมาเป็นสายอักขระ JSON
- json2tulip(JSON) อ่านสายอักขระ JSON ส่งกลับมาเป็น TULIP อ็อบเจ็กต์

ระบบตัวอย่าง TLPedia

TLPedia คือการนำเอา TULIP RDF data model และไลบรารีมาประยุกต์ใช้ เป็นเสมือนตัว reference prototype หรือตัวแอปพลิเคชันต้นแบบอ้างอิง โดยเอา TULIP มาทดลองแปลง

บทความวิกิพีเดียให้อยู่ในรูปแบบ 5-star open data จึงเป็นที่มาของชื่อ TLPedia โดยผลลัพธ์ที่ได้คือ ชุดข้อมูลตัวอย่าง TLPedia เดต้าเซ็ต, แอปพลิเคชันตัวอย่าง TLPedia server เพื่อให้บริการ API และ TLPedia browser เพื่อให้บริการเข้าถึงข้อมูลด้วยเว็บแอปพลิเคชัน

1. โครงสร้างส่วนประกอบของระบบตัวอย่าง TLPedia



ภาพที่ 56 การนำ TULIP ไลบรารีมาประยุกต์ใช้ในการทำระบบตัวอย่าง TLPedia

ภาพที่ 56 เป็นส่วนประกอบต่าง ๆ ของระบบตัวอย่าง TLPedia ซึ่งประกอบไปด้วย 3 ส่วนหลักคือ ส่วนที่ใช้สร้างและบริหารจัดการเดต้าเซ็ต หรือ TLPedia Dataset Manager, ส่วนที่ให้บริการเดต้าเซ็ต หรือ TLPedia Server และส่วนสุดท้ายคือส่วนที่นำเดต้าเซ็ตไปใช้งาน TLPedia Browser ซึ่งมีรายละเอียดดังนี้

1.1 ระบบสร้างดาต้าเซ็ต

TLPedia Dataset Manager จะใช้ในการสร้างและบริหารจัดการ TLPedia dataset พัฒนาด้วยภาษา Python โดยมีส่วนประกอบภายในคือ TULIP.py โดยพัฒนาออกมาให้เป็นไลบรารีอยู่บน PyPI repository[31] (ซอร์สโค้ดอยู่บน Github repository[32]) โดยสามารถ install โดย pip และสามารถ import เข้าไปในโปรแกรมที่เขียนโดย Python ใด ๆ เพื่อให้สามารถทำการแปลงข้อมูลที่อยู่ในรูปแบบ HTML ให้อยู่ในรูปแบบ TULIP RDF serialization และการแปลงย้อนกลับ รวมทั้งการถ่ายโอน TULIP อ็อบเจ็กต์จากหน่วยความจำไปสู่ JSON และอ่านย้อนกลับ

[31] TULIP-RDF Python library. Julthep Nandakwang. Available from: https://pypi.org/project/tulip-rdf/

[32] TULIP-RDF Python library source code. Julthep Nandakwang. Available from: https://github.com/julthep/tulip/

ระบบ TLPedia จะใช้ TULIP.py ในการดึงเอาข้อมูลจาก Wikipedia มาลง triplestore ที่ผู้วิจัยเลือกใช้ ในกรณีนี้คือ RDF4J หรือชื่อเดิมคือ Sesame มี web portal ในการจัดการเดต้าเซ็ต มี SPARQL endpoint เพื่อให้บริการเดต้าเซ็ตในรูปแบบลิงก์เดต้าระดับ 5 ดาว ในขณะเดียวกันก็มีฐานข้อมูล TLPedia เดต้าเซ็ตในรูปของไฟล์ให้ดาวน์โหลดด้วย

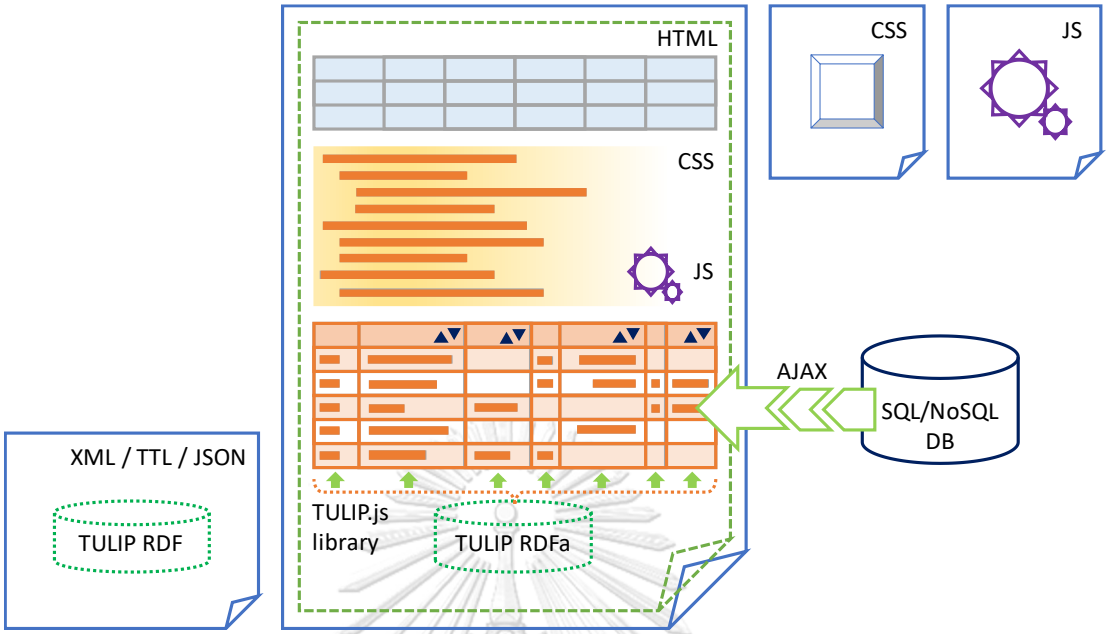
1.2 ระบบที่ให้บริการเดต้าเซ็ต

TLPedia Server คือการนำเอาข้อมูลจากเดต้าเซ็ตไปให้บริการในรูปแบบ Web API พัฒนาด้วย Python โดยใช้ CherryPy เป็น Web framework ตัว API Server นี้ใช้เป็นตัวบริการ API ให้กับเว็บแอปพลิเคชันต่าง ๆ เข้ามาใช้เป็นอีกทางเลือกหนึ่งนอกจากเข้าถึง TULIP เดต้าเซ็ตผ่าน SPARQL endpoint แบบลิงก์เดต้าระดับ 5 ดาว ซึ่ง API Server ก็จะใช้ TULIP.py ไลบรารีเช่นเดียวกันในการแปลง TULIP เดต้าเซ็ตกลับมาเป็น HTML หรือส่งออกมาเป็นชุด RDF ทริปเปิล

1.3 ส่วนที่นำเดต้าเซ็ตไปใช้งาน

TLPedia Browser เป็นเว็บแอปพลิเคชันตัวอย่างที่ใช้ในการเข้าถึงข้อมูล TLPedia เดต้าเซ็ตผ่าน TLPedia API แล้วนำมาแสดงผลบนเว็บเบราเซอร์ อย่างที่กล่าวไปก่อนหน้านี้ ตัว TULIP.js ที่ใช้เป็นไลบรารีในการเข้าถึงข้อมูล TULIP มาจากการใช้ Transcrypt แปลง TULIP.py มาอีกต่อหนึ่งเพื่อให้สามารถทำงานได้บน JavaScript engine บนเว็บเบราเซอร์ เนื่องจากเป็นเว็บแอปพลิเคชันผู้วิจัยก็สามารถใส่ฟีเจอร์ต่าง ๆ เช่น การนำ chartJS ไลบรารีมาเสริมเพื่อเอาข้อมูล TULIP มาแสดงผลเป็นชาร์ต เป็นต้น ส่วนประกอบทั้งหมดนี้ทำงานอยู่บนชุด TULIP RDF Schema vocabulary ชุดเดียวกันตามที่กำหนด specification ในบทที่แล้ว

2. ซีแมนติกเว็บเพจโดยการนำตารางและรายการแบบ 5 ดาวเข้าไปฝังในเว็บเพจ



ภาพที่ 57 การใช้ TULIP.js library ในการนำเข้าข้อมูลในรูปแบบ TULIP เพื่อแสดงผลในเว็บเพจ

โดยทั่วไปเว็บเพจปกติจะมีส่วนประกอบคือ HTML ที่ใช้เก็บเนื้อความตัวอักษรรวมไปถึงลิงก์ไปที่รีซอร์สไฟล์ที่เกี่ยวข้อง เช่น รูปภาพ เสียง หรือวิดิโอ ฯลฯ และจะมี CSS ที่ใช้กำหนดรูปแบบการแสดงผล รวมไปถึง JavaScript ที่ทำให้เกิดผลในเรื่องของกลไกต่าง ๆ ในเชิงลอจิกหรือโปรแกรมมิ่ง และข้อมูลก็อาจจะถูกทำการดึงด้วย AJAX มาจากแอปพลิเคชันเซิร์ฟเวอร์ผ่านทาง API ที่เชื่อมโยงข้อมูลมาจาก SQL หรือ NoSQL เซิร์ฟเวอร์อีกที

แต่ถ้าหากมี TULIP เราก็สามารถใส่ข้อมูล TULIP RDF ลงไปในเว็บเพจได้ตรง ๆ โดยเราใช้ TULIP JavaScript ไลบรารีอ่านข้อมูลที่เป็น TULIP ลงไปอัพเดทเนื้อความในเว็บเพจได้เลยแบบสด ๆ หรือ on the fly ซึ่งข้อมูลก็สามารถนำมาได้จากหลายวิธี เช่น เราสามารถผังข้อมูล RDF ลงไปในไฟล์ HTML ได้เลยแบบ RDFa หรืออาจนำเข้ามาจาก RDF TripleStore ไม่ว่าจะเป็นข้อมูล RDF ในรูปแบบใดก็ตาม เช่น XML, TTL, JSON ดังนั้นข้อมูลของเราก็จะเป็น 5 ดาวสมบูรณ์แบบ เป็นข้อมูลที่เป็นลิงก์เดต้าอย่างแท้จริงที่ถูกนำเอามาแสดงผล

ปัญหาและวิธีการแก้ไข

1. ปัญหาเกี่ยวกับระบบต้นแบบ

1.1 ชื่อรีซอร์สมาจากชื่อไฟล์ HTML, อิลิเมนต์ <title> หรือจากอิลิเมนต์ <h1>

มีหลายแนวทางในการกำหนดชื่อรีซอร์ส เช่น นำมาจากชื่อของ HTML ไฟล์, จาก <title> อิลิเมนต์ หรือจาก <h1> อิลิเมนต์ ซึ่งจะมีข้อแตกต่างกันในหลายกรณี เช่น สำหรับบทความวิกิพีเดีย สามส่วนนี้จะใกล้เคียงกันมาก ความแตกต่างมีเพียงแค่ หากชื่อบทความที่ปรากฏอยู่ใน <h1> อิลิเมนต์มีช่องว่าง ชื่อไฟล์ HTML ก็จะถูกแทนด้วยเครื่องหมาย _ (ขีดเส้นใต้) และหากชื่อบทความมีอักขระพิเศษหรือตัวอักษรที่ไม่ใช่ภาษาอังกฤษก็จะถูกแทนที่ด้วยการทำ percent-encoding ตามมาตรฐาน RFC3986[107] ส่วนชื่อบทความใน <title> อิลิเมนต์จะเหมือนกับใน <h1> อิลิเมนต์ยกเว้นจะปิดท้ายด้วยข้อความ “ - Wikipedia”

สำหรับบทความในเว็บเพจทั่วไป จะมีความไม่แน่นอนในการตั้งชื่อไฟล์ HTML เป็นอย่างมาก และอิลิเมนต์ <title> ก็อาจจะประกอบด้วยข้อความส่วนเกินหลายอย่าง เช่น ชื่อองค์กรและคีย์เวิร์ดสำหรับการทำ search engine optimization หรือ SEO[33] ดังนั้นถึงแม้ว่าการใช้อิลิเมนต์ <h1> จะไม่มีข้อกำหนดตายตัวทำให้มีโอกาสที่จะมีจำนวนมากกว่า 1 อิลิเมนต์หรือไม่มีเลย แต่ก็มีแนวปฏิบัติที่นิยมใช้กันว่าควรจะมี <h1> เพียงแค่ 1 อิลิเมนต์ในแต่ละเว็บเพจ ไม่มากและไม่น้อยไปกว่านี้[34]

ดังนั้นแนวทางที่ผู้วิจัยเลือกใช้สำหรับงานวิจัยนี้จึงเลือกใช้อิลิเมนต์ <h1> ตัวแรกสุดที่พบในไฟล์ HTML ในการกำหนดชื่อรีซอร์สซึ่งจะตรงกันกับแนวทางที่ DBpedia เลือกใช้สำหรับบทความจากวิกิพีเดีย ทำให้สามารถแม็ปรีซอร์สของทั้งสองเดต้าเซ็ตเข้าหากันได้โดยง่าย

1.2 การดึงข้อมูลชื่อรายการ

นอกจากแท็ก <lh> ที่ปรากฏในฉบับร่างของ HTML3[35] ซึ่งถูกตัดทิ้งไปใน HTML3.2 ซึ่งเป็นฉบับใช้งานจริง ก็ยังไม่มีการกำหนดมาตรฐานในการระบุชื่อของรายการ[36]เหมือนกับการกำหนดชื่อตารางโดยแท็ก <caption> แม้ว่าจะมีการนำเสนอให้กำหนดแท็กมาตรฐานขึ้นมาในลักษณะ

---

[33] What is the H1 Tag and Why it is Important for SEO. Available from: https://www.reliablesoft.net/h1-tag/

[34] <h1>–<h6>: The HTML Section Heading elements. Available from: https://developer.mozilla.org/en-US/docs/Web/HTML/Element/Heading_Elements

[35] List Header. W3C. Available from: https://www.w3.org/MarkUp/html3/listheader.html

[36] Digital Publishing and Accessibility in W3C Documents. W3C. Available from: https://www.w3.org/TR/dpub-accessibility/#semantic-list-head

เดียวกัน[37] แต่วิธีที่ใช้กันโดยทั่วไปในการกำหนดชื่อของรายการคือใช้ header tags ในการแสดงหัวเรื่องของรายการ[38] (เช่น <h2> tag ถึง <h6> tag) ในทำนองเดียวกัน วิกิพีเดียก็ได้ออกข้อกำหนดให้ใช้ section header ในการตั้งชื่อให้กับรายการ[39] ซึ่งในที่สุดแล้วก็จะถูกแปลงให้เป็น header tags ในรูปแบบ HTML page เมื่อแสดงผลให้กับผู้ใช้ ดังนั้นผู้วิจัยจึงแก้ปัญหาโดยการดึง header tags ตัวที่อยู่ก่อนหน้ารายการนำมาเป็นชื่อของรายการ ซึ่งจะยอมรับในกรณีที่อาจจะมีโอกาสที่รายการ 2 รายการหรือมากกว่า มีชื่อรายการซ้ำกันเนื่องจากภายใต้ header tag เดียวกันมีรายการมากกว่า 1 รายการ

ส่วนในกรณีของรายการย่อยที่ซ้อนอยู่ใน nested list จะไม่มีปัญหานี้เนื่องจากรายการในรายการหลักจะกลายเป็นชื่อของรายการย่อยโดยอัตโนมัติตามโครงสร้างของโมเดล TULIP ที่ถูกออกแบบไว้

1.3 การดึงข้อมูลชื่อตาราง

ข้อมูลตาราง HTML ส่วนใหญ่จะมีกำหนดชื่อตารางอยู่ในรูปแบบของ <caption> แท็กมาตามปกติ แต่จำนวนไม่น้อยที่ชื่อตารางไปอยู่ในส่วนที่เป็น header หรือ footer ของตารางโดยอยู่ในเซลล์ข้อมูลในรูปแบบของ <td colspan> แท็ก หากเป็นแบบนั้นก็จะไม่สามารถแยกได้อย่างชัดเจนว่าข้อมูลในเซลล์นั้นเป็นชื่อของตารางหรือเป็นส่วนหนึ่งของข้อมูลตาราง ดังนั้นหากไม่พบ <caption> แท็กผู้วิจัยจะแก้ปัญหาโดยใช้วิธีเดียวกันกับการหาชื่อรายการ คือการนำเอา header tags ที่อยู่ก่อนหน้าตารางมาเป็นชื่อของตาราง

1.4 การแยกชนิดของข้อมูลตัวอักษรและตัวเลข

หากข้อมูลต้นทางมีทั้งตัวอักษรและตัวเลขให้กำหนดเป็นชนิดตัวอักษรภาษาอังกฤษ หากข้อมูลต้นทางมีแต่เฉพาะตัวเลขรวมทั้งเครื่องหมาย . และเครื่องหมาย , ขั้นแยกทุก 3 ตัวหรือนำหน้าด้วยเครื่องหมาย - โดยไม่มีช่องว่างอยู่ที่ตำแหน่งใด ๆ ไม่ว่าจะชิดขวาหรือชิดซ้าย ให้กำหนดข้อมูลเป็นชนิดตัวเลขแบบจำนวนเต็มหรือทศนิยมขึ้นอยู่กับข้อมูลต้นทาง (โดยตัดเครื่องหมาย , ออกไป)

2. ปัญหาที่มาจากแนวทางปฏิบัติในการใช้งานของวิกิพีเดียเอง

มีปัญหาหลายอย่างในการแปลงข้อมูลจากรายการและตารางของวิกิพีเดีย ที่มีต้นตอมาจากแนวทางปฏิบัติซึ่งเป็นคำแนะนำปกติทั่วไปในการใช้งานของวิกิพีเดียเอง รวมทั้งแนวทางที่ถูกต้องที่

[37] The lost LH element. Available from: http://brian.moonspot.net/2006/08/09/the-lost-lh-element/

[38] list head/caption/title [proposal]. WICG. Available from: https://discourse.wicg.io/t/proposal-list-head-caption-title/1832

[39] Manual of Style/Lists. Wikipedia. Available from: https://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style/Lists#List_naming

เป็นข้อกำหนดไว้สำหรับการเขียนบทความของวิกิพีเดีย ที่เรียกว่า Manual of Style (MoS)[40] ซึ่งประมวลมาจาก guidelines[41] ที่เกี่ยวข้องในการสร้างส่วนประกอบต่าง ๆ ในบทความของวิกิพีเดีย ตัวอย่างเช่น ข้อกำหนดในการสร้างลิงก์ของวิกิพีเดียคือไม่ควรสร้างลิงก์ที่ไปยังบทความอื่นซ้ำกันหลายแห่งในหน้าบทความเดียวกัน แต่ควรสร้างไว้เพียงแค่ลิงก์เดียวตรงคำแรกที่พบในหน้าบทความนั้น เพื่อไม่ให้เกิดสิ่งที่เรียกว่าลิงก์ซ้ำซ้อน หรือ overlinking[42] ซึ่งในกรณีของงานวิจัยนี้ ผู้วิจัยต้องการให้เกิดลิงก์ซ้ำซ้อนขึ้นในทุกข้อความ(ที่มีความหมายเดียวกัน)ที่ปรากฏในบทความ เช่น หากมีลิงก์ไปที่รีซอร์สอื่นในเซลล์ของตารางหนึ่งแล้ว หากข้อความเหมือนกันที่อยู่ที่เซลล์อื่นของตารางอื่นก็ควรมีลิงก์ไปที่รีซอร์สเดียวกัน เพราะในกรณีที่มีการคิวรีข้อมูล ณ เซลล์อื่นจะได้มีลิงก์ครบถ้วนไม่ตกหล่นขาดหายไป ดังตัวอย่างตามภาพที่ 58 เป็นข้อมูลส่วนหนึ่งจากบทความชื่อ “Annie Award for Best Animated Feature”

1990s [edit]

Year	Film	Studios
1992	***Beauty and the Beast***	**Walt Disney Pictures / Walt Disney Feature Animation**
	Bébé's Kids	Hyperion / Paramount Pictures
	FernGully: The Last Rainforest	20th Century Fox Animation / FAI Films / Kroyer Films
1993	***Aladdin***	**Walt Disney Pictures / Walt Disney Feature Animation**
	Little Nemo: Adventures in Slumberland	Tokyo Movie Shinsha
	Once Upon a Forest	20th Century Fox Animation / Hanna-Barbera Productions

ภาพที่ 58 ตัวอย่างข้อมูลตารางจากวิกิพีเดียที่ไม่มีลิงก์ซ้ำซ้อน

จะพบว่ามี 3 ลิงก์คือ Walt Disney Pictures, Walt Disney Feature Animation และ 20th Century Fox Animation ไม่ได้ถูกใส่ให้ครบถ้วนเพื่อไม่ให้เกิดลิงก์ซ้ำซ้อน (มีลิงก์เฉพาะในปี ค.ศ. 1992 แต่ในปี ค.ศ. 1993 ไม่ได้ถูกใส่ลิงก์ไว้) ซึ่งเป็นแนวปฏิบัติที่ถูกต้องของวิกิพีเดีย แต่มีผลกระทบคือ ทำให้ลิงก์ดังกล่าวก็ไม่ถูกสร้างขึ้นที่ระบบ TLPedia ด้วยเช่นกัน ตามภาพที่ 59 ซึ่งหากผู้ใช้งานระบบ TLPedia มีการคัดกรองข้อมูลให้แสดงเฉพาะบางส่วนของตาราง เช่น หากเลือกแสดงผลเฉพาะปี ค.ศ. 1993 ก็จะไม่มีลิงก์ทั้ง 3 ลิงก์ดังกล่าวให้ใช้งานได้ เป็นต้น

[40] Manual of Style. Wikipedia. Available from: https://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style

[41] List of guidelines. Wikipedia. Available from: https://en.wikipedia.org/wiki/Wikipedia:List_of_guidelines

[42] Manual of Style/Overlinking. Available from: https://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style/Linking#Overlinking

1990s		
Year	Film	Studios
1992	Beauty and the Beast	Walt Disney Pictures / Walt Disney Feature Animation
	Bébé's Kids	Hyperion / Paramount Pictures
	FernGully: The Last Rainforest	20th Century Fox Animation / FAI Films / Kroyer Films
1993	Aladdin	Walt Disney Pictures / Walt Disney Feature Animation
	Little Nemo: Adventures in Slumberland	Tokyo Movie Shinsha
	Once Upon a Forest	20th Century Fox Animation / Hanna-Barbera Productions

ภาพที่ 59 ตัวอย่างข้อมูลตารางที่แปลงมาจากวิกิพีเดียที่ไม่มีลิงก์ซ้ำซ้อน

ผู้วิจัยพบว่าหนทางแก้ปัญหาอย่างง่ายวิธีหนึ่งคือ ค้นหาคำแต่ละคำที่มีลิงก์เชื่อมโยง แล้วเพิ่มเติมลิงก์นี้ไปยังคำเหมือนกันคำอื่นที่พบในหน้าเดียวกันทุกคำ ทั้งนี้โดยอนุโลมบนสมมติฐานว่าคำเหมือนกันในบทความเดียวกัน น่าจะมีความหมายเดียวกัน และสามารถมีลิงก์เชื่อมโยงไปที่รีซอร์สปลายทางเดียวกันได้

3. ปัญหาที่มาจากข้อมูลต้นฉบับผิดพลาด

จากที่ได้ทำการทดลองกับข้อมูลบางส่วน พบว่าปัญหาหลายอย่างมาจากความผิดพลาดของข้อมูลต้นฉบับที่วิกิพีเดียเอง เนื่องจากวิกิพีเดียเป็นเว็บที่อนุญาตให้ผู้ใช้สามารถแก้ไขข้อมูลได้เอง ดังนั้นบทความจำนวนหนึ่งในวิกิพีเดียจึงมีโอกาสผิดพลาดได้ตลอดเวลา ทั้งจากที่ถูกต้องอยู่แล้วแต่ถูกแก้ไขจนผิดพลาดไปโดยไม่ตั้งใจและจงใจ หรือเป็นข้อมูลใหม่ที่เพิ่มเข้าไปแต่ไม่สอดคล้องกับข้อมูลเก่าที่มีอยู่แล้ว ทำให้รูปแบบการแสดงผลผิดพลาดไป ซึ่งในหลายกรณีวิกิพีเดียจะมีบ็อต[43](ซอฟท์แวร์ที่ออกแบบให้ทำงานแทนคนโดยอัตโนมัติและทำงานต่อเนื่องอยู่ตลอดเวลา[44])อยู่หลายตัวคอยตรวจสอบความผิดพลาดในหลายประเภท แต่ก็มีหลายกรณีที่บ็อตไม่สามารถตรวจพบความผิดพลาดจนผู้ใช้ข้อมูลมาเห็นแล้วจึงได้ทำการแก้ไขโดยมนุษย์ หรือในบางกรณีที่แม้แต่มนุษย์ก็ยากที่จะมองเห็นความผิดพลาดที่เกิดขึ้นเนื่องจากเป็นจุดที่สังเกตเห็นได้ลำบากหากดูข้อมูลนั้นในรูปแบบปกติ แต่คอมพิวเตอร์อาจจะมองเห็นความแตกต่างของข้อมูลนั้นได้โดยง่ายหากดูในอีกรูปแบบหนึ่ง ดังตัวอย่างตามภาพที่ 60 เป็นข้อมูลตารางทางการเงินจากบทความชื่อ “Microsoft”

[43] Bots. Wikipedia. Available from: https://en.wikipedia.org/wiki/Wikipedia:Bots

[44] Software bot. Wikipedia. Available from: https://en.wikipedia.org/wiki/Software_bot

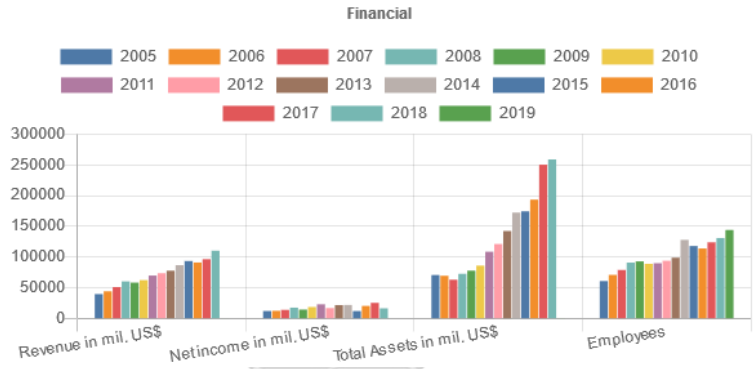
Year	Revenue in mil. US$[142]	Net income in mil. US$[142]	Total Assets in mil. US$[142]	Employees[142]
2005	39,788	12,254	70,815	61,000
2006	44,282	12,599	69,597	71,000
2007	51,122	14,065	63,171	79,000
2008	60,420	17,681	72,793	91,000
2009	58,437	14,569	77,888	93,000
2010	62,484	18,760	86,113	89,000
2011	69,943	23,150	108,704	90,000
2012	73,723	16,978	121,271	94,000
2013	77,849	21,863	142,431	99,000
2014	86,833	22,074	172,384	128,000
2015	93,580	12,193	174,472	118,000
2016	91,154	20,539	193,468	114,000
2017	96,571	25,489	250,312	124,000
2018	110,360	16,571	258,848	131,000
2019	125.843	39.240	286.556	144,106

ภาพที่ 60 ตารางทางการเงินจากบทความวิกิพีเดียชื่อ “Microsoft”

ซึ่งมีตัวเลขที่ผิดพลาดอยู่ซึ่งไม่สามารถสังเกตเห็นได้โดยง่ายด้วยสายตาของมนุษย์ และตอนนี้ยังไม่มีบ็อตของวิกิพีเดียตัวไหนสามารถตรวจสอบพบเจอได้ ซึ่งเมื่อนำข้อมูลมาแปลงและแสดงผลโดยใช้ TLPedia Browser จะได้ตามภาพที่ 61 พร้อมชาร์ต

Financial

Year	Revenue in mil. US$	Net income in mil. US$	Total Assets in mil. US$	Employees
2005	39,788	12,254	70,815	61,000
2006	44,282	12,599	69,597	71,000
2007	51,122	14,065	63,171	79,000
2008	60,420	17,681	72,793	91,000
2009	58,437	14,569	77,888	93,000
2010	62,484	18,760	86,113	89,000
2011	69,943	23,150	108,704	90,000
2012	73,723	16,978	121,271	94,000
2013	77,849	21,863	142,431	99,000
2014	86,833	22,074	172,384	128,000
2015	93,580	12,193	174,472	118,000
2016	91,154	20,539	193,468	114,000
2017	96,571	25,489	250,312	124,000
2018	110,360	16,571	258,848	131,000
2019	125.843	39.240	286.556	144,106

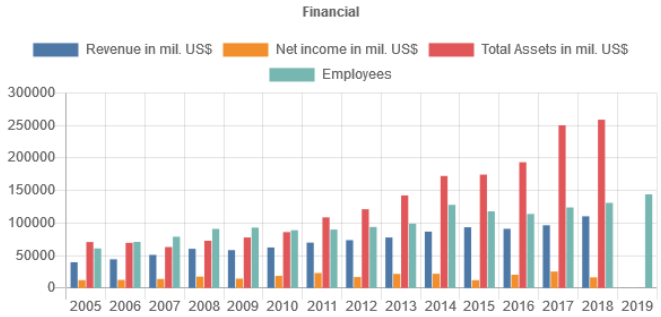


ภาพที่ 61 ตารางทางการเงินจากบทความ “Microsoft” เมื่อแสดงด้วย TLPedia พร้อมชาร์ต

เมื่อสังเกตก็จะพบว่าดูเป็นปกติดีอยู่ แต่เมื่อเรานำมาพลิกมุมมองของตารางใหม่ จาก column-major เป็น row-major ชาร์ตที่แสดงก็จะถูกพลิกมุมมองไปด้วย ดังภาพที่ 62 ก็จะเริ่มสังเกตเห็นว่าบางแท่งของบาร์ชาร์ตหายไป

Financial

Year	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019
Revenue in mil. US$	39,788	44,282	51,122	60,420	58,437	62,484	69,943	73,723	77,849	86,833	93,580	91,154	96,571	110,360	125.843
Net income in mil. US$	12,254	12,599	14,065	17,681	14,569	18,760	23,150	16,978	21,863	22,074	12,193	20,539	25,489	16,571	39.240
Total Assets in mil. US$	70,815	69,597	63,171	72,793	77,888	86,113	108,704	121,271	142,431	172,384	174,472	193,468	250,312	258,848	286.556
Employees	61,000	71,000	79,000	91,000	93,000	89,000	90,000	94,000	99,000	128,000	118,000	114,000	124,000	131,000	144,106

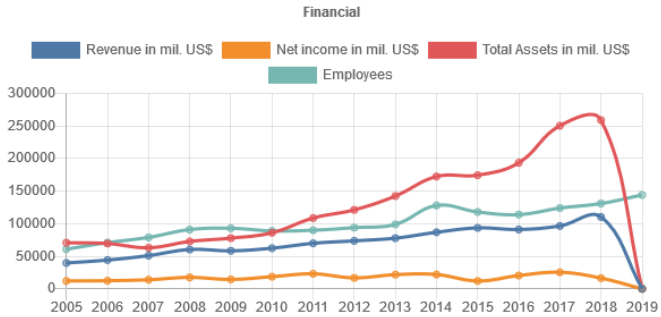


ภาพที่ 62 เมื่อทำการพลิกมุมมองของตารางจาก column-major เป็น row-major

หากเราทำการเปลี่ยนแปลงชนิดของชาร์ต จากบาร์ชาร์ตหรือกราฟแท่ง เป็นไลน์ชาร์ตหรือกราฟเส้น ดังภาพที่ 63 ก็จะเห็นข้อผิดพลาดของข้อมูลได้อย่างชัดเจน ว่าตัวเลขของปี ค.ศ. 2019 ผิดพลาดไป 1,000 เท่า

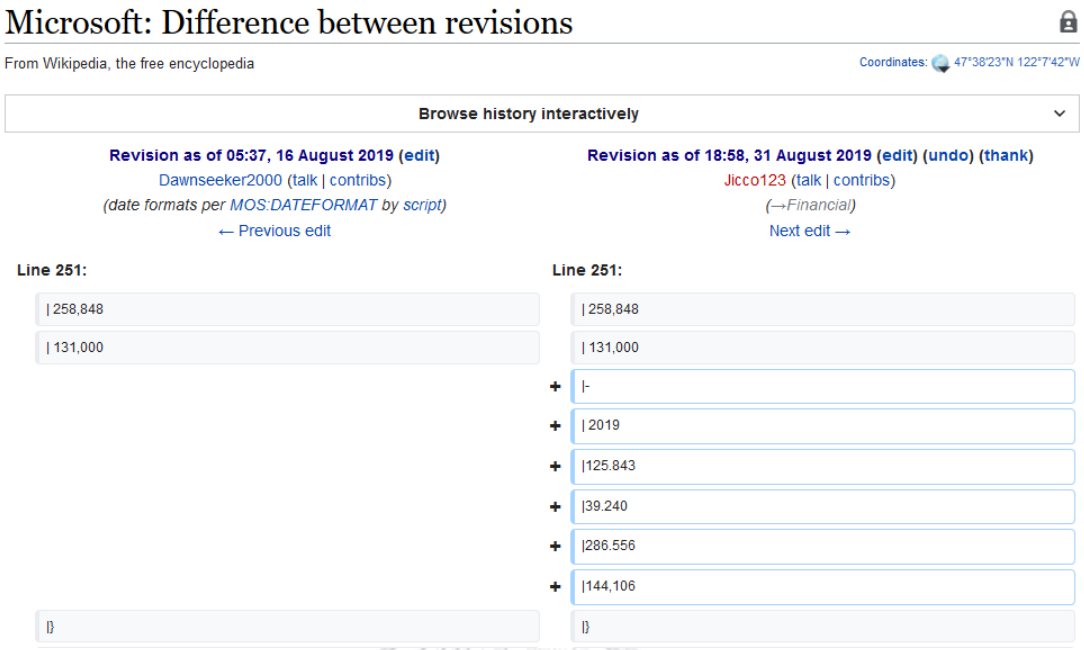
Financial

Year	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019
Revenue in mil. US$	39,788	44,282	51,122	60,420	58,437	62,484	69,943	73,723	77,849	86,833	93,580	91,154	96,571	110,360	125.843
Net income in mil. US$	12,254	12,599	14,065	17,681	14,569	18,760	23,150	16,978	21,863	22,074	12,193	20,539	25,489	16,571	39.240
Total Assets in mil. US$	70,815	69,597	63,171	72,793	77,888	86,113	108,704	121,271	142,431	172,384	174,472	193,468	250,312	258,848	286.556
Employees	61,000	71,000	79,000	91,000	93,000	89,000	90,000	94,000	99,000	128,000	118,000	114,000	124,000	131,000	144,106



ภาพที่ 63 เมื่อทำการเปลี่ยนรูปแบบชาร์ตจากกราฟแท่งเป็นกราฟเส้น

เมื่อย้อนกลับไปดูข้อมูลที่วิกิพีเดียตามภาพที่ 64 พบว่าข้อมูลที่ผิดพลาดนี้ถูกนำเข้าบันทึกไว้ในบทความนี้ตั้งแต่วันที่ 31 สิงหาคม 2019



Microsoft: Difference between revisions

From Wikipedia, the free encyclopedia

Coordinates: 47°38′23″N 122°7′42″W

Browse history interactively

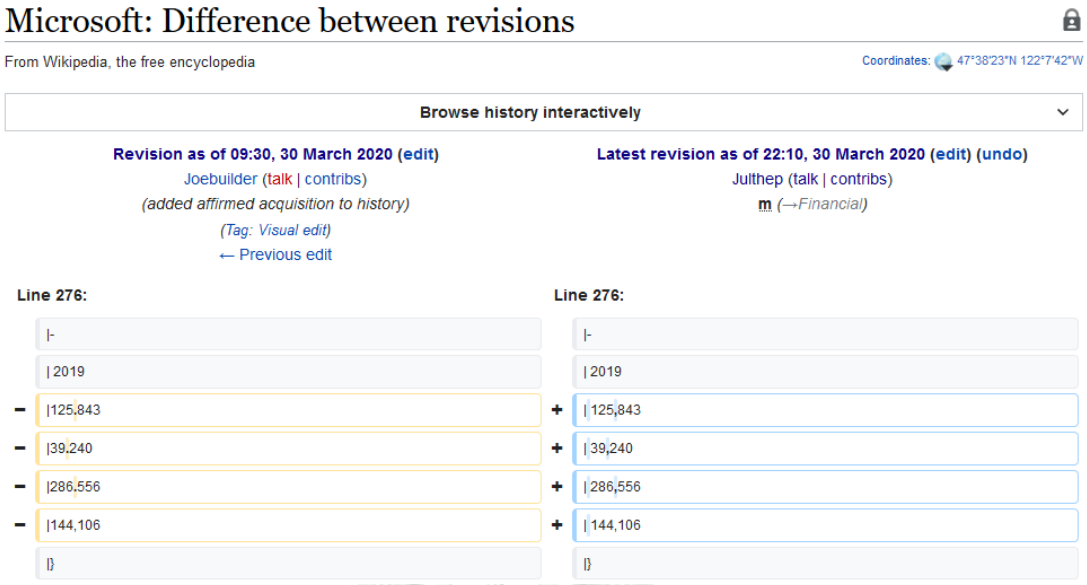
Revision as of 05:37, 16 August 2019 (edit)	Revision as of 18:58, 31 August 2019 (edit) (undo) (thank)
Dawnseeker2000 (talk | contribs)	Jicco123 (talk | contribs)
(date formats per MOS:DATEFORMAT by script)	*(→Financial)*
← Previous edit	Next edit →

Line 251:	Line 251:
| 258,848	| 258,848
| 131,000	| 131,000
	+ |-
	+ | 2019
	+ |125.843
	+ |39.240
	+ |286.556
	+ |144,106
|}	|}

ภาพที่ 64 ข้อมูลของบทความ “Microsoft” ในวิกิพีเดียที่บันทึกไว้ตั้งแต่วันที่ 31 สิงหาคม 2019

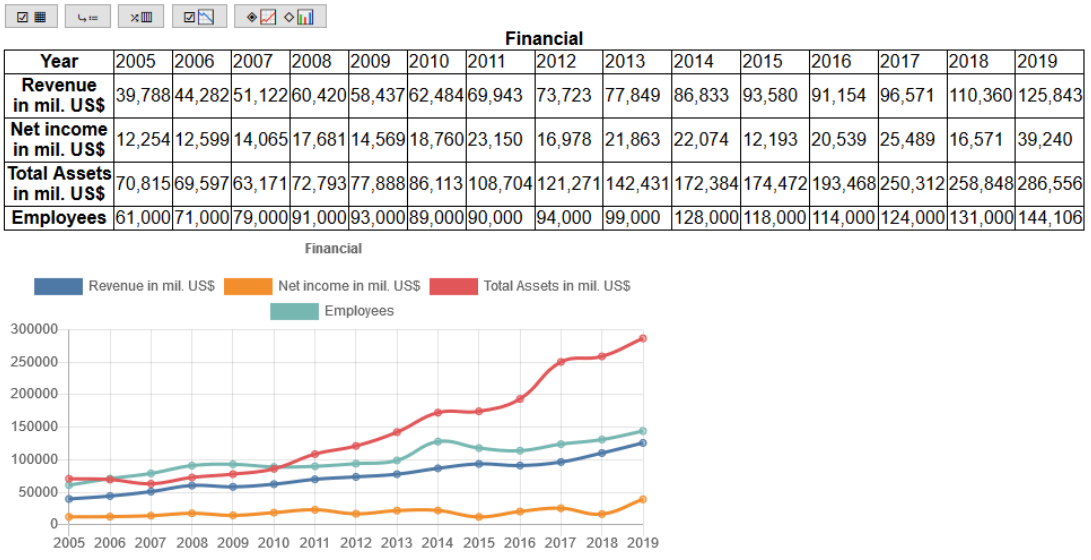
ผู้วิจัยคาดเดาว่าผู้บันทึกเป็นผู้ที่ไม่ได้ใช้ภาษาอังกฤษเป็นภาษาหลัก ซึ่งประเทศเหล่านั้นส่วนใหญ่จะใช้ decimal comma[45] คือการใช้เครื่องหมายจุลภาค (,) คั่นระหว่างเลขทศนิยม และใช้เครื่องหมายมหัพภาค (.) คั่นระหว่างหลักพัน ดังนั้นด้วยความเคยชินผู้บันทึกจึงใส่ข้อมูลที่เป็น decimal comma ไปผสมรวมลงไปในตารางเดิมที่เป็น decimal point ตามมาตรฐานของวิกิพีเดียภาษาอังกฤษ

[45] Decimal separator – Countries using decimal comma. Wikipedia. Available from: https://en.wikipedia.org/wiki/Decimal_separator#Countries_using_decimal_comma



ภาพที่ 65 ผู้วิจัยทำการแก้ไขเครื่องหมายจากเดิม decimal comma เป็น decimal point

ดังนั้นเมื่อผู้วิจัยพบว่าข้อมูลในวิกิพีเดียมีความผิดพลาด ในฐานะวิกิพีเดียนจึงได้ทำการปรับปรุงแก้ไขเครื่องหมายจาก decimal comma เป็น decimal point ตามภาพที่ 65 จะเห็นได้ว่าข้อผิดพลาดนี้ถูกบันทึกอยู่ในวิกิพีเดียเป็นเวลาถึง 7 เดือนเต็ม ก่อนที่จะถูกตรวจพบและแก้ไขโดยผู้วิจัยเมื่อวันที่ 30 มีนาคม 2020



Financial

Year	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019
Revenue in mil. US$	39,788	44,282	51,122	60,420	58,437	62,484	69,943	73,723	77,849	86,833	93,580	91,154	96,571	110,360	125,843
Net income in mil. US$	12,254	12,599	14,065	17,681	14,569	18,760	23,150	16,978	21,863	22,074	12,193	20,539	25,489	16,571	39,240
Total Assets in mil. US$	70,815	69,597	63,171	72,793	77,888	86,113	108,704	121,271	142,431	172,384	174,472	193,468	250,312	258,848	286,556
Employees	61,000	71,000	79,000	91,000	93,000	89,000	90,000	94,000	99,000	128,000	118,000	114,000	124,000	131,000	144,106

ภาพที่ 66 เมื่อแสดงผลตารางและชาร์ตใหม่ หลังจากปรับปรุงแก้ไขข้อมูลเสร็จเรียบร้อย

หลังจากปรับปรุงแก้ไขข้อมูลเสร็จเรียบร้อย เมื่อแสดงผลตารางและชาร์ตที่อัพเดทใหม่ใน TLPedia ก็จะได้ดังภาพที่ 66

ผลลัพธ์ที่ได้จากการทดลอง

ผู้วิจัยได้ทำการทดสอบการแปลงข้อมูลโดยใช้ระบบทดสอบ TLPedia สร้างเดต้าเซ็ตขึ้นมาจากกลุ่มของบทความจากวิกิพีเดียจำนวน 389,319 บทความ ซึ่งนำมาจากชุดของบทความ 10 ชุดที่ถูกคัดกรองโดยการใช้ Infobox template เป็นตัวแบ่งกลุ่ม โดยยืนพื้นจากบทความในชุดของ film แล้วคัดเลือกชุดบทความอีก 9 ชุดที่มีความเชื่อมโยงถึงกันในเนื้อหาของบทความที่เกี่ยวกับภาพยนตร์ โดยส่วนใหญ่ คือ automobile, award, book, character, company, film_awards, historic_site, musical_artist และ recurring_event โดยแต่ละชุดบทความมีความแตกต่างกันทั้งในแง่ของจำนวนบทความและสัดส่วนปริมาณของข้อมูลตารางและรายการที่อยู่ในบทความโดยเฉลี่ย ใช้เวลาในการแปลงข้อมูลทั้งหมดรวมทั้งสิ้น 129 ชั่วโมง 31 นาที 31 วินาที คิดเป็นระยะเวลาในการแปลงข้อมูลโดยเฉลี่ย 1.198 วินาทีต่อบทความ โดยทำงานบนเครื่องเซิร์ฟเวอร์เสมือนขนาด 4 แกน ด้วยตัวประมวลผล Intel Xeon E5-2640 v3 2.60GHz หน่วยความจำขนาด 4GB และเชื่อมต่ออินเตอร์เน็ตด้วยความเร็ว 200 Mbps

ตารางที่ 9 ผลการทดลองแปลงข้อมูลจากบทความวิกิพีเดีย

Article Datasets	No. of Articles	No. Of Issues	Issues Articles Ratio	Total Dataset Size (KB) WIKI	Total Dataset Size (KB) TULIP	Avg. Article Size (KB) WIKI	Avg. Article Size (KB) TULIP	TULIP WIKI Ratio	Processing Time Total (hrs.)	Processing Time Article (sec.)
automobile	7,204	2	0.028%	665,228	163,716	92	23	24.6%	2:15:10	1.126
award	9,847	85	0.863%	884,944	318,584	90	32	36.0%	4:27:39	1.631
book	43,572	3	0.007%	2,372,978	633,188	54	15	26.7%	11:42:09	0.967
character	6,181	4	0.065%	597,512	142,872	97	23	23.9%	1:55:48	1.124
company	65,911	42	0.064%	4,326,895	1,194,928	66	18	27.6%	19:51:24	1.085
film	139,079	33	0.024%	7,771,535	2,409,876	56	17	31.0%	48:29:13	1.255
film_awards	2,133	12	0.563%	169,932	54,592	80	26	32.1%	0:51:33	1.450
historic_site	8,868	1	0.011%	618,304	142,308	70	16	23.0%	2:56:25	1.194
musical_artist	101,426	88	0.087%	7,083,445	2,188,580	70	22	30.9%	35:02:47	1.244
recurring_event	5,098	15	0.294%	413,420	129,688	81	25	31.4%	1:59:23	1.405
Total	389,319	285	0.073%	24,904,193	7,378,332	64	19	29.6%	129:31:31	1.198

จากผลการทดลองแปลงข้อมูลตามตารางที่ 9 วิเคราะห์ผลได้ดังนี้ บทความที่มีปริมาณข้อมูลตารางและรายการต่อบทความสูงที่สุดคือบทความที่เกี่ยวกับการจัดดับภาพยนตร์ และงานมอบรางวัลที่มีเป็นประจำทุกปี เช่น award, film_awards และ recurring_event ซึ่งบทความในกลุ่มนี้ส่วนใหญ่จะประกอบด้วยตารางและรายการเพื่อให้สะดวกในการนำเสนอเนื้อหา โดยวัดได้จากสัดส่วน TULIP/WIKI ratio คือการนำเอาปริมาณโดยเฉลี่ยของไฟล์ HTML ที่ได้มาจาก TULIP ที่สกัดเฉพาะ

ข้อมูลตารางและรายการ นำมาหารด้วยไฟล์ HTML ต้นฉบับของวิกิพีเดียเอง สัดส่วนร้อยละที่สูงกว่าหมายถึงมีปริมาณตารางและรายการอยู่ในบทความมากกว่า ซึ่งบทความที่มีสัดส่วนที่ต่ำที่สุดคือ book และ historic_site ที่ส่วนใหญ่จะเป็นเนื้อหาเชิงพรรณนาอยู่ภายในบทความ

ซึ่งสอดคล้องกับปริมาณข้อผิดพลาดที่เกิดจากปัญหาของตารางในบทความของชุดข้อมูล award, film_awards และ recurring_event โดยเกิดได้จากหลายสาเหตุเพราะตารางในวิกิพีเดียส่วนใหญ่จะถูกสร้างและแก้ไขด้วยมนุษย์ ซึ่งจะมีความผิดพลาดทางรูปแบบปะปนอยู่บ้าง แม้ว่าจะมองไม่ออกด้วยสายตามนุษย์แต่ระบบตรวจพบได้และทำการบันทึกข้อผิดพลาดเอาไว้ โดยในตารางได้สรุปไว้ในสดมภ์ Issues/Articles หมายถึงจำนวนบทความที่พบข้อผิดพลาดในตารางและรายการเมื่อเปรียบเทียบกับปริมาณบทความทั้งหมด

จำนวนข้อผิดพลาดในเนื้อหาของตารางสามารถแยกประเภทได้ดังตารางที่ 10

ตารางที่ 10 จำนวนข้อผิดพลาดของตารางที่พบในบทความต้นฉบับแยกตามประเภท

Error messages	Count
list index out of range	61
invalid literal for int() with base 10: '2;'	45
invalid literal for int() with base 10: ''	15
invalid literal for int() with base 10: '"2"'	10
resource not found	9
invalid literal for int() with base 10: '2"'	8
invalid literal for int() with base 10: '2data-sort-value=""'	5
invalid literal for int() with base 10: '3;'	5
invalid literal for int() with base 10: '3data-sort-value=""'	5
invalid literal for int() with base 10: '4"'	5
invalid literal for int() with base 10: '5;'	5
invalid literal for int() with base 10: '7"'	5
invalid literal for int() with base 10: "2""	4
invalid literal for int() with base 10: '1”'	4
invalid literal for int() with base 10: '5"'	4
invalid literal for int() with base 10: '“2”'	3

Error messages	Count
invalid literal for int() with base 10: '1"'	3
invalid literal for int() with base 10: '3"'	3
invalid literal for int() with base 10: '6"'	3
invalid literal for int() with base 10: '6”'	3
invalid literal for int() with base 10: 'center'	3
invalid literal for int() with base 10: "1""	2
invalid literal for int() with base 10: "3""	2
invalid literal for int() with base 10: "5"	2
invalid literal for int() with base 10: '`'	2
invalid literal for int() with base 10: '“1"'	2
invalid literal for int() with base 10: '“3”'	2
invalid literal for int() with base 10: '10"'	2
invalid literal for int() with base 10: '10;'	2
invalid literal for int() with base 10: '2 data-sort-value='	2
invalid literal for int() with base 10: '2”'	2
invalid literal for int() with base 10: '4;'	2
invalid literal for int() with base 10: '7;'	2
invalid literal for int() with base 10: '8”'	2
invalid literal for int() with base 10: 'a'	2
invalid literal for int() with base 10: 'row'	2
invalid literal for int() with base 10: '!'	1
invalid literal for int() with base 10: '!1"'	1
invalid literal for int() with base 10: "3"	1
invalid literal for int() with base 10: "4"	1
invalid literal for int() with base 10: '???????'	1
invalid literal for int() with base 10: '‘’2’’'	1
invalid literal for int() with base 10: '“4”'	1
invalid literal for int() with base 10: '“8”'	1
invalid literal for int() with base 10: '»1"'	1

Error messages	Count
invalid literal for int() with base 10: '1 data-sort-value='	1
invalid literal for int() with base 10: '1!'	1
invalid literal for int() with base 10: '1.75'	1
invalid literal for int() with base 10: '1” data-sort-value='	1
invalid literal for int() with base 10: '1”"'	1
invalid literal for int() with base 10: '10”'	1
invalid literal for int() with base 10: '11”'	1
invalid literal for int() with base 10: '12"'	1
invalid literal for int() with base 10: '14%'	1
invalid literal for int() with base 10: '15”'	1
invalid literal for int() with base 10: '15='	1
invalid literal for int() with base 10: '17 data-sort-value='	1
invalid literal for int() with base 10: '1pe'	1
invalid literal for int() with base 10: '2 align='	1
invalid literal for int() with base 10: '2!'	1
invalid literal for int() with base 10: '2"'	1
invalid literal for int() with base 10: '2.5'	1
invalid literal for int() with base 10: '2:'	1
invalid literal for int() with base 10: '2“'	1
invalid literal for int() with base 10: '2+'	1
invalid literal for int() with base 10: '2='	1
invalid literal for int() with base 10: '25"'	1
invalid literal for int() with base 10: '4"'	1
invalid literal for int() with base 10: '4”'	1
invalid literal for int() with base 10: '4=1'	1
invalid literal for int() with base 10: '4data-sort-value=""'	1
invalid literal for int() with base 10: '5”'	1
invalid literal for int() with base 10: '6” style='	1
invalid literal for int() with base 10: '6style="text-align:center"'	1

Error messages	Count
invalid literal for int() with base 10: '8"'	1
invalid literal for int() with base 10: '8;'	1
invalid literal for int() with base 10: '8” align=center'	1
invalid literal for int() with base 10: '9"'	1
invalid literal for int() with base 10: '9align="center"'	1
invalid literal for int() with base 10: 'B'	1
invalid literal for int() with base 10: 'C'	1
invalid literal for int() with base 10: 'data-sort-value=""'	1
invalid literal for int() with base 10: 'style="width:215px;"'	1
Total	285

โดยส่วนใหญ่ข้อผิดพลาดจะเกิดมาจากการที่ผู้แก้ไขข้อมูลทำการแก้ไขตารางที่อยู่ในรูปแบบ Wikitext markdown โดยตรง โดยไม่ใช้เครื่องมือสำหรับการแก้ไขตารางที่มีเพิ่มเติมให้ใช้ในวิกิพีเดียรุ่นหลัง ๆ ทำให้เกิดข้อผิดพลาดในหลายกรณี ซึ่งโดยส่วนใหญ่ TULIP ไลบรารีสามารถแก้ไขข้อผิดพลาดได้เอง ยกตัวอย่างเช่น กรณีที่มีตัวอักษรที่ไม่ใช่ตัวเลขเกินมาในข้อมูลที่ควรจะมีเฉพาะตัวเลขเท่านั้น แต่มีอยู่กรณีหนึ่งซึ่งเป็นกรณีที่พบมากที่สุดคือการกำหนดขอบเขตของเซลล์หรือเลย์เอาต์ผิดพลาดทำให้ตารางมีความไม่สมบูรณ์ ยกตัวอย่างเช่น บทความชื่อ Khwina (เอดิชันไอดี 972439834 เมื่อวันที่ 12 สิงหาคม ค.ศ. 2020) พบว่ามีตารางดังภาพที่ 67 ที่มีความไม่สมบูรณ์เนื่องมาจากความผิดพลาดในการกำหนดขอบเขตของเซลล์ที่ไม่ถูกต้อง ซึ่งในกรณีนี้ TULIP ไลบรารียังไม่สามารถทำการแก้ไขตารางนี้ได้โดยอัตโนมัติ

Awards

Award	Category	Winner's name	Result			
2nd Bodo Film Festival 2017	Best Film-	Phaylaw Basumatary	Won			
	Best Actor- Male	Omprakash Kherkatary	Won -	Best Villain- Male	Jesus Kherkatary	Won

ภาพที่ 67 ตารางในวิกิพีเดียที่เกิดข้อผิดพลาดจากการแก้ไขที่ไม่สมบูรณ์

บทที่ 6

บทสรุปงานวิจัยและข้อเสนอแนะ

บทนี้จะกล่าวถึงความรู้ที่ได้จากการดำเนินงานวิจัยนี้ รวมไปถึงผลงานที่เผยแพร่เป็นประโยชน์ต่อสาธารณะ

ความรู้ที่ได้จากการดำเนินงานวิจัย

ผู้วิจัยได้ความรู้มากมายในขณะที่ทำการค้นคว้าเพื่อดำเนินงานวิจัยชิ้นนี้ ได้พบแง่คิดและมุมมองที่แตกต่างออกไปในการออกแบบโมเดลข้อมูลชนิดใหม่เพื่อแก้โจทย์ปัญหาที่ตั้งไว้ ทำให้ต้องทำการทดลองหาความเหมาะสมเพื่อให้สามารถนำไปใช้งานได้จริง อีกทั้งยังได้ศึกษาค้นพบกลวิธีด้านการพัฒนาซอฟต์แวร์หลายประการในการออกแบบและพัฒนาไลบรารีต้นแบบและระบบตัวอย่าง

เปรียบเทียบข้อเสนอกับงานวิจัยที่มีอยู่เดิม

งานวิจัยนี้มีความแตกต่างจากงานวิจัยที่มีอยู่เดิมที่ส่วนใหญ่จะเน้นในแง่ของการสกัดข้อมูลจากตารางและรายการออกมาเป็นข้อเท็จจริงในรูปแบบต่าง ๆ แต่งานวิจัยนี้มุ่งเน้นไปที่การสกัดข้อมูลจากตารางและรายการมาเป็นเดต้าเซ็ตในรูปแบบของลิงก์เดต้าที่ระดับ 5 ดาว นอกจากนั้นชุดของ RDF ทริปเปิลที่ได้ยังสามารถนำมาสร้างตารางและรายการได้ในรูปแบบเดียวกับข้อมูลต้นทาง เนื่องจากสกีมาที่ออกแบบไว้มุ่งเน้นไปที่ความสามารถในการเก็บลักษณะโครงสร้างของตารางและรายการต้นฉบับเอาไว้ได้ ที่สำคัญอีกประการหนึ่งคือชุด RDF ทริปเปิลที่ได้มานอกจะเก็บไว้ในฐานความรู้ลิงก์เดต้าแล้ว ยังสามารถนำไปฝังไว้ในไฟล์หีบห่อรูปแบบต่าง ๆ เช่น XML หรือ HTML (ในรูปแบบ RDFa) เพื่อนำไปใช้สร้างตารางและรายการในเว็บเพจหรือแอปพลิเคชันต่าง ๆ พร้อมกับมีข้อเท็จจริงอยู่ภายในผนวกไปด้วยกันเพื่อความสะดวกในการใช้งาน

ความท้าทายของงานวิจัย

ความท้าทายของงานวิจัยมีดังนี้ เรื่องแรกเกี่ยวกับเทคนิควิธีการแปลงข้อมูลจากตารางและรายการมาเป็น RDF ทริปเปิล ถึงแม้จะมีงานวิจัยมากมาย[108-110] ที่เกี่ยวกับการแปลงข้อมูลตาราง

และรายการเป็นข้อมูลรูปแบบอื่น ๆ แต่ปัญหาที่แท้จริงของการแปลงข้อมูลไม่ได้อยู่ที่ “โครงสร้าง” ของข้อมูล แต่อยู่ที่ “รูปแบบ” และ “เนื้อหา” ของข้อมูลต้นฉบับ เช่น จะมีวิธีจัดการกับรูปแบบของตัวอักษรรวมทั้งรูปแบบของเซลล์ของตารางและจะให้ “ความหมาย” กับสิ่งเหล่านี้ได้อย่างไร หรือจะแม็ปสกีมาลงไปที่ column header และ row header ได้อย่างไร[60] จำเป็นต้องเปลี่ยนแปลงเนื้อหาข้อความหรือไม่เพื่อให้สอดคล้องกันตลอดทั้งเดต้าเซ็ต การเลือกใช้เทคนิคที่หลากหลายเพื่อจัดการกับความกำกวม[111] ทั้ง schema matching[112, 113] หรือ ontology mapping[114, 115] จะกำหนดสกีมาอย่างไรและควรกำหนดเป็นมาตรฐานหรือไม่[116] รวมทั้งจะจัดการความเป็นระเบียบและความกระชับของเนื้อหาที่แปลงมาได้อย่างไร

เรื่องถัดมาเป็นการออกแบบการรีพรีเซนต์ตารางและรายการด้วย RDF ทริปเปิล โดยที่ยังสามารถรักษาคุณสมบัติทั้งสองด้านของสารสนเทศนี้ไว้ได้ครบถ้วน คุณสมบัติแรกคือ RDF ทริปเปิลที่ได้จะต้องสามารถนำมาสร้างตารางและรายการได้ในรูปแบบต้นทาง และในรูปแบบอื่น ๆ ที่แตกต่างจากเดิม เพื่อวัตถุประสงค์ในการนำเสนอข้อมูลเดียวกันในรูปแบบที่หลากหลายมากขึ้น เช่น นำมาสนับสนุนการสร้างอินโฟกราฟฟิกและแผนภูมิต่าง ๆ เป็นต้น คุณสมบัติอีกอย่างหนึ่งคือ RDF ทริปเปิลที่ได้ จะต้องสามารถนำมาใช้งานได้ในรูปแบบลิงก์เดต้าโดยสมบูรณ์ คือในข้อเท็จจริงที่แปลงได้มาจะต้องสามารถใช้ SPARQL คิวรีในการสืบค้นคำตอบ หรือเชื่อมโยงความสัมพันธ์กันในระหว่างเดต้าเซ็ตเดียวกันและเดต้าเซ็ตอื่น ๆ ได้อีกด้วย

ในกรณีของรายการ แทนที่แต่ละรายการจะถูกนำมาสร้างเป็นข้อเท็จจริงได้โดยตรง อาจจำเป็นต้องเจอกรณีที่แต่ละรายการมีลักษณะเป็นประโยคข้อความยาว ๆ ตั้งแต่หนึ่งประโยคขึ้นไป ซึ่งอาจจะต้องนำเทคนิคต่าง ๆ ทางด้าน NLP มาใช้ เช่น named-entity recognition (NER), named entity linking (NEL) หรือ named entity disambiguation (NED) ในการสกัดข้อเท็จจริงออกมาจากประโยคเหล่านั้น

ผลงานที่เผยแพร่เป็นประโยชน์ต่อสาธารณะ

Contributions ที่ได้จากงานวิจัยนี้สามารถสรุปเป็นหัวข้อได้ดังนี้

1. TULIP RDF เดต้าโมเดลและ RDF vocabulary/namespace เพื่อรีพรีเซนต์ตารางและรายการ (รวมไปถึง DOM elements อื่น ๆ) ให้อยู่ในรูปแบบลิงก์เดต้าระดับ 5 ดาว

โดย specification และ vocabulary/namespace prefix อยู่ที่ http://rdf.vc/tulip/spec และ http://purl.org/tulip/ns# ตามลำดับ

2. Python reference library สำหรับแปลงตารางและรายการ (รวมไปถึง DOM elements แบบอื่น ๆ) ในรูปแบบ HTML ให้อยู่ในรูปแบบ TULIP Python อ็อบเจ็กต์ และ TULIP RDF ในฟอร์แมทต่าง ๆ เช่น Turtle, N-Triples, Notation3 ฯลฯ โดยมีซอร์สโค้ดอยู่ที่ http://github.com/julthep/tulip/ และไลบรารีแพ็คเกจ PyPI อยู่ที่ http://pypi.org/project/tulip-rdf/

3. JavaScript reference library สำหรับอ่านตารางและรายการ (รวมไปถึง DOM elements แบบอื่น ๆ) ในรูปแบบ TULIP RDF ในฟอร์แมทต่าง ๆ เข้ามาอยู่ในรูปแบบ TULIP JavaScript อ็อบเจ็กต์เพื่อให้สามารถนำมาแสดงผลด้วย HTML ในรูปแบบต่าง ๆ ซอร์สโค้ดอยู่ที่ http://github.com/julthep/tulip.js/

4. เดต้าเซ็ตที่ได้จากการแปลงข้อมูลบทความจาก Wikipedia มาอยู่ในรูปแบบ TULIP RDF มีชื่อว่า TLPedia อยู่ที่ http://www.tlpedia.org/datasets/ และสามารถเข้าถึงผ่าน namespace ที่ http://tlpedia.org/resource (หรือ http://tlpedia.org/object ในรูปแบบ JSON อ็อบเจ็กต์ของ TULIP คลาส และ http://tlpedia.org/page ในรูปแบบ HTML ที่ผ่านการแปลงมาแล้วโดย TULIP ไลบรารี) โดยมี SPARQL endpoint อยู่ที่ http://rql.tlpedia.org/sparql

5. โปรแกรมตัวอย่างในการนำ TULIP เดต้าเซ็ตไปใช้นำการจำลองว่าเครื่องสามารถจะเข้าใจข้อมูล TULIP ในรูปแบบ fully structured data ได้ในลักษณะแบบเดียวกับที่มนุษย์จะเข้าใจข้อมูลในรูปแบบ HTML สาธิตโดยการนำข้อมูลใน TULIP มาแสดงผลในรูปแบบ HTML ผ่าน TLPedia browser ที่ http://dataplay.cc/tlpedia/

สรุปสาระสำคัญที่ได้จากงานวิจัย

งานวิจัยนี้มุ่งเน้นไปที่การทำให้เครื่องมีความเข้าใจในข้อมูลที่ถูกสร้างโดยมนุษย์มาเพื่อให้มนุษย์เข้าใจโดยตรง ไม่ว่าจะเป็นข้อมูลไร้โครงสร้างหรือกึ่งโครงสร้าง ซึ่งอาจจะมีความแตกต่างไปจากข้อมูลที่สร้างขึ้นมาโดยเครื่องด้วยวิธีการต่าง ๆ เช่นการบันทึกข้อมูลเชิงโครงสร้างเข้าไปในระบบโดยตรงผ่านโปรแกรมสำเร็จรูป, การเก็บข้อมูลโดยเซนเซอร์, หรือการสังเคราะห์ข้อมูลโดยเครื่อง เป็น

ต้น ความเข้าใจในข้อมูลชุดนี้นี้อาจจะเป็นไปในรูปแบบเดียวกันกับมนุษย์ ตัวอย่างเช่น การดูตารางแล้วเห็นความสัมพันธ์ของข้อมูลสองมิติ ดูชาร์ตแล้วเห็นแนวโน้มและความผิดปกติของข้อมูล เป็นต้น หรืออาจจะเข้าใจในรูปแบบที่แตกต่างจากมนุษย์โดยสิ้นเชิง ตัวอย่างเช่น การทำเหมืองข้อมูลเพื่อค้นพบ insight บางอย่างจากข้อมูล ซึ่งไม่สามารถมองเห็นได้ในทางตรง เป็นต้น ส่วนการที่มนุษย์จะได้ประโยชน์จากเครื่องมือที่ได้มาจากการที่เครื่องสามารถเข้าใจข้อมูลนั้น เป็นผลโดยทางอ้อมหรือ indirect result ซึ่งมีความเป็นไปได้อย่างหลากหลายนอกเหนือไปจากที่ได้ยกตัวอย่างโดยระบบต้นแบบในงานวิจัย

ทุกวันนี้มีการสร้างโปรแกรมมากมายที่สามารถนำเข้าข้อมูลได้จากหลายแหล่งข้อมูล ไม่ว่าข้อมูลนั้นจะอยู่ในรูปแบบเปิดหรือแบบปิด ตามที่ได้มีการแยกแยะไว้เป็นลำดับขั้นตามหลักของลิงก์โอเพ่นเดต้าระดับ 5 ดาว แต่โปรแกรมโดยส่วนใหญ่เป็นการสร้างมาโดยเฉพาะสำหรับข้อมูลใดข้อมูลหนึ่งเท่านั้น และสร้างโดยมนุษย์ (ระบุอย่างเจาะจงคือนักวิทยาศาสตร์ข้อมูล) ที่ต้องศึกษาเข้าใจที่มา, คุณลักษณะ และรูปแบบโครงสร้างของข้อมูลก่อนที่จะนำมาใช้ แต่งานวิจัยนี้มุ่งเน้นทำให้ข้อมูลที่หลายหลายเหล่านั้นมาอยู่ในรูปแบบเดียวกัน, ในโครงเสร้างเดียวกัน และมีความเชื่อมโยงถึงกันทั้งหมด ไม่ว่าจะอยู่ในโดเมนใด ๆ เพื่อให้เครื่องสามารถนำข้อมูลเหล่านั้นมาวิเคราะห์โดยรวมได้ในแบบทั่วไปไม่เฉพาะเจาะจง ซึ่งในท้ายที่สุดจะทำให้เราสามารถมีโปรแกรมเพียงหนึ่งเดียวสำหรับข้อมูลที่มีความหลากหลาย ทั้งที่มา, คุณลักษณะ, และรูปแบบโครงสร้าง

รายการอ้างอิง

[1] Berners-Lee T, Cailliau R, Luotonen A, Nielsen HF, Secret A. The World-Wide Web. Commun ACM. 1994;37(8):76-82.

[2] Berners-Lee T. Information management: A proposal. Switzerland: CERN; 1989. Report No.: CERN-DD-89-001-OC.

[3] Spivack N. Web 3.0--The Best Official Definition Imaginable [Internet]. 2007. Available from: http://www.novaspivack.com/technology/web-3-0-the-best-official-definition-imaginable [Accessed: 2019-12-20].

[4] Richardson W. Blogs, Wikis, Podcasts, and Other Powerful Web Tools for Classrooms. 2nd ed. California: Corwin Press; 2009.

[5] Berners-Lee T, Hendler J, Lassila O. The Semantic Web: A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities. Scientific American. 2001;284(5):34-43.

[6] Bratt S. Semantic Web and Other W3C Technologies to Watch [Internet]. W3C; 2006. Available from: https://www.w3.org/2006/Talks/1023-sb-W3CTechSemWeb [Accessed: 2019-12-20].

[7] Codd EF. A relational model of data for large shared data banks. Commun ACM. 1970;13(6):377-387.

[8] Berners-Lee T, Connolly D, Kagal L, Scharf Y, Hendler J. N3Logic: A logical framework for the World Wide Web. Theory and Practice of Logic Programming. 2008;8(3):249-269.

[9] Shadbolt N, Hall W, Berners-Lee T. The Semantic Web Revisited. Intelligent Systems, IEEE. 2006;21(3):96-101.

[10] Bizer C, Heath T, Berners-Lee T. Linked Data - The Story So Far. Semantic services, interoperability and web applications: emerging concepts. 2009:205-227.

[11] Bizer C. The emerging web of linked data. IEEE intelligent systems. 2009;24(5).

[12] Färber M, Ell B, Menne C, Rettinger A. A Comparative Survey of DBpedia, Freebase, OpenCyc, Wikidata, and YAGO. Semantic Web Journal. 2015;1(1):1-5.

[13] Färber M, Bartscherer F, Menne C, Rettinger A. Linked Data Quality of DBpedia, Freebase, OpenCyc, Wikidata, and YAGO. Semantic Web. 2018;9(1):77-129.

[14] Pillai SG, Soon L-K, Haw S-C. Comparing DBpedia, Wikidata, and YAGO for Web Information Retrieval. Intelligent and Interactive Computing. Singapore: Springer; 2019. p. 525-535.

[15] Ringler D, Paulheim H, editors. One knowledge graph to rule them all? Analyzing the differences between DBpedia, YAGO, Wikidata & co. Joint German/Austrian Conference on Artificial Intelligence (Künstliche Intelligenz); 2017 2017 Sep 25. Cham: Springer.

[16] Lehmann J, Isele R, Jakob M, Jentzsch A, Kontokostas D, Mendes PN, et al. DBpedia – A large-scale, multilingual knowledge base extracted from Wikipedia. Semantic Web. 2015;6(2):167-195.

[17] Auer S, Lehmann J. What Have Innsbruck and Leipzig in Common? Extracting Semantics from Wiki Content. In: Franconi E, Kifer M, May W, editors. 4th European Semantic Web Conference, ESWC 2007, June 3-7, 2007 Proceedings; 2007/01/01; Innsbruck, Austria: Springer Berlin Heidelberg; 2007. p. 503-517.

[18] Bizer C, Lehmann J, Kobilarov G, Auer S, Becker C, Cyganiak R, et al. DBpedia - A crystallization point for the Web of Data. Web Semantics: Science, Services and Agents on the World Wide Web. 2009;7(3):154-165.

[19] Auer S, Bizer C, Kobilarov G, Lehmann J, Cyganiak R, Ives Z. DBpedia: A Nucleus for a Web of Open Data. In: Aberer K, Choi K-S, Noy N, Allemang D, Lee K-I, Nixon L, et al., editors. 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, November 11-15, 2007 Proceedings; 2007/01/01; Busan, Korea: Springer Berlin Heidelberg; 2007. p. 722-735.

[20] Isbell J, Butler MH. Extracting and re-using structured data from wikis. Bristol: Digital Media Systems Laboratory of Hewlett-Packard Development Company; 2007 14 November. Report No.: HPL-2007-182.

[21] Suchanek FM, Kasneci G, Weikum G, editors. YAGO: A Core of Semantic Knowledge Unifying WordNet and Wikipedia. Proceedings of the 16th

international conference on World Wide Web; 2007; Banff, Alberta, Canada. 1242667: ACM.

[22] Navigli R, Ponzetto SP. BabelNet: building a very large multilingual semantic network. Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics; Uppsala, Sweden. 1858704: Association for Computational Linguistics; 2010. p. 216-225.

[23] Melo Gd, Weikum G. MENTA: Inducing Multilingual Taxonomies from Wikipedia. Proceedings of the 19th ACM International Conference on Information and Knowledge Management; Toronto, ON, Canada. 1871577: ACM; 2010. p. 1099-1108.

[24] Bollacker K, Evans C, Paritosh P, Sturge T, Taylor J, editors. Freebase: a collaboratively created graph database for structuring human knowledge. Proceedings of the 2008 ACM SIGMOD international conference on Management of data; 2008 Jun 9: ACM.

[25] Bollacker K, Tufts P, Pierce T, Cook R, editors. A platform for scalable, collaborative, structured information integration. Intl Workshop on Information Integration on the Web (IIWeb'07); 2007 July.

[26] Dong X, Gabrilovich E, Heitz G, Horn W, Lao N, Murphy K, et al. Knowledge vault: a web-scale approach to probabilistic knowledge fusion. Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining; New York, New York, USA. 2623623: ACM; 2014. p. 601-610.

[27] Bollacker K, Cook R, Tufts P, editors. Freebase: a shared database of structured general human knowledge. Proceedings of the 22nd national conference on Artificial intelligence-Volume 2; 2007 Jul 22: AAAI Press.

[28] Vrandečić D, Krötzsch M. Wikidata: a free collaborative knowledgebase. Communications of the ACM. 2014;57(10):78-85.

[29] Vrandečić D. The rise of Wikidata. IEEE Intelligent Systems. 2013;28(4):90-95.

[30] Abián D, Guerra F, Martínez-Romanos J, Trillo-Lado R, editors. Wikidata and DBpedia: a comparative study. Semanitic Keyword-based Search on Structured Data Sources; 2017 Sep 11; Cham: Springer.

[31] Ismayilov A, Kontokostas D, Auer S, Lehmann J, Hellmann S. Wikidata through the Eyes of DBpedia. Semantic Web. 2018;9(4):493-503.

[32] Frey J, Hofer M, Obraczka D, Lehmann J, Hellmann S, editors. DBpedia FlexiFusion the Best of Wikipedia> Wikidata> Your Data. International Semantic Web Conference; 2019 Oct 26; Cham: Springer.

[33] Lenat DB. CYC: a large-scale investment in knowledge infrastructure. Commun ACM. 1995;38(11):33-38.

[34] Lenat DB. The voice of the turtle: Whatever happened to AI? AI Magazine. 2008;29(2):11.

[35] Lenat DB. Building a Machine Smart Enough to Pass the Turing Test. In: Epstein R, Roberts G, Beber G, editors. Parsing the Turing Test: Philosophical and Methodological Issues in the Quest for the Thinking Computer. Dordrecht: Springer Netherlands; 2009. p. 261-282.

[36] Han L, Finin T, Parr C, Sachs J, Joshi A. RDF123: From Spreadsheets to RDF. In: Sheth A, Staab S, Dean M, Paolucci M, Maynard D, Finin T, et al., editors. The Semantic Web - ISWC 2008. Lecture Notes in Computer Science. 5318: Springer Berlin Heidelberg; 2008. p. 451-466.

[37] Sahoo SS, Halb W, Hellmann S, Idehen K, Thibodeau Jr T, Auer S, et al. A survey of current approaches for mapping of relational databases to RDF. W3C RDB2RDF Incubator Group Report; 2009.

[38] Yang Y. Web table mining and database discovery: Simon Fraser University; 2002.

[39] Yang Y, Luk W-S. A framework for web table mining. Proceedings of the 4th international workshop on Web information and data management; McLean, Virginia, USA. 584940: ACM; 2002. p. 36-42.

[40] Pivk A, Cimiano P, Sure Y. From Tables to Frames. In: McIlraith SA, Plexousakis D, van Harmelen F, editors. The Semantic Web – ISWC 2004: Third International Semantic Web Conference, Hiroshima, Japan, November 7-11, 2004 Proceedings. Berlin, Heidelberg: Springer Berlin Heidelberg; 2004. p. 166-181.

[41] Embley DW, Tao C, Liddle SW. Automatically Extracting Ontologically Specified Data from HTML Tables of Unknown Structure. In: Spaccapietra S, March ST, Kambayashi Y, editors. Conceptual Modeling — ER 2002: 21st International

Conference on Conceptual Modeling Tampere, Finland, October 7–11, 2002 Proceedings. Berlin, Heidelberg: Springer Berlin Heidelberg; 2003. p. 322-337.

[42] Tijerino YA, Embley DW, Lonsdale DW, Ding Y, Nagy G. Towards Ontology Generation from Tables. World Wide Web. 2005;8(3):261-285.

[43] Tijerino YA, Embley DW, Lonsdale DW, Nagy G, editors. Ontology generation from tables. Proceedings of the Fourth International Conference on Web Information Systems Engineering, WISE; 2003 10-12 Dec. 2003.

[44] Chu X, He Y, Chakrabarti K, Ganjam K. TEGRA: Table Extraction by Global Record Alignment. Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data; Melbourne, Victoria, Australia. 2723725: ACM; 2015. p. 1713-1728.

[45] Eberius J, Werner C, Thiele M, Braunschweig K, Dannecker L, Lehner W. DeExcelerator: a framework for extracting relational data from partially structured documents. Proceedings of the 22nd ACM international conference on Information & Knowledge Management; San Francisco, California, USA. 2508210: ACM; 2013. p. 2477-2480.

[46] Venetis P, Halevy A, Madhavan J, Paşca M, Shen W, Wu F, et al. Recovering semantics of tables on the web. Proc VLDB Endow. 2011;4(9):528-538.

[47] Cafarella MJ, Halevy A, Wang DZ, Wu E, Zhang Y. WebTables: exploring the power of tables on the web. Proc VLDB Endow. 2008;1(1):538-549.

[48] Balakrishnan S, Halevy AY, Harb B, Lee H, Madhavan J, Rostamizadeh A, et al., editors. Applying WebTables in Practice. CIDR; 2015.

[49] Cafarella MJ, Halevy AY, Zhang Y, Wang DZ, Wu E, editors. Uncovering the Relational Web. WebDB; 2008 Jun 13.

[50] Wang Y, Hu J, editors. A machine learning based approach for table detection on the web. Proceedings of the 11th international conference on World Wide Web; 2002 May; Honolulu, Hawaii, USA. 511478: ACM.

[51] Cafarella MJ, Halevy A, Khoussainova N. Data integration for the relational web. Proc VLDB Endow. 2009;2(1):1090-1101.

[52] Elmeleegy H, Madhavan J, Halevy A. Harvesting relational tables from lists on the web. Proc VLDB Endow. 2009;2(1):1078-1089.

[53] Wong YW, Widdows D, Lokovic T, Nigam K, editors. Scalable attribute-value extraction from semi-structured text. Data Mining Workshops, 2009 ICDMW'09 IEEE International Conference on; 2009 Dec 6: IEEE.

[54] Gonzalez H, Halevy A, Jensen CS, Langen A, Madhavan J, Shapley R, et al. Google fusion tables: data management, integration and collaboration in the cloud. Proceedings of the 1st ACM symposium on Cloud computing; Indianapolis, Indiana, USA. 1807158: ACM; 2010. p. 175-180.

[55] Lehmberg O, Ritze D, Meusel R, Bizer C. A Large Public Corpus of Web Tables containing Time and Context Metadata. Proceedings of the 25th International Conference Companion on World Wide Web; Montréal, Québec, Canada. 2889386: International World Wide Web Conferences Steering Committee; 2016. p. 75-76.

[56] Ritze D, Bizer C. Matching web tables to DBpedia-a feature utility study. context. 2017;42(41):19-31.

[57] Bhagavatula CS, Noraset T, Downey D. Methods for exploring and mining tables on Wikipedia. Proceedings of the ACM SIGKDD Workshop on Interactive Data Exploration and Analytics; Chicago, Illinois. 2501516: ACM; 2013. p. 18-26.

[58] Galkin M, Mouromtsev D, Auer S. Identifying Web Tables: Supporting a Neglected Type of Content on the Web. In: Klinov P, Mouromtsev D, editors. Knowledge Engineering and Semantic Web: 6th International Conference, KESW 2015, Moscow, Russia, September 30 - October 2, 2015, Proceedings. Cham: Springer International Publishing; 2015. p. 48-62.

[59] Wang Y, Phillips IT, Haralick RM. Table structure understanding and its performance evaluation. Pattern Recognition. 2004;37(7):1479-1497.

[60] Adelfio MD, Samet H. Schema extraction for tabular data on the web. Proc VLDB Endow. 2013;6(6):421-432.

[61] Kushmerick N. Wrapper induction: Efficiency and expressiveness. Artificial Intelligence. 2000;118(1):15-68.

[62] Kushmerick N, Weld DS, Doorenbos R, editors. Wrapper induction for information extraction. Proceedings of the International Joint Conference on Artificial Intelligence; 1997.

[63] Thamvijit D, Chanlekha H, Sirigayon C, Permpool T, Kawtrakul A, editors. Person information extraction from the web. 6th Symposium of Natural Language Processing; 2005.

[64] Cohen WW, Hurst M, Jensen LS. A flexible learning system for wrapping tables and lists in HTML documents. Proceedings of the 11th international conference on World Wide Web; Honolulu, Hawaii, USA. 511477: ACM; 2002. p. 232-241.

[65] Chen H-H, Tsai S-C, Tsai J-H, editors. Mining tables from large scale HTML texts. Proceedings of the 18th conference on Computational linguistics - Volume 1; 2000; Saarbrücken, Germany. 990845: Association for Computational Linguistics.

[66] Wu X, Cao C, Wang Y, Fu J, Wang S. Extracting Knowledge from Web Tables Based on DOM Tree Similarity. In: Lehner F, Fteimi N, editors. Knowledge Science, Engineering and Management: 9th International Conference, KSEM 2016, Passau, Germany, October 5-7, 2016, Proceedings. Cham: Springer International Publishing; 2016. p. 302-313.

[67] Augenstein I. Joint Information Extraction from the Web Using Linked Data. In: Mika P, Tudorache T, Bernstein A, Welty C, Knoblock C, Vrandečić D, et al., editors. The Semantic Web – ISWC 2014: 13th International Semantic Web Conference, Riva del Garda, Italy, October 19-23, 2014 Proceedings, Part II. Cham: Springer International Publishing; 2014. p. 505-512.

[68] Mintz M, Bills S, Snow R, Jurafsky D. Distant supervision for relation extraction without labeled data. Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2; Suntec, Singapore. 1690287: Association for Computational Linguistics; 2009. p. 1003-1011.

[69] Nadeau D, Sekine S. A survey of named entity recognition and classification. Lingvisticæ Investigationes. 2007;30(1):3-26.

[70] Gupta R, Sarawagi S. Answering table augmentation queries from unstructured lists on the web. Proc VLDB Endow. 2009;2(1):289-300.

[71] Hanifah HL, Akbar S. Table Extraction from Web Pages Using Conditional Random Fields to Extract Toponym Related Data. Journal of Physics: Conference Series. 2017;801(1):012064.

[72] Pinto D, McCallum A, Wei X, Croft WB. Table extraction using conditional random fields. Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval; Toronto, Canada. 860479: ACM; 2003. p. 235-242.

[73] Zhang X, Lv P, Zhao C, Wang J. A Method for Materials Knowledge Extraction from HTML Tables Based on Sibling Comparison. International Journal of Software Engineering and Knowledge Engineering. 2016;26(06):897-926.

[74] Zhang X, Liu X, Li X, Pan D. MMKG: An approach to generate metallic materials knowledge graph based on DBpedia and Wikipedia. Computer Physics Communications. 2017;211(Supplement C):98-112.

[75] Limaye G, Sarawagi S, Chakrabarti S. Annotating and searching web tables using entities, types and relationships. Proc VLDB Endow. 2010;3(1-2):1338-1347.

[76] Wang J, Wang H, Wang Z, Zhu KQ. Understanding Tables on the Web. In: Atzeni P, Cheung D, Ram S, editors. Conceptual Modeling: 31st International Conference ER 2012, Florence, Italy, October 15-18, 2012 Proceedings. Berlin, Heidelberg: Springer Berlin Heidelberg; 2012. p. 141-155.

[77] Wu W, Li H, Wang H, Zhu KQ. Probase: a probabilistic taxonomy for text understanding. Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data; Scottsdale, Arizona, USA. 2213891: ACM; 2012. p. 481-492.

[78] Lee T, Wang Z, Wang H, Hwang S-w. Web scale taxonomy cleansing. Proceedings of the VLDB Endowment. 2011;4(12):1295-1306.

[79] Liu X, Song Y, Liu S, Wang H, editors. Automatic taxonomy construction from keywords. Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining; 2012; Beijing, China. 2339754: ACM.

[80] He Y, Ganjam K, Chu X. SEMA-JOIN: joining semantically-related tables using big table corpora. Proc VLDB Endow. 2015;8(12):1358-1369.

[81] Wang Y, He Y. Synthesizing Mapping Relationships Using Table Corpus. Proceedings of the 2017 ACM International Conference on Management of Data; Chicago, Illinois, USA. 3064010: ACM; 2017. p. 1117-1132.

[82] Chakrabarti K, Chaudhuri S, Chen Z, Ganjam K, He Y, Redmond W. Data services leveraging Bing's data assets. Bulletin of the IEEE Computer Society Technical Committee on Data Engineering. 2016;39(3):15-28.

[83] Bhat S, Mustafi J, inventors; International Business Machines Corporation (Armonk, NY, US), assignee. Automatic data interpretation and answering analytical questions with tables and charts. United States2016.

[84] Etzioni O, Cafarella M, Downey D, Kok S, Popescu A-M, Shaked T, et al. Web-Scale Information Extraction in KnowItAll (Preliminary Results). Proceedings of the 13th international conference on World Wide Web; New York, NY, USA. 988687: ACM; 2004. p. 100-110.

[85] Etzioni O, Cafarella M, Downey D, Popescu A-M, Shaked T, Soderland S, et al. Unsupervised named-entity extraction from the Web: An experimental study. Artificial Intelligence. 2005;165(1):91-134.

[86] Yates A, Cafarella M, Banko M, Etzioni O, Broadhead M, Soderland S. TextRunner: open information extraction on the web. Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations; Rochester, New York. 1614177: Association for Computational Linguistics; 2007. p. 25-26.

[87] Banko M, Cafarella MJ, Soderland S, Broadhead M, Etzioni O, editors. Open Information Extraction from the Web. IJCAI; 2007 January.

[88] Bird S, Day D, Garofolo J, Henderson J, Laprun C, Liberman M. ATLAS: A flexible and extensible architecture for linguistic annotation. arXiv preprint cs/0007022. 2000.

[89] Cunningham H, Maynard D, Bontcheva K, Tablan V, editors. GATE: an Architecture for Development of Robust HLT Applications. Proc of the 40th Anniversary Meeting of the Association for Computational Linguistics; 2002.

[90] Cunningham H. GATE, a General Architecture for Text Engineering. Computers and the Humanities. 2002;36(2):223-254.

[91] Hellmann S, Lehmann J, Auer S, Brümmer M. Integrating NLP Using Linked Data. In: Alani H, Kagal L, Fokoue A, Groth P, Biemann C, Parreira JX, et al., editors. The Semantic Web – ISWC 2013: 12th International Semantic Web Conference, Sydney, NSW, Australia, October 21-25, 2013, Proceedings, Part II. Berlin, Heidelberg: Springer Berlin Heidelberg; 2013. p. 98-113.

[92] Krötzsch M, Vrandečić D, Völkel M. Semantic MediaWiki. In: Cruz I, Decker S, Allemang D, Preist C, Schwabe D, Mika P, et al., editors. 5th International Semantic Web Conference, ISWC 2006, November 5-9, 2006 Proceedings; 2006/01/01; Athens, GA, USA: Springer Berlin Heidelberg; 2006. p. 935-942.

[93] Krötzsch M, Vrandečić D, Völkel M, Haller H, Studer R. Semantic Wikipedia. Web Semantics: Science, Services and Agents on the World Wide Web. 2007;5(4):251-261.

[94] Völkel M, Krötzsch M, Vrandečić D, Haller H, Studer R. Semantic Wikipedia. Proceedings of the 15th international conference on World Wide Web; Edinburgh, Scotland. 1135863: ACM; 2006. p. 585-594.

[95] Shafranovich Y. Common format and MIME type for comma-separated values (CSV) files [Internet]. IETF; 2005. Available from: https://tools.ietf.org/html/rfc4180 [Accessed: 2019-12-20].

[96] Raggett D. HTML Tables [Internet]. IETF; 1996. Available from: https://tools.ietf.org/html/rfc1942 [Accessed: 2019-12-20].

[97] Berners-Lee T, Connolly D. Hypertext Markup Language - 2.0 [Internet]. IETF; 1995. Available from: https://tools.ietf.org/html/rfc1866 [Accessed: 2019-12-20].

[98] Nandakwang J, Chongstitvatana P. Extract semantic web knowledge from Wikipedia tables and lists. 8th International Conference on Knowledge and Smart Technology, KST 2016, Feb 3-6, 2016; 3 Feb. 2016; Chiangmai, Thailand: IEEE; 2016. p. 108-113.

[99] Abdallah SA, Ferris B. The Ordered List Ontology Specification [Internet]. 2010. Available from: http://purl.org/ontology/olo/core# [Accessed: 2019-12-20].

[100] Ciccarese P, Peroni S. The Collections Ontology: creating and handling collections in OWL 2 DL frameworks. Semantic Web. 2014;5(6):515-529.

[101] Leigh J, Wood D. An Ordered RDF List. W3C Workshop — RDF Next Steps; June 26-27, 2010; National Center for Biomedical Ontology (NCBO), Stanford, Palo Alto, CA, USA: W3C; 2010.

[102] Nandakwang J, Chongstitvatana P. TULIP: A Five-Star Table and List - from Machine-Readable to Machine-Understandable Systems. In: Okoye K, editor. Linked Open Data-Applications, Trends and Future Developments: IntechOpen; 2020.

[103] van den Bosch A, Bogers T, de Kunder M. Estimating search engine index size variability: a 9-year longitudinal study. Scientometrics. 2016;107(2):839-856.

[104] Han J, Kamber M, Pei J. Chapter 13 - Data Mining Trends and Research Frontiers. Data Mining (Third Edition). Boston: Morgan Kaufmann; 2012. p. 585-631.

[105] Han J, Kamber M. Chapter 10 - Mining Object, Spatial, Multimedia, Text, and Web Data. Data Mining (Second Edition). Boston: Morgan Kaufmann; 2006. p. 591-648.

[106] Hovy E, Navigli R, Ponzetto SP. Collaboratively built semi-structured content and Artificial Intelligence: The story so far. Artificial Intelligence. 2013;194(Supplement C):2-27.

[107] Berners-Lee T, Fielding R, Masinter L. Uniform Resource Identifier (URI): Generic Syntax (RFC 3986). IETF Network Working Group. 2005.

[108] Zanibbi R, Blostein D, Cordy JR. A survey of table recognition - Models, observations, transformations, and inferences. Document Analysis and Recognition. 2004;7(1):1-16.

[109] Embley DW, Hurst M, Lopresti D, Nagy G. Table-processing paradigms: a research survey. International Journal of Document Analysis and Recognition (IJDAR). 2006;8(2):66-86.

[110] Corrêa AS, Zander P-O. Unleashing Tabular Content to Open Data: A Survey on PDF Table Extraction Methods and Tools. Proceedings of the 18th Annual International Conference on Digital Government Research; Staten Island, NY, USA. 3085278: ACM; 2017. p. 54-63.

[111] Li Y, Gao J, Meng C, Li Q, Su L, Zhao B, et al. A Survey on Truth Discovery. SIGKDD Explor Newsl. 2016;17(2):1-16.

[112] Shvaiko P, Euzenat J. A Survey of Schema-Based Matching Approaches. In: Spaccapietra S, editor. Journal on Data Semantics IV. Berlin, Heidelberg: Springer Berlin Heidelberg; 2005. p. 146-171.

[113] Rahm E, Bernstein PA. A survey of approaches to automatic schema matching. The VLDB Journal. 2001;10(4):334-350.

[114] Saleem A. Semantic Web Vision : survey of ontology mapping systems and evaluation of progress [Thesis]2006.

[115] Choi N, Song I-Y, Han H. A survey on ontology mapping. SIGMOD Rec. 2006;35(3):34-41.

[116] Hurst M, editor Layout and language: Challenges for table understanding on the web. Proceedings of the International Workshop on Web Document Analysis; 2001 8 Sep.

บรรณานุกรม

ประวัติผู้เขียน

ชื่อ-สกุล	จุลเทพ นันทขว้าง
วัน เดือน ปี เกิด	18 กุมภาพันธ์ 2513
สถานที่เกิด	
วุฒิการศึกษา	วท.ม. (วิทยาศาสตร์คอมพิวเตอร์) ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย วท.บ. (วิทยาการคอมพิวเตอร์) (เกียรตินิยมอันดับหนึ่ง) ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยธรรมศาสตร์
ที่อยู่ปัจจุบัน	116/48 นาวง 21 สีกัน ดอนเมือง กรุงเทพฯ 10210
ผลงานตีพิมพ์	"TULIP: A Five-Star Table and List - from Machine-Readable to Machine-Understandable Systems," IntechOpen, DOI: 10.5772/intechopen.91406. May 6th 2020. Available from: https://www.intechopen.com/books/linked-open-data-applications-trends-and-future-developments/tulip-a-five-star-table-and-list-from-machine-readable-to-machine-understandable-systems "Extract semantic web knowledge from Wikipedia tables and lists," presented at the Knowledge and Smart Technology (KST), 2016 8th International Conference on, Feb. 3-6, 2016, Chiangmai, Thailand, 2016. "Mind Map Based Semantic Web Browser for Tablets," presented at the 4th International e-Learning Conference, IEC 2012, June 14-15, 2012, Nonthaburi, Thailand, 2012.