รายงานสหกิจศึกษาฉบับสมบูรณ์

เครื่องมือค้นหาในขอบเขตเฉพาะ
Domain Specified Search Engine

นายอรรถสิทธิ์ สินธัญญาธรรม

สาขาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2560

รายงานสหกิจศึกษาฉบับสมบูรณ์

เครื่องมือค้นหาในขอบเขตเฉพาะ

Domain Specified Search Engine

นายอรรถสิทธิ์ สินธัญญาธรรม

สาขาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2560

ชื่อโครงงานสหกิจศึกษา เครื่องมือค้นหาในขอบเขตเฉพาะ
ชื่อ-สกุล นักศึกษา นายอรรถสิทธิ์ สินธัญญาธรรม
คณะ วิศวกรรมศาสตร์ **ภาควิชา** วิศวกรรมคอมพิวเตอร์
ชื่อ-สกุล อาจารย์นิเทศ อ.บัณฑิต พัสยา และ อ.จิระศักดิ์ สิทธิกร
ชื่อ-สกุล ผู้นิเทศงาน นายวิภาส สุตันตยาวลี
สถานประกอบการ บริษัท แบ็คยาร์ด จำกัด ประเทศไทย

บทคัดย่อ

ผลิตภัณฑ์นี้มีมีวัตถุประสงค์เพื่อศึกษา และพัฒนาเครื่องมือค้นหาที่สามารถค้นหาข้อมูลเพื่อการตรวจสอบอาชญากรรมและสิ่งผิดกฎหมายภายในโซเชียลมีเดีย เพื่อที่จะทำให้เจ้าหน้าที่ตรวจสอบข้อมูลจากโซเชียลมีเดียได้อย่างสะดวกและรวดเร็วขึ้น

ผู้พัฒนาเลือกที่จะใช้ Scrapy เครื่องมือรวบรวมข้อมูลจากเว็บไซต์ต่างๆ เพื่อนำไปรวบรวมข้อมูลจากเว็บไซต์อย่าง Facebook และเครื่องมือค้นหาอย่าง Elasticsearch ซึ่งเป็นซอฟแวร์ที่เปิดให้ใช้อย่างอิสระที่ทางบริษัทที่ความชำนาญในการใช้งานอยู่แล้ว และเครื่องมือตัดคำภาษาไทย ICU ซึ่งเป็นผลิตภัณฑ์ของ IBM ที่ใช้จัดการข้อมูลประเภท Unicode โดย ICU จะตัดคำในรูปแบบคำภาษาไทยเป็นการตัดคำให้ความยาวของตัวอักษรสั้นที่สุดและยังมีความหมายอยู่ และสุดท้ายเครื่องมือการแปลงภาษาไทยเป็นสัทอักษรสากล Epitran ที่พัฒนาโดย David R. Mortensen ซึ่งเป็นเครื่องมือที่ออกใบอนุญาตเป็น MIT License เพื่อใช้ในการค้นหาคำพ้องเสียงในภาษาไทย

โดยผลิตภัณฑ์นี้สามารถทำตามความต้องการของระบบที่ลูกค้าต้องการได้ทุกประการ

คำสำคัญ: เครื่องมือค้นหา, Elasticsearch, Unicode, ICU, Docker, Web Crawler, Inverted Index, full text search, สัทอักษรสากล, คำไวพจน์

Co-operative Title: Domain Specified Search Engine

Student Intern Name: Mr. Atthasit Sintunyatum

Faculty: Engineering **Department:** Computer Engineering

Advisor Name: Mr.Bundit Pasaya and Mr.Jirasak Sittigorn

Mentor Name: Mr.Vipas Sutantayawalee

Company: Backyard Co, Ltd.

ABSTRACT

The objective of this product is to study and develop the search engine. The search engine can search data from the social media for crime verification and illegal action on the social media platform. It helps officer to verify data from social media faster and easier than without the tool.

My team chose Scrapy as a web crawler framework for collecting data from the social media such as Facebook. Elasticsearch is an open source search engine software which my company has expertise in utilize it. ICU by IBM for character segmentation and Epitran as a tool for transliterating orthographic text as IPA (International Phonetic Alphabet) develop by David R. Mortensen to search homophone words.

Finally, this product can be implemented all requirements from the customer.

Keywords: Search Engine, Elasticsearch, Unicode, ICU, Docker, Web Crawler, Inverted Index, Full Text Search, Phonetic Alphabet, Synonym

กิตติกรรมประกาศ

ปริญญานิพนธ์นี้เสร็จสมบูรณ์ได้ด้วยความช่วยเหลือจากหลายท่านทั้งทางตรงและทางอ้อม ซึ่งจะสำเร็จลงไม่ได้หากปราศจากความช่วยเหลือของบุคคลเหล่านี้

ขอขอบคุณ อาจารย์ผู้นิเทศงาน อาจารย์บัณฑิต พัสยา ซึ่งเป็นผู้ที่มานิเทศงาน และช่วยให้คำแนะนำในการทำงาน การแก้ไขปัญหา และจุดบกพร่องของโครงงาน ซึ่งทำให้โครงงานสมบูรณ์มากยิ่งขึ้น

ขอขอบคุณ อาจารย์จิระศักดิ์ สิทธิกร ซึ่งเป็นผู้ที่คอยดูแล และให้คำแนะนำในด้านของงานสหกิจศึกษา ซึ่งทำให้การทำโครงงานสะดวกมากยิ่งขึ้น

ขอขอบคุณ นายวิภาส สุตันตยาวลี หัวหน้างานที่ให้คำปรึกษา แนะนำหลักการในการทำงาน และรูปแบบของงานที่เป็นมาตรฐานตามหลักการที่ถูกต้องอย่างสม่ำเสมอ ซึ่งทำให้ตัวผลิตภัณฑ์ชิ้นนี้สำเร็จลุล่วงอย่างมีคุณภาพ

ขอขอบคุณ นายกฤต นิรันดรกุลชัย ผู้ดูแลและคอยให้คำปรึกษา ความรู้ทางด้านการเขียนโปรแกรม ทำให้ผลิตภัณฑ์ชิ้นนี้สำเร็จไปได้ด้วยดี

สุดท้ายนี้ขอขอบคุณ บิดา มารดา ครอบครัว เพื่อนๆ ตลอดจนผู้เกี่ยวข้องที่ไม่ได้กล่าวนามทุกท่าน ที่เป็นกำลังใจ ให้การสนับสนุน และความช่วยเหลือในการทำโครงงานครั้งนี้จนสำเร็จลุล่วงไปได้

อรรถสิทธิ์ สินธัญญาธรรม

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

ตารางที่ **หน้า**

สารบัญภาพ

รูปที่ **หน้า**

บทที่ 1
บทนำ

1.1 ความเป็นมา และความสำคัญ

Google หลายคนรู้จัก Google ในนาม Search Engine ที่มีคนใช้มากที่สุดในโลก โดยสิ่งที่ Search Engine ของ Google ทำคือรวมรวบข้อมูลต่างๆ จากเว็บไซต์ส่วนใหญ่บนโลกนี้ และให้ผู้ใช้ค้นหาผ่านการพิมพ์ keyword ในช่องค้นหาและกดปุ่มค้นหา แล้ว Google จะโชว์ผลลัพธ์เป็นเว็บไซต์ที่เกี่ยวข้องกับ keyword ที่เราต้องการค้นหาออกมาทั้งหมด ถึงแม้จะเป็น Search Engine ที่ทรงพลังแต่ก็ยังมีข้อจำกัดบางอย่างที่ Google ทำไม่ได้ เช่น ข้อมูลที่องค์กรไม่ต้องการเปิดเผย ข้อมูลจาก API ต่างๆ ข้อมูลที่ต้องทำล็อคอินก่อนถึงจะมีสิทธิ์เข้าถึงข้อมูล เป็นต้น จากข้อจำกัดเหล่านี้จึงเกิดเป็นผลิตภัณฑ์ของบริษัทขึ้นคือ “เครื่องมือค้นหาในขอบเขตเฉพาะ”

1.2 วัตถุประสงค์ของงาน

1) เพื่อค้นหาข้อความในขอบเขตเฉพาะ
2) เพื่อศึกษาการประมวลผลภาษาไทยสำหรับการค้นหาข้อมูล

1.3 ขอบเขตของการทำงาน

เป็นเครื่องมือการค้นหาเพื่อให้ผู้หาค้นพบสิ่งที่ต้องการค้นหาได้ง่ายขึ้น โดยระบบแบ่งออกเป็น 3 ส่วนหลักๆ ดังนี้

1) เครื่องมือรวบรวมข้อมูล (Web crawling tool)
2) เครื่องมือค้นหา (Search engine)
3) เว็บไซต์แสดงผลการค้นหา (Website)

1.4 วิธีดำเนินการทำงาน

1) รับทราบความต้องการของระบบจากลูกค้า

2) ศึกษาเครื่องมือต่างๆ ของระบบ

3) ชี้แจง/ปรับแต่ง ความต้องการของระบบ

4) ออกแบบระบบ

5) เขียนโปรแกรม

6) ทดสอบระบบ

7) สาธิตระบบแก่ลูกค้า

8) จัดทำรายงาน

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1) ผู้ใช้สามารถค้นพบสิ่งที่ตนต้องการค้นหาได้ง่ายขึ้น

2) ตรงตามความต้องการของระบบของลูกค้า

บทที่ 2
แนวคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

1) Docker Container

เป็นการจำลองสภาพแวดล้อมสำหรับการเปิดบางบริการโดยเฉพาะ ทำให้ใช้ทรัพยากรในการน้อยแต่ละบริการน้อย และสามารถเปิดบริการได้ในคอมพิวเตอร์เครื่องในก็ได้ที่รับรองบริการ docker [11]

2) Web Crawler

ก่อนที่เราจะสร้างเครื่องมือการค้นหานั้น เราจำเป็นต้องมีเอกสารที่ต้องการค้นหาก่อน ซึ่งการรวบรวมเอกสารจากเว็บไซด์ต่างๆ นั้นจะถูกเรียกว่า web crawler ซึ่งมีวิธีการหลักๆ ก็คือการเข้าถึงเว็บไซด์ใดๆ เว็บไซด์หนึ่ง เก็บรวบรวมข้อมูลในเว็บไซด์นั้น และไปยัง url อื่นที่ไม่เคยไปที่อยู่ในเว็บไซด์นั้นไปเรื่อยๆ จนกระทั้งครบทุก url [6]

3) เครื่องมือค้นหา (Search Engine)

เป็นเครื่องมือสำหรับใช้ค้นหาข้อมูลในเอกสารที่มีอยู่

4) Inverted Index

เป็นโครงสร้างข้อมูลสำหรับกระบวนการค้นหา โดยจะนำเอกสารมาย่อยเป็นคำและสร้างรายการที่ถูกเรียงลำดับของคำที่แตกต่างกัน และจดจำว่าคำแต่ละคำมาจากเอกสารใดบ้าง [7]

อย่างเช่นตัวอย่าง

1. The man who fell the sky
2. The lazy man is falling in love

จากประโยคข้างต้นจะได้ผลลัพธ์ของ inverted index ดังนี้

ตารางที่ 2.1 ตัวอย่างของ inverted index

Term	Doc 1	Doc 2
boy		X
falling		X
fell	X	
in		X
is		X
lazy		X
love		X
man	X	X
Sky	X	
The	X	X
who	X	

เราจะเห็นปัญหาจาก inverted index ดังนี้

- คำที่เหมือนกัน เช่น The, the
- คำที่พบเห็นได้ทั่วไป (Stop words) เช่น the
- คำที่มีความหมายเหมือนกัน เช่น fell, falling

จากปัญหาดังกล่าวจะมีผลกระทบต่อความยากง่ายของการค้นหาของผู้ใช้ จึงจำเป็นต้องมีกระบวนการก่อนที่จะสร้าง inverted index ซึ่งกระบวนการนี้ แต่ละภาษาก็จะมีกระบวนการที่แตกต่างกัน

5) การตัดคำภาษาไทย (Thai Character Segmentation)

เนื่องด้วย Invert Indexes เป็นวิธีการจัดเก็บข้อมูลเพื่อค้นหาโดยนำเอกสารมาย่อยเป็นคำแต่เนื่องด้วยลักษณะของภาษาไทยเป็นภาษาที่ไม่มีตัวอักษรแบ่งคำชัดเจนจึงทำให้จำเป็นต้องกระบวนการในการตัดคำในภาษาไทย

6) คำไวพจน์ (Synonyms)

คำไวพจน์ พจนานุกรมฉบับราชบัณฑิตยสถาน พุทธศักราช ๒๕๔๒ ให้คำอธิบายว่า "คำที่เขียนต่างกันแต่มีความหมายเหมือนกันหรือใกล้เคียงกันมาก เช่น มนุษย์ กับ คน, บ้าน กับ เรือน, รอ กับ คอย, ป่า กับ ดง, คำพ้องความ ก็ว่า" (พจนานุกรม ๒๕๔๒ หน้า ๑๐๙๑) [12]

7) สัทอักษรสากล (International Phonetic Alphabet)

คือสัทอักษรชุดหนึ่งที่พัฒนาโดยสมาคมสัทศาสตร์สากล โดยมุ่งหมายให้เป็นสัญกรณ์มาตรฐานสำหรับการแทนเสียงพูดในทุกภาษา นักภาษาศาสตร์ใช้สัทอักษรสากลเพื่อแทนหน่วยเสียงต่างๆ ที่อวัยวะออกเสียงของมนุษย์สามารถเปล่งเสียงได้ [13]

2.2 งานวิจัยที่เกี่ยวข้อง

1) Character Cluster Based Thai Information Retrieval (TCC) [10]

เป็นบทความวิจัยเพื่อแก้ 3 ปัญหาหลักๆ ของการค้นหาคำภาษาไทยดังนี้

1) ปัญหาการไม่ค้นพบเอกสารเมื่อค้นหาด้วยคำว่า A ซึ่งเป็นคำย่อยของคำ B ซึ่ง

$$A = \beta$$

$$B = \alpha\beta\gamma$$

เมื่อ α, β และ γ เป็นคำย่อยของ B

2) ปัญหาการค้นพบเอกสารเมื่อค้นหาด้วยคำว่า A ซึ่งเป็นคำย่อยของคำ B ซึ่ง

$$A = \beta$$

$$B = \alpha\beta\gamma$$

เมื่อ α, β และ γ เป็นคำย่อยของ B และ $\alpha\beta$ หรือ $\beta\gamma$ เป็นคำที่ไม่สามารถแยกกันได้

3) ปัญหาการค้นพบเอกสารเมื่อค้นหาด้วยคำว่า A ซึ่งเป็นคำย่อยของคำ B ซึ่ง

$$A = \beta$$

$$B = \alpha\beta\gamma$$

เมื่อ α, β และ γ เป็นคำย่อยของคำ B และ $\alpha\beta\gamma$ เป็นคำที่ไม่สามารถแยกกันได้

ปัญหาที่ 1) สามารถแก้ได้โดย full-text search อยู่แล้วแต่ TCC สามารถแก้ปัญหาทั้ง 3 ได้เพราะ TCC จะช่วยทำให้มั่นใจได้ว่าคำค้นหา A จะไม่เป็นคำย่อยในคำ B

2) International Components for Unicode (ICU) [5]

เป็นผลิตภัณฑ์ของ IBM ที่ใช้จัดการกับ Unicode อีกทั้งยังมีฟังก์ชันในการตัดคำภาษาไทย

3) Epitran [3]

เป็นโมดูลในภาษา python สำหรับแปลงภาษาต่างๆ ให้เป็นสัทอักษรโดยออกใบอนุญาตเป็น MIT License พัฒนาขึ้นโดย David R. Mortensen นักวิจัยและพัฒนาเกี่ยวกับภาษาศาสตร์

บทที่ 3
วิธีดำเนินงาน

3.1 วิธีดำเนินการทำงาน

1) รับทราบความต้องการของระบบจากลูกค้า

จากการพูดคุยกับลูกค้า พบว่าลูกค้ามีความต้องการระบบค้นหาข้อความจากเพจต่างๆ ที่ กำหนดให้จาก facebook และระบบค้นหามี ฟังก์ชันต่างๆ ตามที่กำหนดดังนี้

1) สามารถค้นหาด้วย AND OR NOT ได้

2) สามารถค้นหาด้วยคำพ้องเสียงได้

3) สามารถค้นด้วยคำที่ใกล้เคียงได้

4) สามารถค้นคำซ้ำจากการค้นหา

2) ศึกษาเครื่องมือต่างๆ ของระบบ

จากความต้องการของระบบ พบว่าในปัจจุบันทางบริษัทมีโปรแกรมที่สามารถดึงข้อมูล จากเพจต่างๆ ที่กำหนดให้จาก facebook ผ่านทาง Scrapy framework อยู่แล้ว และจากการศึกษา Search Engine ต่างๆ พบว่า Elasticsearch สามารถทำตามความต้องการของระบบได้ 3 ข้อดังนี้

1) สามารถค้นหาด้วย AND OR NOT ได้

2) สามารถค้นด้วยคำที่ใกล้เคียงได้

4) สามารถค้นคำซ้ำจากการค้นหา

และทางบริษัทเองมีความชำนาญในการใช้ Elasticsearch อยู่แล้ว และฟังก์ชันข้อ 2) สามารถค้นหาด้วยคำพ้องเสียงได้ ทางผู้จัดทำได้ใช้ Epitran ซึ่งเป็นโมดูลในภาษา python ที่สามารถแปลงคำภาษาไทยให้เป็นสัทอักษรได้

3) ชี้แจงความต้องการของระบบ

จากการศึกษาเครื่องมือต่างๆ ของระบบ จะพบว่าสามารถทำตามความต้องการของระบบได้ จึงนำเครื่องมือต่างๆ ของระบบไปเสนอลูกค้า

4) ออกแบบระบบ

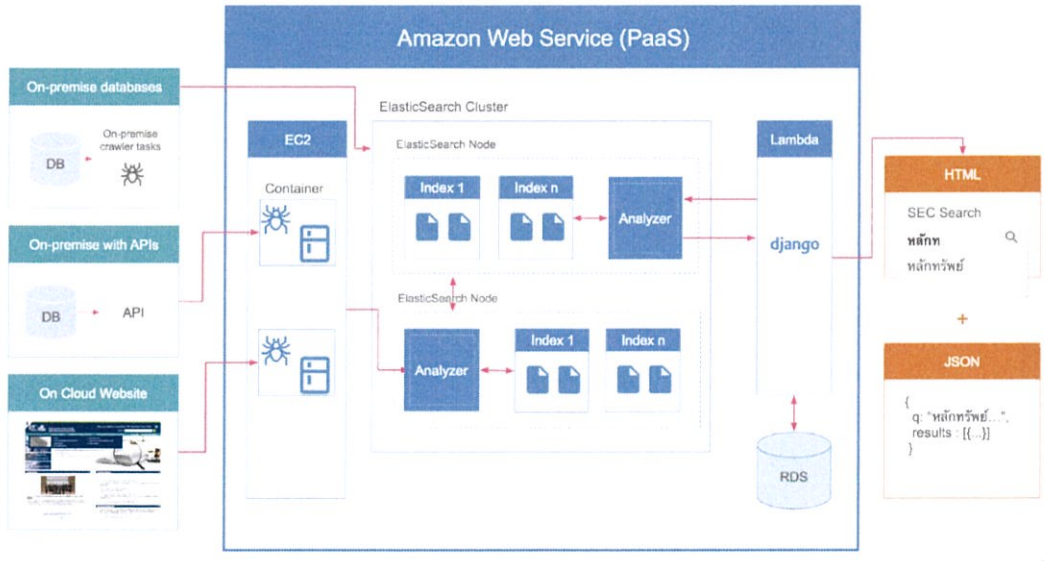
ระบบจะแบ่งออกเป็น 3 บริการ หลักๆ ดังนี้

1) Web crawler

2) Elasticsearch

3) Website & API

และทุกบริการ จะให้บริการผ่านบน docker container เพื่อง่ายต่อการพัฒนาเป็นทีม และการขึ้นเป็นผลิตภัณฑ์



รูปที่ 3.1 Infrastructure ของระบบ

5) เขียนโปรแกรม

จากการออกแบบระบบ จึงต้องเขียนแบ่งขั้นตอนการเขียนโปรแกรมออกเป็น 3 ส่วนหลักๆ ดังนี้

1) การพัฒนาโปรแกรมดึงข้อมูลจากเพจต่างๆบน facebook เพื่อให้สามารถส่งข้อมูล เข้าสู่ elasticsearch ได้

2) บริการที่ใช้ติดต่อกับ elasticsearch เพื่อส่งข้อมูลไปยังเว็บไซต์

3) เว็บไซต์ในการแสดงผลการค้นหา

6) ทดสอบระบบ

การทดสอบเครื่องมือค้นหาจำเป็นต้องใช้ผลการค้นหาของผู้ใช้เพื่อพัฒนาผลลัพธ์การ ค้นหาให้ดียิ่งขึ้น ซึ่งทางผู้จัดทำยังไม่ได้เปิดให้บริการนี้ ทางผู้จัดทำจึงทดลองค้นหาผ่านเครื่องมือ ค้นหาเองเพื่อให้ตรงตามความต้องการของระบบของลูกค้าเบื้องต้น

7) สาธิตระบบแก่ลูกค้า

การสาธิตระบบแก่ลูกค้า สามารถทำตามความต้องการของระบบได้ถูกต้องครบถ้วนและ ลูกค้ามีความพึงพอใจ

3.2 แผนการดำเนินงาน

ตารางที่ 3.1 แผนการดำเนินงาน

ลำดับ	หัวข้องาน	เดือนที่ 1				เดือนที่ 2				เดือนที่ 3				เดือนที่ 4			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	รับความต้องการของระบบจากลูกค้า																
2	ศึกษาเครื่องมือต่างๆ ของระบบ																
3	ชี้แจงความต้องการของระบบ																
4	ออกแบบระบบ																
5	เขียนโปรแกรม																
6	ทดสอบระบบ																
7	สาธิตระบบแก่ลูกค้า																
8	จัดทำรายงาน																

บทที่ 4
ผลการทำงาน

4.1 การทดสอบการค้นหาด้วย AND OR NOT

การค้นหาด้วย AND OR NOT สามารถทำได้โดยการใช้การ query แบบ querystring ของ elasticsearch โดยผลลัพธ์จะได้ดังนี้

A) เมื่อค้นหาด้วยคำว่า กาฬ จะได้ผลลัพธ์ที่มีคำว่า กาฬ ทั้งหมด 18 เอกสาร

B) เมื่อค้นหาด้วยคำว่า กาฬ AND นาน จะได้ผลลัพธ์ที่มีคำว่า กาฬ และคำว่า นาน ทั้งหมด 2 เอกสาร

C) เมื่อค้นหาด้วยคำว่า กาฬ NOT นาน จะได้ผลลัพธ์ที่มาคำว่า กาฬ แต่ไม่มีคำว่า นาน ทั้งหมด 16 เอกสาร

D) เมื่อค้นหาด้วยคำว่า นาน NOT กาฬ จะได้ผลลัพธ์ที่มีคำว่า นาน แต่ไม่มีคำว่า กาฬ ทั้งหมด 3,550 เอกสาร

E) เมื่อค้นหาด้วยคำว่า กาฬ OR นาน จะได้ผลลัพธ์ที่มีคำว่า กาฬ หรือคำว่า นาน อย่างใดอย่างหนึ่ง ทั้งหมด 3,568 เอกสาร

จากผลลัพธ์ด้านบนสรุปได้ว่า

$$C = \alpha - (\beta \cap \alpha)$$
$$B = \beta \cap \alpha$$
$$A = \alpha$$
$$\therefore C = A - B \rightarrow 16 = 18 - 12 \rightarrow 16 = 16$$

และจาก

$$\alpha \, or \, \beta = \alpha + \beta - (\beta \cap \alpha)$$
$$E = A + D$$
$$\therefore 3{,}568 = 18 + 3{,}550 \rightarrow 3{,}568 = 3{,}568$$

ดังนั้นจึงสรุปได้ว่าฟังก์ชัน AND OR NOT สามารถทำได้ถูกต้อง

4.2 การทดสอบการค้นหาด้วยฟังก์ชันคำพ้องเสียง

เมื่อค้นหาคำทางโปรแกรมจากตัดคำเหล่านั้นและแปลงเป็น phonetic alphabet เพื่อนำไปค้นหาใน phonetic field ดังนั้นเมื่อค้นหาคำว่า กาฬ และเลือกใช้แสดงผลลัพธ์คำพ้องเสียงด้วย ผลลัพธ์การค้นหาพบว่ามีคำว่า การ กาล กานต์ อยู่ด้วย

สรุปได้ว่าฟังก์ชันคำพ้องเสียงสามารถทำได้ถูกต้อง

4.3 การทดสอบการค้นหาด้วยฟังก์ชันคำใกล้เคียง

การค้นหาคำใกล้เคียงสามารถทำได้โดยใส่พารามิเตอร์ fuzzy ให้แก่ elasticsearch และตั้งค่าว่าสามารถแตกต่างได้ทั้งหมดกี่ตัวอักษร ในผลิตภัณฑ์นี้เลือกให้สามารถแตกต่างกันได้ 2 ตัวอักษร โดย fuzzy จะคิดคะแนนความแตกต่างตาม levenshtein distance algorithm

เมื่อค้นหาคำว่า กนข้าว และเลือกใช้แสดงผลลัพธ์คำใกล้เคียงด้วย ผลลัพธ์การค้นหาพบว่ามีคำว่า กินข้าว กับข้าว อยู่ด้วย

สรุปได้ว่าฟังก์ชันคำใกล้เคียงสามารถทำได้ถูกต้อง

4.4 การทดสอบการค้นหาด้วยฟังก์ชันค้นหาซ้ำจากการค้นหา

ฟังก์ชันค้นหาซ้ำจากการค้นหาคือการใช้การ AND ในการ query แบบ querystring ของ elasticsearch แต่เพื่อให้ง่ายต่อการใช้งานจึงเพิ่งช่องการค้นหาขึ้นและใส่ keyword ที่ต้องการค้นหาซ้ำในช่องนั้นแทน เมื่อค้นหาคำว่า กาฬ และเพิ่มคำว่า นาน ไปในฟังก์ชันค้นหาซ้ำจากการค้นหา จะได้ผลลัพธ์ทั้งหมด 2 เอกสาร ซึ่งเหมือนกับผลลัพธ์ของ กาฬ AND นาน

สรุปได้ว่าฟังก์ชันค้นหาซ้ำจากการค้นหาสามารถทำได้ถูกต้อง

4.5 การทดสอบผลลัพธ์ในช่อง “หรือคุณหมายถึง”

เป็นการทดสอบผลลัพธ์ที่ได้จากการนำ keyword ที่ผู้ใช้ค้นหามาใช้วิธีการ term suggestion ใน elasticsearch เพื่อช่วยในการแนะนำคำศัพท์ที่ถูกต้องให้แก่ผู้ใช้ โดยผลลัพธ์ของการทำ term suggestion สามารถแนะนำคำศัพท์ที่ถูกต้องเมื่อพิมพ์ keyword ผิดบางตัวอักษรได้ในระดับหนึ่ง

บทที่ 5
สรุปผลการทำงาน และข้อเสนอแนะ

5.1 บทสรุป

สามารถทำตามความต้องการของระบบของลูกค้าได้อย่างครบถ้วน แต่การปรับปรุงผลการค้นหาทำได้ยากเนื่องจากผลิตภัณฑ์ยังไม่ได้เปิดให้บริการจริง จึงอาจจะมีปัญหาการค้นหาแล้วได้ผลลัพธ์ไม่ตรงตามที่ต้องการในบาง keyword

5.2 ปัญหาอุปสรรค และแนวทางแก้ไข

1) การตัดคำในภาษาไทยนั้นยังไม่มีเครื่องมือตัวไหนที่สามารถตัดคำภาษาไทยได้ถูกต้อง 100% ทำให้มีโอการเกิดการผิดพลาดในผลการค้นหาได้ แนวทางการแก้ไขคือ เลือกเครื่องมือการตัดคำที่เหมาะสมกับข้อความในเอกสารให้มากที่สุด

2) เนื่องจากยังไม่มีการใช้งานจริงเกิดขึ้น ทำให้การปรังปรุงผลิตภัณฑ์โดยดูจากประวัติการค้นหาของผู้ใช้ทำไม่ได้ แนวทางการแก้ไขคือ ผู้พัฒนาระบบทดลองการค้นหาด้วยตนเองและปรับปรุงผลการค้นหาเองระดับหนึ่ง

3) การใช้ฟังก์ชันคำพ้องเสียง ผลลัพธ์ไม่สามารถ highlight คำที่เป็น keyword ได้ดังนั้นจึงอาจทำให้เข้าใจผิดว่าได้ผลลัพธ์ที่ผิดพลาดได้ แนวทางการแก้ไขคือ ศึกษาระบบการ highlight ตามตำแหน่งของตัวอักษรของ elasticsearch และนำมาพัฒนาผลิตภัณฑ์

4) ปัญหาที่เกิดขึ้นใน API แนะนำคำค้นหาคือ เกิดจากข้อมูลในเอกสารเยอะเกินไปทำให้เกิดข้อจำกัดเรื่องหน่วยความจำทำให้ไม่สามารถใช้งาน API นี้ได้ แนวทางการแก้ไขคือ การเปลี่ยนคำแนะนำการค้นหาจากเดิมเป็นการใช้คำแนะนำจากเอกสารที่มีอยู่เป็นการแนะนำจากประวัติการค้นหาของผู้ใช้แทน

5.3 แนวทางการพัฒนาต่อ

1) การใช้ประวัติการค้นหาของผู้ใช้มาพัฒนาผลิตภัณต์ต่อ

2) การเพิ่มคำไวพจน์เพื่อให้ผู้ใช้งานได้ผลลัพธ์ตรงตามที่ต้องการมากขึ้น

บรรณานุกรม

[1] Alex Brasetvik. (2013). *Elasticsearch from the Bottom up, Part1*. Retrieved December 2, 2017, from https://www.elastic.co/blog/found-elasticsearch-from-the-bottom-up

[2] Alex Brasetvik. (2014). *Elasticsearch from the Top Down*. Retrieved December 2, 2017, from https://www.elastic.co/blog/found-elasticsearch-top-down

[3] David Mortensen. *Epitran A tool for transcribing orthographic text as IPA.* Retrieved September 4, 2017, from https://github.com/dmort27/epitran

[4] Doug Turnbull, John Berryman. *Relevant Search: With applications for Solr and Elasticsearch.* Manning Publications, 2016

[5] *ICU - International Components for Unicode.* Retrieved December 2, 2017, from http://site.icu-project.org

[6] *Inverted Indexing for Text Retrieval.* Retrieved December 2, 2017, from http://www.dcs.bbk.ac.uk/~dell/teaching/cc/book/ditp/ditp_ch4.pdf

[7] *Inverted Index.* Retrieved December 2, 2017, from https://www.elastic.co/guide/en/elasticsearch/guide/current/inverted-index.html

[8] Matthew Lee, Hinman Radu, Gheorghe and Roy Russo. *Elasticsearch in Action.* Manning Publications, 2015

[9] Otis Gospodnetic, Erik Hatcher, Michael McCandless. *Lucene in Action.* 2nd ed, Manning Publications, 2010

[10] T. Theeramunkong, V. Sornlertlamvanich, T. Tanhermhong, W. Chinnan. *Character Cluster Based Thai Information Retrieval.* IRAL '00 Proceedings of the fifth international workshop on Information retrieval with Asian languages, 2000.

[11] *WHAT IS A CONTAINER.* Retrieved December 2, 2017, from https://www.docker.com/what-container

[12] วันทนา มาศวรรณา. (2013). *คำไวพจน์.* Retrieved December 2, 2017, from https://www.gotoknow.org/posts/517687

[13] *สัทอักษรสากล*. Retrieved December 2, 2017, from https://th.wikipedia.org/wiki/สัทอักษรสากล