การเพิ่มประสิทธิภาพแบบจำลองหัวข้อด้วยสภาพแวดล้อมแบบข้อมูลขนาดใหญ่

Performance Improving Topic Modeling with Big Data Environment

ธนกร ญานกาย* และ วนิดา แก่นอากาศ

Thanakorn Yarnguy* and Wanida Kanarkard

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยขอนแก่น

Department of Computer Engineering, Faculty of Engineering, Khonkaen University

*E-mail: thanakorn.y@kkumail.com

Received: Nov 21, 2018

Revised: Jun 15, 2019

Accepted: Oct 30, 2019

บทคัดย่อ

การทำเหมืองข้อมูลเป็นวิธีการหนึ่ง ที่ใช้ในการค้นหาองค์ความรู้ในข้อมูล ข้อมูลประเภทข้อความเป็นข้อมูลประเภทที่สามารถค้นหาองค์ความรู้ได้หลากหลายรูปแบบ เช่น การสรุปข้อความ การหาความหมายแฝง การหาหัวข้อ การจัดกลุ่มข้อความ Latent Dirichlet Allocation (LDA) เป็นเทคนิคหนึ่งที่ใช้ในการค้นหา หัวข้อ(topic) ของข้อมูล และสามารถทำการเพิ่มประสิทธิภาพได้ด้วยการปรับปรุงค่าโดยใช้ optimization อัลกอริทึ่ม ซึ่งผู้วิจัยใช้ Ant colony optimization ในการปรับค่าตัวแปร ซึงการค้นหาหัวข้อจากเอกสาร มักจะใช้เวลาในการคำนวณค่านาน ผู้วิจัยจึงประยุกต์ใช้ map-reduce ซึ่งเป็นการทำงานภายใต้สภาพแวดล้อมของ Hadoop มาช่วยในการประมวลผลเพื่อให้สามารถทำงานได้เร็วขึ้นและทำการวัดค่าประสิทธิภาพของอัลกอริทึ่ม LDA ผลการวิจัยพบว่า การประมวลผลชุดข้อมูลด้วยอัลกอริทึ่ม LDA ที่ปรับปรุงค่าตัวแปรโดย ACO ที่ทำงานโดย Map-reduce มีความเร็วในการประมวลผลที่สูงขึ้น

คำสำคัญ: แม็พรีดิวซ์ อัลกอริทึ่มอาณาจักรมด การทำเหมืองข้อความ Latent Dirichlet Allocation.

Abstract

Data mining is a method which is used to find knowledge in data. There are many techniques to find the knowledge in text data such as document summation, latent meaning, document topics, and document clustering. Latent Dirichlet Allocation (LDA) is an algorithm used to find hidden topics of the document, it can improve performance b parameters tuning. We use Ant colony optimization (ACO) to optimize LDA parameters.it takes a long time to calculate the topic from many documents. In this work, we apply a map-reduce programming technique which works under the Hadoop environment to accurately calculate time. The results have shown that processing documents with LDA with optimizing parameters by ACO under Hadoop environment is obviously faster with much improved performance compared to the one without map-reduce.

Keywords: Map-reduce, Ant colony optimization, Text mining, Latent Dirichlet Allocation.

1. บทนำ

การศึกษาเกี่ยวกับการค้นหาองค์ความรู้ในข้อมูลคือการทำเหมืองข้อมูล (Data mining) เป็นการค้นหาองค์ความรู้จากข้อความปริมาณมากให้เหลือเพียงส่วนสำคัญหรือองค์ความรู้ที่ต้องการ เนื่องจากข้อมูลมีหลากหลายประเภท ทั้งรูปแบบ ข้อความ รูปภาพ วีดีโอ การประมวลผลข้อมูลจึงมีการจำแนกย่อยลงไปอีก ในการประมวลผลข้อมูลประเภทข้อความจึงถูกเรียกว่า text mining ซึ่งเป็นกระบวนการสร้าง

องค์ความรู้จากข้อมูลประเภทข้อความ (text) ในกระบวนการนี้ต้องใช้เทคนิคอื่นๆ เข้าช่วย เช่น การประมวลผลภาษาธรรมชาติ (Natural language processing) ในการประมวลผลข้อมูลประเภทข้อความจำนวนมาก เทคนิคหนึ่งที่นิยมคือ การค้นหาคำสำคัญโดยการใช้อัลกอริทึ่มประเภท topic modeling

LDA (Latent Dirichlet Allocation) [1] เป็นแบบจำลองความน่าจะเป็นสำหรับข้อมูลที่ไม่ต่อเนื่อง เช่น คลังข้อมูล (corpus) แบบจำลองนี้ถูกสร้างขึ้นโดยมีแนวคิดว่า ในเอกสาร (document) จะประกอบด้วย หัวข้อ (topic) รวมกันอยู่แบบสุ่ม ซึ่งหัวข้อเหล่านี้ได้กระจัดกระจาย อยู่ในกลุ่มของคำศัพท์(word) ของเอกสารนั้นๆ LDA ได้ถูกนำมาใช้เพื่อหาหัวข้อของข้อมูล อัลกอริทึ่มในการทำ text mining สามารถเพิ่มประสิทธิภาพได้ โดยการปรับปรุงค่าตัวแปรที่เหมาะสม LDA มีตัวแปรที่ใช้ในการกำหนดค่า คือ α และ β ซึ่งการลือกค่าที่เหมาะสมจะทำให้ได้รับผลการทดลองที่ดีขึ้น กระบวนการในการค้นหาค่าที่เหมาะสมคือการใช้ optimization algorithm ซึ่ง Ant colony optimization [2] เป็นอัลกอริทึ่มหนึ่งที่นิยมใช้ในการ optimization โดย ACO เป็นการเลียนแบบพฤติกรรมในธรรมชาติ ของมด เพื่อหาเส้นทางในการเดินทางไปยังแหล่งอาหารและในการเดินทางมีการปล่อยฟีโรโมนไปตามเส้นทางที่เดินไป เมื่อมดตัวอื่นๆ เดินทางมาพบ ฟีโรโมนที่ปล่อยไว้ก็จะหยุดการสุ่มหาเส้นทาง และเดินทางตามเส้นทางที่มีฟีโรโมน และสามารถเดินทางระหว่างแหล่งอาหารและรัง ได้อย่างถูกต้อง และสามารถนำมาประยุกต์ในงานด้าน text mining ได้ เช่น การใช้ ACO ในการปรับค่าพารามิเตอร์ของ LDA [3] แต่การคำนวณอัลกอริทึ่มใช้ระยะเวลาในการคำนวณมาก ในการแก้ปัญหาด้านเวลานี้ จึงมีการประยุกต์ใช้เทคนิคอื่นๆเข้าช่วย เช่น การลดเวลาในการคำนวณโดยกระจายการคำนวณ หรือใช้เทคนิคทางด้านการเขียนโปรแกรมเข้าช่วย ปัจจุบันมีเทคโนโลยีในการจัดการข้อมูลขนาดใหญ่ (Big data) ที่นิยมใช้ในการประมวลผลข้อมูล และช่วยลดเวลาในการคำนวณได้ และ Hadoop [4] คือ กรอบงาน (frame work) ที่ใช้ในการสร้างสภาพแวดล้อมในการประมวลผลข้อมูลขนาดใหญ่และสามารถกระจายข้อมูลและการประมวลผลไปยังเครื่องลูกข่ายได้ และยังมี map-reduce [5] ซึ่งเป็นวิธีการในการเขียนโปรแกรมที่ช่วยให้สามารถสร้างคำสั่งในการประมวลผลข้อมูลโดยกระบวนการ map จะทำการคำนวณและส่งผลลัพธ์ไปยังกระบวนการ reduce ซึ่งเป็นผลให้สามารถแบ่งการประมวลผลเป็นงานย่อยๆได้

ในงานวิจัยนี้ผู้วิจัยได้พัฒนากระบวนการใน การประมวลอัลกอริทึ่มโดยใช้ map-reduce เข้ามาช่วยในการลดเวลาในการคำนวณโดยใช้สภาพแวดล้อมของ Hadoop ซึงเป็นการพัฒนาต่อยอดจากงานวิจัยเดิม [3] ซึ่งผู้วิจัยได้ทำการ หาค่าที่เหมาะสมของพารามิเตอร์สำหรับ LDA โดยการใช้ ACO และทดสอบโดยใช้ข้อมูลมาตรฐาน จาก UCI ในบทความวิจัยนี้ได้แบ่งเนื้อหาออกเป็นหลายส่วนซึ่งประกอบด้วย บทนำ ที่กล่าวถึงความสำคัญของปัญหา และทฤษฎีที่เกี่ยวข้องในส่วนที่สอง และวิธีการดำเนินการวิจัย ผลการทดลอง และตามด้วยการสรุปผลการทดลองในส่วนสุดท้าย

2. วัสดุอุปกรณ์และวิธีการวิจัย

2.1 ทฤษฎีที่เกี่ยวข้อง

ในงานวิจัยนี้ผู้วิจัยได้ทำการศึกษาเกี่ยวกับการทำเหมืองข้อมูลของข้อมูลประเภทข้อความ ซึ่งเป็นการค้นหาหัวข้อ จากชุดข้อมูล โดยการใช้แบบจำลองหัวข้อ (Topic modelling) และได้ใช้ทฤษฎีที่เกี่ยวข้องในการหาค่าที่เหมาะสม ซึ่งเป็นอัลกอริทึ่มด้านการประมวลผลแบบกลุ่ม (Swarm Intelligence) และใช้เทคโนโลยีการจัดการข้อมูลขนาดใหญ่ซึ่งมี Hadoop เป็นระบบหลักและใช้หลักการเขียนโปรแกรมแบบ Map-reduce

Topic modeling

แบบจำลองหัวข้อ เป็นการสร้างแบบจำลองการกระจายตัวของข้อมูล เพื่อนำมาใช้ในการจัดกลุ่มข้อมูลแบบจำลองหัวข้อนี้มีพื้นฐานมาจากแนวคิดที่ว่าในเอกสารเกิดจากการรวมตัวของหลาย ๆ หัวข้อซึ่งแต่ละหัวข้อมีการแจกแจงความน่าจะเป็นของคำที่เกิดขึ้นหลายๆ คำในแต่ละหัวข้อนั้น

กำหนดให้ คำศัพท์ (word) ซึ่งเป็นหน่วยพื้นฐานของข้อมูลอยู่ดัชนีคำศัพท์ของไวยากรณ์หนึ่ง {1,....,v}

เอกสาร คือลำดับแบบอนุกรมของคำศัพท์ แทนด้วย $W= w_1 ,w_2 , ... , w_n \}$

คลังข้อมูล คือหน่วยของเอกสารหลายๆเอกสารมารวมกัน แทนด้วย

$D = \{W_1,W_2,.....,W_n\}$

Latent Dirichlet Allocation (LDA)

Latent Dirichlet Allocation (LDA) [1] เป็นแบบจำลองความน่าจะเป็นสำหรับข้อมูลที่ไม่ต่อเนื่อง เช่น คลังข้อมูล แบบจำลองนี้ถูกสร้างขึ้นโดยมีแนวคิดว่า ในเอกสาร จะประกอบด้วย หัวข้อรวมกันอยู่แบบสุ่ม ซึ่งหัวข้อเหล่านี้ได้กระจัดกระจาย อยู่ในกลุ่มของคำศัพท์ ของเอกสารนั้นๆ

LDA มีกระบวนการดังนี้

1.เลือกค่า N – Poisson (ξ)

2.เลือกค่า θ – Dir(α)

3.ในแต่ละ W_n จากคำศัพท์ N

3.1 เลือกหัวข้อ Z_n – Multinomial(θ)

3.2 เลือกคำศัพท์ W_n จาก $p(w_n \mid z_n, \beta)$ ความน่าจะเป็นของคำศัพท์บนหัวข้อ

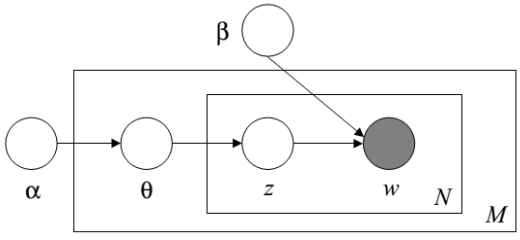

Figure 1 Overview of the LDA system [1]

LDA แตกต่างจากอัลกอริทึ่มอื่นๆ ที่มีความสามารถในการจัดกลุ่มเอกสารคือ มีการแบ่งการทำงานเป็น 3 ชั้น และมีการกระบวนการทำซ้ำ (repeat) ในการเลือกหัวข้อ เป็นผลให้เอกสารหนึ่งๆ อาจมีความเกี่ยวโยงกับหลาย ๆ หัวข้อได้

Optimization with Swarm intelligence

Swarm Intelligence (SI) [6] เป็นวิธีการที่ใช้ในการแก้ปัญหาทางด้าน Optimization โดยใช้หลักการของชีววิทยา เช่นการรวมฝูงของนก มด เป็นต้น ตัวอย่างของระบบ SI ที่นิยมใช้อย่างแพร่หลาย คือ Particle swarm intelligence

(PSO) เป็นระบบที่เลียนแบบพฤติกรรมการรวมฝูงของสัตว์ต่างๆ เช่น การรวมฝูงของนก ปลา พฤติกรรมทางสังคมของมนุษย์ และเป็นเครื่องมือที่ใช้ในการแก้ปัญหา optimization และมักถูกนำมาเปรียบเทียบกับอัลกอริทึมอื่นๆ ที่ใช้ในการแก้ปัญหาประเภทเดียวกัน เช่น Genetic Algorithm (GA) [7], Simulated Annealing (SA) [8] และอื่นๆ ส่วน SI อีกระบบหนึ่งที่เป็นที่นิยม คือ Ant Colony Optimization (ACO) เป็นระบบที่เลียนแบบพฤติกรรมการหาอาหารของมด ที่ใช้ในการแก้ปัญหา optimization ซึ่งแนวคิดหลักคือการสื่อสารระหว่างมด และการใช้ฟีโรโมน เพื่อช่วยในการหาระยะทางที่สั้นที่สุดระหว่างรังและแหล่งอาหาร

Ant Colony Optimization (ACO)

Ant Colony Optimization (ACO) [2] ในธรรมชาติของมดมีพฤติกรรมในการหาอาหาร โดยใช้หลักการสุ่ม (Random) เพื่อหาเส้นทางในการเดินทางไปยังแหล่งอาหาร และในการเดินทางมีการปล่อยฟีโรโมนไปตามเส้นทางที่เดินไป เมื่อมดตัวอื่นๆ เดินทางมาพบฟีโรโมนที่ปล่อยไว้ก็จะหยุดการสุ่มหาเส้นทาง และเดินทางตามเส้นทางที่มีฟีโรโมนและสามารถเดินทางระหว่างแหล่งอาหารและรังได้อย่างถูกต้อง แต่อย่างไรก็ตาม เมื่อระยะเวลาผ่านไปฟีโรโมนที่อยู่ตามเส้นทางจะมีความเจือจางลง โดยเส้นทางที่มีการเดินผ่านน้อยฟีโรโมนจะลดลงเรื่อยๆ แต่เส้นทางที่มีมดเดินทางผ่านมาก ฟีโรโมนจะมีความเข้มมากขึ้นและเส้นทางนี้จะเป็นเส้นทางที่มีระยะทางสั้นที่สุด

ACO จะประกอบด้วยตัวแทน (agent) ที่ทำงานร่วมกันหลายๆตัว ซึ่งเปรียบเสมือน มด ซึ่งจะมีพฤติกรรมการเดินทางไปทั้ง graph ที่แสดงปัญหาที่ต้องแก้ไข โดยมีกระบวนการสำคัญสองอย่างคือ การร่วมมือกัน (cooperation) และการปรับเปลี่ยน (adaptation) ACO จะต้องตรงตามข้อกำหนดดังนี้

1.ปัญหาที่ต้องการแก้ไข ต้องมีความเหมาะสมกับระบบ ACO โดย มด ต้องสามารถปรับปรุง ผลลัพธ์ของปัญหาได้โดยใช้หลักความน่าจะเป็นและใช้หลักการเปลี่ยนแปลงค่า ฟีโรโมน (τ)

กำหนดให้มด m เพื่อใช้ในการแก้ปัญหา ค่าฟีโรโมน τ_{ij} สอดคล้องกับเส้นทางจาก i ไปยัง j ค่าฟีโรโมนมีการเปลี่ยนแปลงดังนี้

$$\tau_{ij} \leftarrow (1-\rho)\cdot\tau_{ij} + \sum_{k=1}^{m}\Delta\tau_{ij}^{k}$$

โดยค่า ρ คืออัตราการระเหยของฟีโรโมน

$\Delta\tau_{ij}^{k}$ คือปริมาณของฟีโรโมนที่อยู่บนเส้นทาง i ไปยัง j

$$\Delta\tau_{ij}^{k}=\begin{cases}\dfrac{Q}{L_k} & \text{ถ้ามด k ใช้เส้นทาง (i,j)}\\ 0 & \text{กรณีอื่นๆ}\end{cases}$$

Q คือ ค่าคงที่ และ L_k คือ ระยะทางที่มด k สร้างขึ้น

ในการหาผลลัพธ์ของปัญหา มดจะเลือกสถานที่โดยกระบวนการสุ่ม (Stochastic mechanism) เมื่อมด k อยู่ในสถานที่ i และต้องการสร้างผลลัพธ์ย่อย ความน่าจะเป็นในการเดินทางไปสถานที่ j คือ

$$p_{ij}^{k}=\begin{cases}\dfrac{\tau_{ij}^{\alpha}\cdot n_{ij}^{\beta}}{\sum_{c_{ij}\in N(s^p)}\tau_{ij}^{\alpha}\cdot n_{ij}^{\beta}} & \text{ถ้า } c_{ij}\in N(s^p)\\ 0 & \text{กรณีอื่นๆ}\end{cases}$$

เมื่อ $N(s^p)$ คือ เซตของความเป็นไปได้ ของเส้นทาง (i,l) ซึ่ง l เป็นสถานที่ที่มด k ไม่เคยผ่าน และตัวแปร α,β ควบคุมค่าฟีโรโมน

2. มีการกำหนดค่า heuristic function η ที่ใช้วัดคุณภาพขององค์ประกอบที่จะเพิ่มเข้าไปยังผลลัพธ์ย่อย (partial solution)

$$\eta_{ij}=\frac{1}{d_{ij}}$$

เมื่อ d_{ij} คือระยะทางระหว่างสถานที่ i และ j

ซึ่งในงานวิจัยเดิม [3] ผู้วิจัยได้ดำเนินการวิจัยพัฒนาอัลกอริทึม LDA-ACO ใช้ในการค้นหาหัวข้อของเอกสาร โดยใช้ ACO มาประยุกต์ใช้เพื่อปรับค่าพารามิเตอร์และสามารถเพิ่มประสิทธิภาพของอัลกอริทึม LDA ได้แต่ระยะเวลาในการคำนวณจะใช้เวลาเพิ่มมากขึ้น ดังนั้นจึงมีแนวคิดในการลดระยะเวลาในการคำนวณโดยกำหนดสภาพแวดล้อมการทำงานของข้อมูลขนาดใหญ่ ซึ่งเป็นระบบกรอบงานที่ประกอบด้วยส่วนย่อยดังนี้

Hadoop [4] เป็นกรอบงานที่ใช้ในจัดเก็บและประมวลผลข้อมูลชุดขนาดใหญ่ ซึ่งพัฒนาโดย apache ประกอบด้วยส่วนสำคัญดังนี้

1.Hadoop common ประกอบด้วยที่เก็บรวบรวมชุดคำสั่ง และโปรแกรมอรรถประโยชน์ ที่จำเป็นในการทำงานของส่วนอื่นๆ

2.Hadoop distributed file system (HDFS)[9] ระบบไฟล์ที่ใช้ในการเก็บข้อมูลของระบบ Hadoop ถูกออกแบบให้สามารถกระจายข้อมูลเพื่อประมวลผล พร้อมกันในหลายๆ work station ได้และยังทำงานได้ดีกับเครื่องที่มีสมรรถนะไม่สูงมากนัก สามารถเข้าถึงข้อมูลดี จึงเหมาะแก่ชุดข้อมูลที่มีขนาดใหญ่ สถาปัตยกรรมของ HDFS เป็นแบบ master – slave ซึ่งประกอบด้วย master cluster 1 ตัวที่ทำหน้าที่ name node ทำหน้าที่จัดการระบบไฟล์ และ slave node เป็น data node ตามจำนวนที่ต้องการเพื่อเก็บข้อมูล เมื่อผู้ใช้เก็บข้อมูลในระบบ HDFS ข้อมูลไฟล์นั้นจะถูกแบ่งเป็น block และจัดเก็บไว้ใน data node

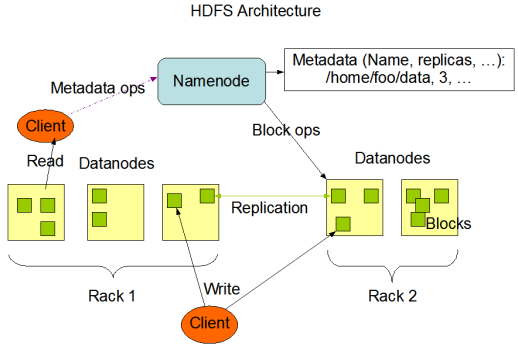

Figure 2 HDFS Architecture [9]

Map-Reduce [5] เป็นกรอบงาน ที่ใช้ในการประมวลผลข้อมูลในระบบฐานข้อมูลขนาดใหญ่ โดยใช้คอมพิวเตอร์จำนวนมาก ซึ่งประกอบกันเป็นกลุ่ม (cluster) โดยทำการสร้างแอปพลิเคชั่น ให้จัดการข้อมูลตามต้องการ ซึ่งข้อมูลที่นำมาประมวลผลอาจเป็นข้อมูลที่เป็นโครงสร้างซึ่งอยู่ในระบบฐานข้อมูลหรือแบบไม่มีโครงสร้างที่เก็บในระบบไฟล์ก็ได้ ระบบของ map-reduce จะมี master node 1 ตัว ทำหน้าที่เป็น Job tracker และที่ worker node จะมี task tracker อยู่กลุ่มละ 1 ตัว การทำงานของ master node จะจัดตารางการทำงานให้แก่ worker node และควรตรวจตราการทำงาน เมื่อมีข้อผิดพลาดจะสามารถสั่งให้ worker node ทำงานนั้น ๆ ใหม่ไป ขั้นตอนการทำงานหลักแบ่งเป็น 2 ขั้นตอน

Map step ขั้นตอนที่ Master node รับข้อมูล input แล้วทำการแบ่งให้เป็นปัญหาย่อยๆ และกระจายสู่ worker node และ worker node สามารถกระทำกระบวนการนี้ซ้ำได้ ซึ่งจะได้โครงสร้างการทำงานแบบต้นไม้ (Tree) worker node เมื่อประมวลผลเสร็จแล้วจะส่งผลลัพธ์กลับสู่ master node รูปแบบของข้อมูลนำเข้าของ map-reduce จะเป็นรูปแบบ <key-value> และผลลัพธ์ที่ได้ก็เป็นรูปแบบ <key-value> เช่นกัน

Reduce step ขั้นตอนที่ master node รวบรวมผลลัพธ์จาก worker node เพื่อสร้างเป็นผลลัพธ์ตามระบบงานของ map-reduce การทำงานหนึ่ง ๆ จะมีรูปแบบดังนี้

(input) <k1,v1> -> **map** -> <k2,v2> -> **combine** -> <k2,v2> -> **reduce** -> <k3, v3> (output)

2.2 วิธีดำเนินการวิจัย

ในส่วนของการดำเนินการวิจัยจะประกอบด้วยการอธิบายข้อมูลที่ใช้ในการทดลอง การเตรียมอุปกรณ์สำหรับประมวลผล การกำหนดคาตัวแปรตางๆให้แก่อัลกอริทึมและการวัดผล

ชุดข้อมูล

ข้อมูลที่ใช้ในการวิจัย คือ ชุดข้อมูลจาก UCI machine learning repository จำนวน 3 ชุด ซึ่งผ่านกระบวนการตัดคำและทำการลบคำที่ไม่สำคัญออกแล้ว ซึ่งชุดข้อมูลนี้ถูกใช้งานอย่างแพร่หลายในการจำแนกข้อมูล (classification) และจัดกลุ่มข้อมูล (clustering) ชุดข้อมูลเหล่านี้ประกอบด้วย NIPS เป็นชุดข้อมูลที่เกี่ยวกับเอกสารงานวิจัยที่ได้จากการบริจาคของกลุ่มนักวิจัยที่ศึกษาเกี่ยวกับอัลกอริทึมการเรียนรู้ , Enron เป็นชุดข้อมูลที่เกี่ยวกับโครงการ CALO - (A Cognitive Assistant that Learns and Organizes) ซึ่งเก็บจากผู้ใช้งานนับร้อยคน KOS- เป็นชุดข้อมูลที่ได้จาก blog ซึ่งคุณลักษณะของข้อมูลทั้งหมดจะแสดงใน (Table 1) โดย D คือ จำนวนเอกสาร N คือจำนวน Token ทั้งหมด และ W คือขนาดของคำศัพท์

Table 1 Summarization of datasets

dataset	D	W	N
Kos	430	906	467714
Nips	500	2419	1,900,000 (approx.)
Enron	9861	8102	6,400,000 (approx.)

ในขั้นตอนการเตรียมเครื่องมือเพื่อให้อัลกอริทึมสามารถทำงานบนระบบ Hadoop ซึ่งเป็นการจัดสภาพแวดล้อมให้มีการทำงานให้รองรับ map-reduce ระบบทำการตั้งตั้ง Hadoop ซึ่งเป็นกรอบงานหลักและทำการติดตั้งชุดคำสั่ง เพื่อให้สามารถพัฒนาโปรแกรมด้วยภาษา python ซึ่งใช้ในการพัฒนาอัลกอริทึม ACO และ ทำการติดตั้งชุดคำสั่งเสริม Gensim ที่มีชุดคำสั่งสำหรับการทำ text mining และให้สามารถเรียกใช้งาน LDA ได้ ระบบฮาร์ดแวร์ที่ใช้ประกอบด้วย CPU intel CPU Core(TM) i7-4700HQ ความเร็วสัญญาณนาฬิกา 2.40 GHz หน่วยความจำหลัก (RAM) 12.0 GB และใช้ระบบปฏิบัติการ Linux 64 bit

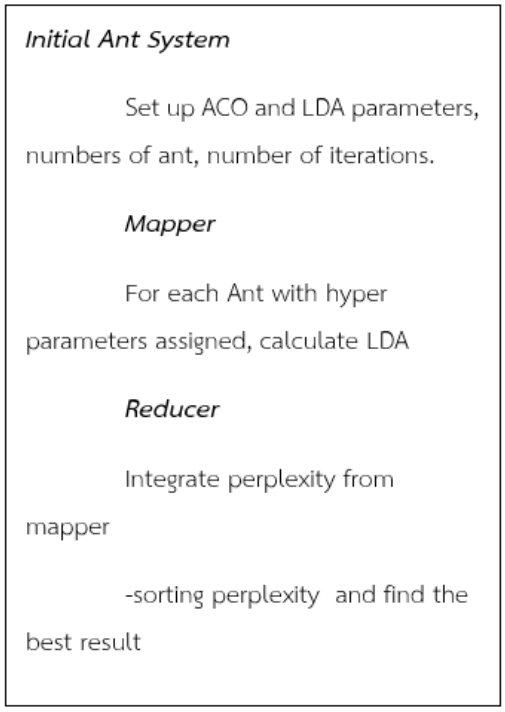

Figure 3 process of Perplexity computation in Map and reduce step.

ขั้นตอนต่อมาคือการกำหนดตัวแปรที่เกี่ยวข้อง ที่จำเป็นต้องใช้ในกระบวนการเริ่มต้นประมวลผล ตามกระบวนการของการทดลอง [3] ซึ่งแบ่งเป็นการกำหนดค่าตัวแปรให้แก่ ACO ประกอบด้วย จำนวนมด (ant) จำนวนรอบในการทำงาน ค่าฟีโรโมนและค่าอัตราการระเหยของฟีโรโมน และทำการแบ่งช่วงค่าตัวแปรของ LDA คือค่า α และ β ซึ่งจะใช้ในการคำนวณ ในแต่ละรอบของการทำงานมดจะทำการปล่อยฟีโรโมนลงบนเส้นทางที่ผ่านซึ่งมีปริมาณตามค่าฟีโรโมน (pheromone constant) และค่าการระเหย (decay constant) ที่กำหนดไว้ มดแต่ละตัวจะทำการเลือกเส้นทางเพื่อเก็บค่า α และ β เพื่อจะส่งนำไปคำนวณ LDA ในกระบวนการ map-reduce ดังแสดงใน (Figure 4)

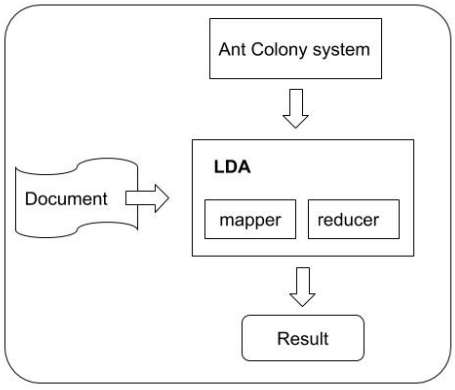

Figure 4 Overview of the system which consist of Hadoop map-reduce and ACO –LDA parameters.

ในกระบวนการของ map-reduce มดแต่ละตัวจะนำค่าพารามิเตอร์ของ LDA มาแล้วทำการประมวลผลในขั้นตอนของ Map ซึ่งระบบจะทำการประมวลผลชุดข้อมูลที่กำหนดด้วย LDA ตามค่าพารามิเตอร์ที่ส่งมา หลังจากประมวลผลแล้วจะได้ค่า Perplexity ซึ่งค่านี้จะถูกส่งไปยังกระบวนการ reduce คู่กับหมายเลขกำกับของมด ในกระบวนการของการ reduce ค่า Perplexity จะถูกเก็บรวบรวมกันและจะเลือกค่าที่ดีที่สุดในแต่ละรอบ (local best) หลังจากจบรอบแล้วค่าฟีโรโมนจะมีการเปลี่ยนแปลงตามค่าคงที่ที่กำหนดไว้ และ ฟีโรโมนในบางเส้นทางจะมีการระเหย เมื่อทำกระบวนการปรับปรุงค่าฟีโรโมนแล้วจะทำการเริ่มกระบวนขั้นต้นอีกจนครบจำนวนรอบการทำงานที่กำหนด เมื่อการทำงานครบทุกรอบแล้วจะทำการคำนวณเพื่อหาค่าที่ดีที่สุดได้ของการคำนวณทั้งหมดเป็นค่า (global Best) และทำการบันทึกเวลาที่ใช้ในการคำนวณทั้งหมด ดังแสดงใน (Figure 5)

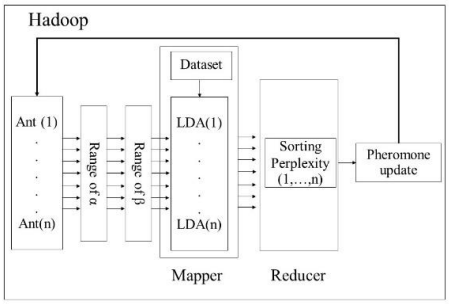

Figure 5 Overview of the system which consist of Hadoop map-reduce and ACO –LDA parameter

การวัดประสิทธิภาพ

การวัดประสิทธิภาพของแบบจำลองหัวข้อ ใช้ค่า Perplexity ซึ่งวัดจากข้อมูลทดสอบที่ใช้กับแบบจำลองหัวข้อที่พัฒนาขึ้น โดยค่า Perplexity ที่มีค่าต่ำ แสดงให้เห็นถึงประสิทธิภาพของแบบจำลองหัวข้อที่ดี สำหรับข้อมูลทดสอบจำนวน M ชุด

$$perplexity\left(D_{test}\right)=\exp\left\{-\frac{\sum_{d=1}^{M}\log p\left(W_d\right)}{\sum_{d=1}^{M}N_d}\right.$$

ส่วนการวัดประสิทธิภาพด้านเวลา ผู้วิจัยทำการวัดระยะเวลาในการคำนวณอัลกอริทึ่ม ACO-LDA โดยเปรียบเทียบระหว่าง ACO-LDA ที่ทำงานภายใต้สภาพแวดล้อมของ Hadoop และ ไม่ได้อยู่ภายใต้สภาพแวดล้อมของ Hadoop และทำการเปรียบเทียบ

3. ผลการวิจัยและอภิปรายผลการวิจัย

ผลการทดลองได้แบ่งออกตามชุดข้อมูลที่ใช้ ในผลของชุดข้อมูล KOS ซึ่งทำการกำหนดค่าเริ่มต้นของอัลกอริทึ่ม LDA คือจำนวนหัวข้อ K=10 และกำหนดค่าเริ่มต้นของ ACO คือ จำนวนมดมีค่าเท่ากับ 5 และทำการประมวลผลค่าทั้งหมด 25 ครั้ง และกำหนดค่าคงที่ของฟีโรโมน คือ 1.0 ค่าการระเหยของฟีโรโมน คือ 0.5 และค่าตัวแปรของ LDA ค่า α จะอยู่ในช่วง 0.005-0.25 และ ค่า จะอยู่ในช่วง 0.0025-0.03 ผลลัพธ์ที่ได้ดังแสดงใน (Table 2)

Table 2 Perplexity ACO-LDA with Hadoop environment machine using KOS dataset.

iteration	Ant	α	β	Perplexity
1	Ant-5	0.05	0.0025	262.114
2	Ant-3	0.05	0.0025	260.103
3	Ant-5	0.15	0.0225	253.357
4	Ant-1	0.25	0.015	**253.300**

ในการคำนวณค่า Perplexity โดยใช้ชุดข้อมูล KOS ใช้เวลาในการคำนวณ 723.405 วินาทีส่วนการคำนวณโดยไม่ใช่สภาพแวดล้อมของ Hadoop ใช้เวลา 838.797 วินาที ผลลัพธ์ของการหาหัวข้อโดยใช้อัลกอริทึม ACO-LDA ของชุดข้อมูล KOS แสดงใน (Table 3)

Table 3 Topic of KOS datasets (K=10)

Topic	Term
0	0.017*house + 0.015*delay + 0.012*republican + 0.011*district + 0.011*democrats + 0.009*senate +0.009*republicans + 0.009*elections + 0.008*gop + 0.007*race
1	0.030*dean + 0.013*clark + 0.013*kerry + 0.012*lieberman + 0.012*primary + 0.010*democratic + 0.010*gephardt + 0.009*edwards + 0.007*dec + 0.007*iowa
2	0.011*bush + 0.009*november + 0.009*kerry + 0.008*poll + 0.007*general + 0.006*campaign +0.006*percent + 0.006*democratic + 0.006*media + 0.006*vote
3	0.025*bush + 0.013*iraq + 0.011*president + 0.011*war + 0.010*kerry + 0.006*bushs + 0.006*administration + 0.005*general + 0.004*people + 0.004*house
4	0.013*bush + 0.010*kerry + 0.008*poll + 0.007*state + 0.007*states + 0.007*democratic +0.005*campaign + 0.005*percent + 0.005*democrats + 0.005*polls
5	0.009*bush + 0.007*party + 0.007*war + 0.006*percent + 0.006*democratic + 0.005*iraq + 0.005*house + 0.004*kerry + 0.004*election + 0.004*state
6	0.024*bush + 0.009*kerry + 0.008*poll + 0.006*party + 0.006*campaign + 0.006*general +0.005*democratic + 0.005*time + 0.005*race + 0.004*election
7	0.008*bush + 0.008*republican + 0.007*republicans + 0.007*democrats + 0.006*democratic + 0.006*party + 0.005*iraq + 0.005*senate + 0.005*people + 0.005*general
8	0.016*kerry + 0.013*bush + 0.007*poll + 0.007*democratic + 0.006*dean + 0.006*clark + 0.005*campaign + 0.005*democrats + 0.005*polls + 0.005*house
9	0.037*november + 0.011*bush + 0.009*poll + 0.009*house + 0.008*kerry + 0.008*senate + 0.008*republicans + 0.008*governor + 0.007*account + 0.007*electoral

ในการประมวลผลชุดข้อมูล Nips ซึ่งทำการกำหนดค่าเริ่มต้มของอัลกอริที่ม LDA คือจำนวนหัวข้อ K=10 และกำหนดค่าเริ่มต้นของ ACO คือ จำนวนมดมีค่าเท่ากับ 5 และทำการประมวลผลค่าทั้งหมด 25 ครั้ง และกำหนดค่าคงที่ของ ฟีโรโมน คือ 1.0 ค่าการระเหยของฟีโรโมน คือ 0.5 และค่าตัวแปรของ LDA ค่า α จะอยู่ในช่วง 0.005-0.25 และ ค่า จะอยู่ในช่วง 0.0025-0.03 ผลลัพธ์ที่ได้ดังแสดงใน (Table 4)

Table 4 Perplexity ACO-LDA with Hadoop environment machine using Nips dataset.

iteration	Ant	α	β	Perplexity
1	Ant-1	0.15	0.03	250.977
2	Ant-2	0.15	0.03	248.651
3	Ant-1	0.025	0.0275	247.905
4	Ant-5	0.15	0.03	**245.506**

ในการคำนวณค่า Perplexity โดยใช้ชุดข้อมูล Nips ใช้เวลาในการคำนวณ 3,840.620 วินาทีส่วนการคำนวณโดยไม่ใช่สภาพแวดล้อมของ Hadoop ใช้เวลา 1,936.258วินาที ในการประมวลผลชุดข้อมูล Enron ซึ่งทำการกำหนดค่าเริ่มต้มของอัลกอริทึ่ม LDA คือจำนวนหัวข้อ K=10 และกำหนดค่าเริ่มต้นของ ACO คือ จำนวนมดมีค่าเท่ากับ 5 และทำการประมวลผลค่าทั้งหมด 25 ครั้ง และกำหนดค่าคงที่ของฟีโรโมน คือ 1.0 ค่าการระเหยของฟีโรโมน คือ 0.5 และค่าตัวแปรของ LDA ค่า α จะอยู่ในช่วง 0.005-0.25 และ ค่า จะอยู่ในช่วง 0.0025-0.03 ผลลัพธ์ที่ได้ดังแสดงใน (Table 5)

Table 5 Perplexity ACO-LDA with Hadoop environment machine using Enron dataset.

Iteration	Ant	α	β	Perplexity
1	Ant-5	0.2	0.015	**325.760**
2	Ant-1	0.25	0.010	329.273
3	Ant-4	0.15	0.015	330.374
4	Ant-5	0.25	0.015	328.911

การคำนวณค่า Perplexity โดยใช้ชุดข้อมูล Enron ใช้เวลาในการคำนวณ 7,115.431 วินาทีส่วนการคำนวณโดยไม่ใช่สภาพแวดล้อมของ Hadoop ใช้เวลา 4,390.067 วินาที

ผลการทดลองแสดงให้เห็นว่าการประยุกต์ใช้ Map-reduce ในการทำงานของอัลกอริทึ่ม LDA-ACO ทำให้ความเร็วในการประมวลผลเพิ่มขึ้น ซึ่งเป็นการทดลองโดยใช้เครื่องคอมพิวเตอร์เพียงเครื่องเดียว ดังตารางแสดงใน (Table 6)

Table 6 Comparison of ACO-LDA computation time.

Dataset	Non Map-reduce	Map-reduce	% improve
KOS	838.797	723.405	**13.76%**
Nips	3,840.620	1,936.258	**49.58%**
Enron	7,115.431	4,390.067	**38.30%**

จาก (Table 5) แสดงให้เห็นว่า การทดลองหาหัวข้อโดยอัลกอริทึ่ม ACO-LDA ในชุดข้อมูล KOS ภายใต้สภาพแวดล้อมของ Hadoop นั้นการทำงานเร็วขึ้น 13.76 % ใช้เวลา 723.405 วินาที หากทำงานโดยไม่ได้อยู่ภายใต้สภาพแวดล้อมของ Hadoop ใช้เวลา 838.797 วินาที ส่วนในชุดข้อมูล Nips การทดลองหาหัวข้อ โดยอัลกอริทึ่ม ACO-LDA ภายใต้สภาพแวดล้อมของ Hadoop นั้นการทำงานเร็วขึ้น 49.58% ใช้เวลา 1,936.258 วินาที หากทำงานโดยไม่ได้อยู่ภายใต้สภาพแวดล้อมของ Hadoop ใช้เวลา 3,840.620 วินาที และ ในชุดข้อมูล Enron ภายใต้สภาพแวดล้อมของ Hadoop นั้นการทำงานเร็วขึ้น 38.30% ใช้เวลา 7,115.431 วินาที หากทำงานโดยไม่ได้อยู่ภายใต้สภาพแวดล้อมของ Hadoop ใช้เวลา 4,390.067 วินาที

จากผลการวิจัยข้างต้น การทำงานภายใต้สภาพแวดล้อมของ Hadoop ในการประมวลผล ACO-LDA จะมีการทำงานหลายรอบตามจำนวนค่าที่ได้กำหนดไว้ เมื่อมีการใช้ map-reduce เข้ามาช่วยจะสามารถแบ่งการประมวลผลอัลกอริทึ่ม เข้าสู่ mapper ในกระบวนการ map ซึ่งเป็นการประมวลผลแบบกระจาย จึงทำให้การประมวลผลเร็วขึ้น และรวบรวมผลลัพธ์จาก mapper ในกระบวนการ reducer แต่การประมวลผลรูปแบบนี้ จะไม่เหมาะกับข้อมูลที่มีขนาดเล็กมาก หรืองานที่ไม่ต้องใช้เวลาในการประมวลผลนาน เพราะจะต้องเสียเวลาในการเริ่มระบบของ Hadoop และ map-reduce

4. สรุปและเสนอแนะ

จากการวิจัยเรื่อง การเพิ่มประสิทธิภาพแบบจำลองหัวข้อด้วยสภาพแวดล้อมแบบข้อมูลขนาดใหญ่ ผู้วิจัยทำการประมวลผลชุดข้อมูล ซึ่งได้จาก UCI ประกอบด้วยชุดข้อมูล KOS , Enron , Nips ซึ่งนิยมใช้ในการวิจัยด้านการทำเหมืองข้อมูลประเภทข้อความ ผู้วิจัยได้ทำการการประยุกต์ใช้ Map-

reduce ซึ่งทำงานบน Hadoop มาช่วยในการทำงานของอัลกอริทึม ACO-LDA ที่ใช้เวลาในการประมวลผลมากและมีการทำงานซ้ำหลายรอบการทำงาน ได้ผลวิจัยคือ ระยะเวลาที่ใช้ในการคำนวณค่า Perplexity ลดลง เนื่องจากมีการใช้ map-reduce ที่ช่วยการประมวลผลแบบแบ่งตาม mapper ที่กำหนดไว้

กิตติกรรมประกาศ

ขอขอบพระคุณ มหาวิทยาลัยกาฬสินธุ์ ที่ได้สนับสนุนงบประมาณในการวิจัยครั้งนี้

เอกสารอ้างอิง

[1] Blei, David M., Andrew Y. Ng, and Michael I. Jordan. 2003. "Latent dirichlet allocation." **Journal of machine Learning research** : 993-1022.

[2] Dorigo, Marco, and Gianni C. 1999. "Ant colony optimization: a new metaheuristic. **Evolutionary Computation**. 1999. CEC

[3] Yarnguy, T. and Kanarkard W. 2018. "Tuning Latent dirichet allocation parameters using ant colony optimization" . **Journal of telecommunication , electronic and computer engineering**. Vol.10 21-24 .

[4] The Apache Software Foundation. 2008**. Apache Hadoop.** https://hadoop.apache.org/ . Accessed 29 October 2018.

[5] The Apache Software Foundation. 2008**. MapReduce Tutorial.** http://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html Accessed 29 October 2018.

[6] Liu, Yang, and Kevin M. Passino. 2000. "Swarm intelligence : Literature overview. "Department of Electrical Engineering, the Ohio State University .

[7] A. Panichella, B. and et al. "How to Effectively Use Topic Models for Software Engineering Tasks ? An Approach Based on Genetic Algorithms."

[8] Kuzmenko, Andrey. 2014. "Simulated Annealing for Dirichlet Priors in LDA." .

[9] Borthakur, D. 2008. **Hadoop file system: architecture guide** . http://archive.cloudera.com/cdh/3/hadoop-0.20.2-cdh3u6/ hdfs_design.pdf. Accessed 21 August 2018.

[10] Hasanpour, E. H. 2010. "PSO algorithm for text clustering based on latent semantic indexing. " **The Fourth Iran Data Mining Conference**. Tehran, Iran.

[11] Latha, K., and R. Rajaram. 2008."An Efficient LSI based Information Retrieval Framework using Particle swarm optimization and simulated annealing pproach." **Advanced Computing and Communications, 2008**. ADCOM 2008. 16th International Conference on. IEEE.