การเปรียบเทียบการคำนวณน้ำหนักดัชนี สำหรับอัลกอริทึมการจัดหมวดหมู่เอกสารภาษาไทย

A Comparative Study on Term Weight Techniques for Thai Document Categorization

นิเวศ จิระวิชิตชัย[1] ปริญญา สงวนสัตย์[2] และ พยุง มีสัจ[3]

Nivet Chirawichitchai[1] Parinya Sanguansat[2] and Phayung Meesad[3]

[1]ภาควิชาเทคโนโลยีสารสนเทศ คณะเทคโนโลยีสารสนเทศ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ กรุงเทพฯ

[2]ภาควิชาเทคโนโลยีสารสนเทศ คณะเทคโนโลยีสารสนเทศ มหาวิทยาลัยรังสิต กรุงเทพฯ

[3]ภาควิชาครุศาสตร์ไฟฟ้า คณะครุศาสตร์อุตสาหกรรม มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ กรุงเทพฯ

บทคัดย่อ

งานวิจัยนี้ได้นำเสนอแบบจำลองการจัดหมวดหมู่เอกสารภาษาไทย โดยมุ่งเน้นศึกษาเปรียบเทียบวิธีการคำนวณค่าน้ำหนักที่เหมาะสม และมีประสิทธิภาพดีที่สุดในการจัดหมู่เอกสารข่าวภาษาไทย จากการทดลองพบว่า การคำนวณค่าน้ำหนักด้วยวิธี ltc weight เมื่อพิจารณาจากค่าเฉลี่ยให้ประสิทธิภาพในการจัดหมวดหมู่เอกสารภาษาไทยดีที่สุด เมื่อเทียบกับการคำนวณน้ำหนักแบบอื่นๆ ในทุกอัลกอริทึม และเมื่อพิจารณาพารามิเตอร์ที่ส่งผลให้ประสิทธิภาพด้านการจัดหมวดหมู่เอกสารภาษาไทยดีที่สุด พบว่าอัลกอริทึม Support Vector Machine ร่วมกับวิธี ltc weight ที่จำนวน 8000 คุณลักษณะ ให้ประสิทธิภาพจัดหมวดหมู่ที่ระดับ 94.50 % รองลงมาเป็นอัลกอริทึม Naïve-Bayes ร่วมกับวิธี ltc weighting ที่จำนวน 10000 คุณลักษณะ ให้ประสิทธิภาพจัดหมวดหมู่ที่ระดับ 90.85 % และสุดท้ายอัลกอริทึม Decision Tree ร่วมกับวิธี ltc weighting ที่จำนวน 8000 คุณลักษณะ ให้ประสิทธิภาพจัดหมวดหมู่ที่ระดับ 74.50 % ตามลำดับ

คำสำคัญ : การคำนวณดัชนี , การจัดหมวดหมู่เอกสาร , เครื่องจักรเรียนรู้

Abstract

This research presented Thai document categorization framework. The comparative study focuses on how to calculate the term weight values that give best performance for Thai document categorization framework. Our experimental results showed that ltc weight method is most effective on Thai document categorization. The best performance for Thai document categorization is support Vector Machine classifier with ltc weight at 8000 features yielded a very high

classification performance with the F1 measure equal to 94.50%. Followed by a Naïve-Bayes classifier with ltc weight at 10000 features yielded performance with the F1 measure equal to 90.85%, and finally a Decision Tree classifier with ltc weighting features of the 8000 performance yielded performance with the F1 measure equal to 74.50%, respectively.

Keywords : Term weighting, Text Categorization, Machine Learning

1. บทนำ

การขยายตัวทางการใช้งานระบบคอมพิวเตอร์และอินเทอร์เน็ต ตลอดช่วงระยะเวลาที่ผ่านมาจนถึงปัจจุบันมีแนวโน้มในการใช้งานเพิ่มมากขึ้นอย่างรวดเร็ว ส่งผลให้เกิดการสร้างและเก็บข้อมูลหลายชนิดในรูปแบบอิเล็กทรอนิกส์ ซึ่งหนึ่งในข้อมูลอิเล็กทรอนิกส์เหล่านี้คือข้อมูลประเภทเอกสาร เช่น จดหมายอิเล็กทรอนิกส์ (E-mail), เว็บเพจ (Web page) เอกสารข่าว (News) และไฟล์งานเอกสารต่างๆ (Document) ซึ่งเป็นข้อมูลที่มีปริมาณและเนื้อหาที่หลากหลายมากขึ้น ทำให้ยากต่อการค้นหาและจัดเก็บหมวดหมู่เอกสาร ดังนั้นการสืบค้นและการจัดการเอกสารจะง่ายและเป็นไปตามความต้องการ ต้องอาศัยการจัดแบ่งเอกสารเป็นกลุ่มหรือหมวดหมู่ให้สอดคล้องและตรงกับดัชนี เพื่อให้จัดเก็บและค้นคืนเอกสารได้อย่างรวดเร็วและมีประสิทธิภาพ จึงมีความจำเป็นต้องอาศัยผู้เชี่ยวชาญในการจัดกลุ่มข้อมูล ฉะนั้นจึงเป็นการยากในการที่จะจัดกลุ่มหรือแยกประเภทเอกสาร ยิ่งหากเอกสารมีปริมาณมากขึ้นทุกๆวัน ทำให้ต้องพึ่งพาทรัพยากรบุคคลในการจัดหมวดหมู่เอกสารเหล่านี้มากตามไปด้วยเช่นกัน ทำให้มีการคิดค้นพัฒนากระบวนการในการจัดหมวดข้อมูลที่มีขนาดใหญ่เหล่านี้ให้เป็นไปแบบอัตโนมัติ เพื่อที่จะสามารถจำแนกกลุ่มข้อมูลเพื่อใช้ประโยชน์จากข้อมูลและการจัดการกับข้อมูลให้มีประสิทธิภาพ รองรับการสืบค้นจากผู้ใช้งานเอกสารอย่างถูกต้องและเหมาะสม

ปัจจุบันได้มีศึกษาเกี่ยวกับการนำวิธีการเรียนรู้ด้วยคอมพิวเตอร์ มาประยุกต์ร่วมกับการประมวลผลภาษาธรรมชาติเพื่อใช้จัดแบ่งกลุ่มเอกสารนั้น สามารถแบ่งได้ 2 ลักษณะ คือ การจัดกลุ่ม (Clustering) และการจำแนกหมวดหมู่ (Classification หรือ Categorization) การจัดกลุ่มเอกสาร คือ การแบ่งกลุ่มตามเนื้อหาของเอกสารโดยไม่มีการกำหนดกลุ่มหรือหมวดหมู่ของเอกสารไว้ก่อน ซึ่งจะเป็นการแบ่งกลุ่มตามลักษณะของเอกสาร โดยเอกสารที่มีลักษณะเหมือนกันจะอยู่ด้วยกัน ส่วนการจำแนกหมวดหมู่เอกสาร คือ การแบ่งกลุ่มตามเนื้อหาของเอกสาร โดยที่มีการกำหนดกลุ่มหรือหมวดหมู่ของเอกสารไว้ก่อน โดยจะเปรียบเทียบเอกสารกับต้นแบบในแต่ละหมวดหมู่เอกสาร จะถูกจัดอยู่ในหมวดหมู่ที่ต้นแบบมีลักษณะคล้ายกับตัวมันเองมากที่สุด โดยผลลัพธ์ที่ได้จากวิธีการเรียนรู้ด้วยคอมพิวเตอร์นั้น ความถูกต้องใกล้เคียงกับผลการจำแนกหมวดหมู่ของเอกสารที่ทำโดยมนุษย์ ทำ

ให้ประหยัดแรงงานมนุษย์เป็นอย่างมากเพราะไม่ต้องอาศัยผู้เชี่ยวชาญในการจำแนกประเภทเอกสาร หรือปรับเปลี่ยนหมวดหมู่ของเอกสาร [1-2]

ขั้นตอนที่สำคัญอย่างมากในการจัดหมวดหมู่เอกสารขั้นตอนหนึ่งก็คือ การสร้างดัชนีให้กับคุณลักษณะที่สกัดได้จากเอกสาร โดยการคำนวณค่าน้ำหนักที่จะมาใช้เป็นค่าคุณลักษณะของเอกสาร หรืออาจจะเรียกได้ว่าการเป็นการหาค่าน้ำหนักให้กับดัชนี เพื่อจะใช้ในการเรียนรู้ในการสร้างแบบจำลองต่อไป จากความสำคัญดังกล่าวผู้วิจัยจึงมีแนวคิดที่จะทดสอบวิธีการคำนวณค่าน้ำหนักดัชนีแบบต่างๆที่มีการใช้งานทางด้านการค้นคืนสารสนเทศ (Information Retrieval) ในภาษาต่างประเทศ พบว่ามีประสิทธิภาพการค้นคืนในเกณฑ์ที่ดี มาประยุกต์ในการให้น้ำหนักดัชนีกับการจัดหมวดหมู่เอกสารภาษาไทย โดยทำการทดสอบประสิทธิภาพในการจัดหมวดหมู่เอกสารกับอัลกอริทึมต้นไม้ตัดสินใจ (Decision Tree) ซัพพอร์ทเวกเตอร์แมชชีน (Support Vector Machine) และเนอีฟเบย์ (Naïve-Bayes) [3]

2. ทฤษฎีที่เกี่ยวข้อง

2.1 การสกัดคุณลักษณะ (Feature Extraction)

วัตถุประสงค์ของขั้นตอนการสกัดคุณลักษณะเอกสารคือการดึงคุณลักษณะ (Feature) ของเอกสารออกมา กับการลดขนาดเอกสารลง ซึ่งการดึงคุณลักษณะออกมานั้น ก่อนอื่นเราต้องการกำหนดก่อนว่าจะใช้อะไร เป็นตัวแทนคุณลักษณะของเอกสาร และใช้คำใดแทนคุณลักษณะเอกสารนั้น จากการสำรวจงานวิจัยที่ผ่านมาทั้งในประเทศและต่างประเทศพบว่า ส่วนใหญ่จะใช้คำเดี่ยวเป็นตัวแทนคุณลักษณะของเอกสาร นอกจากนั้น ยังสามารถใช้ วลี หรือกลุ่มของคำ ประโยคฯลฯ แทนคุณลักษณะของเอกสารได้เช่นกัน ตัวแทนคุณลักษณะที่นิยมใช้ในการจัดหมวดหมู่เอกสารประเภทข้อความคือ ถุงคำ (Bag of words) ซึ่งจะเก็บอยู่ในรูปแบบของเวกเตอร์ โดยองค์ประกอบของเวกเตอร์อาจจะแทนด้วยค่าความจริง (Boolean features) หรือ ค่าความถี่ของคำ เป็นต้น [3-6]

2.2 การตัดคำ (Word Segmentation)

เนื่องจากข้อความในภาษาไทยมีการเขียนติดกัน ไม่มีช่องว่างเหมือนภาษาต่างประเทศ จึงมีผู้คิดค้นพัฒนาการตัดคำ (Word Segmentation) ในภาษาไทย ซึ่งมีวิธีการตัดคำแบ่งได้เป็น หลักการตัดคำโดยใช้กฎ (Rule Base Approach) หลักการตัดคำโดยใช้อัลกอริทึม (Algorithm Approach) หลักการตัดคำโดยใช้พจนานุกรม (Dictionary Approach) และหลักการตัดคำโดยใช้คลังข้อมูล (Corpus Base Approach) แต่ละวิธีการต่างๆ ก็ให้ผลในด้านความถูกต้อง ความรวดเร็วของการทำงานและปริมาณการใช้ทรัพยากรต่างๆ ที่แตกต่างกัน จากการศึกษาเรื่องตัดคำสำหรับการจัดหมวดหมู่เอกสารภาษาไทย พบปัญหาด้านการหาขอบเขตของคำ เนื่องจากไม่มีการเขียนแบ่งพยางค์ คำ หรือประโยค ไม่มีหลักเกณฑ์ตายตัวในการใช้ช่องว่างในภาษาเขียน การสะกดคำมีรูปแบบซับซ้อน มีคำ

ยืม คำทับศัพท์ คำเฉพาะจำนวนมากและคำมีความกำกวมสูง จากการศึกษาเปรียบเทียบประสิทธิภาพวิธีดังกล่าว พบว่าวิธีตัดคำที่เหมาะสมกับการจัดหมวดหมู่เอกสารคือวิธีการตัดคำแบบยาวที่สุด (Longest Matching) [4]

2.3 การกำจัดคำหยุด (Stop-Word List Removal)

เป็นการนำคำที่ไม่มีนัยสำคัญออกโดยที่ไม่ทำให้ความหมายของเอกสารเปลี่ยนแปลง คำที่ไม่มีนัยสำคัญ ในที่นี้หมายถึงคำที่ใช้กันโดยทั่วไปไม่มีความหมายสำคัญต่อเอกสาร เมื่อตัดออกจากเอกสารแล้วไม่ทำให้ใจความของเอกสารเปลี่ยนแปลง ตัวอย่างเช่น คำบุพบท เป็นคำที่ใช้เชื่อมคำหรือกลุ่มคำให้สัมพันธ์กัน คำสันธานเป็นคำที่ทำหน้าที่เชื่อมคำกับคำ คำสรรพนามเป็นคำที่ใช้แทนคำนามที่กล่าวถึงมาแล้วในประโยค เป็นต้น จึงถือได้ว่าคำหยุดเป็นคุณลักษณะที่ไม่เกี่ยวข้องหรือไม่มีประโยชน์ในการค้นคืนหรือการจำแนกหมวดหมู่ ดังนั้นการกำจัดคำหยุดจึงเป็นกระบวนการที่ควรทำก่อนการจัดทำดัชนี เพื่อกำจัดคุณลักษณะที่ไม่เป็นประโยชน์และลดขนาดของดัชนีลง ซึ่งจะช่วยประหยัดทั้งพื้นที่และเวลาในการประมวลผล [5-7]

2.4 การหารากศัพท์ (Stemming)

เป็นการหารูปเดิมของคำ หรือหาคำที่มีความหมายคล้ายกัน เพื่อปรับรวมให้เป็นคำเดียวกัน การหารากศัพท์เป็นกระบวนการที่ควรทำก่อนการจัดทำดัชนี ทำให้สามารถลดขนาดของดัชนีลงและเพิ่มประสิทธิภาพในการค้นคืนหรือการจำแนกหมวดหมู่ การหารากศัพท์ของคำภาษาไทยนั้นจะใช้วิธีการรวบรวมคำศัพท์ที่มีความหมายคล้ายกัน หรือมีรากศัพท์เดียวกันไว้เป็นรายการคำศัพท์ เพื่อใช้ในการเปรียบเทียบหารากศัพท์ วิธีการนี้ต้องอาศัยผู้เชี่ยวชาญทางภาษาและใช้เวลาในการเก็บรวบรวมและจัดทำรายการคำศัพท์ [8-9]

2.5 การสร้างดัชนี (indexing)

เนื่องจากคอมพิวเตอร์ไม่สามารถจำแนกหมวดหมู่ของเอกสารซึ่งเป็นภาษาธรรมชาติโดยตรงได้ดังนั้นจึงต้องแปลงเอกสารให้อยู่ในรูปแบบ ที่คอมพิวเตอร์สามารถใช้ในการเรียนรู้ได้ ขั้นตอนในการแปลงเอกสาร เรียกว่า การทำดัชนี (Indexing) เพื่อสร้างตัวแทนเนื้อหาของเอกสาร (Document Representation) สำหรับใช้ในกระบวนการเรียนรู้ วัตถุประสงค์ของการสร้างดัชนีคือ การคำนวณหาค่าที่จะมาใช้เป็นค่าคุณลักษณะของเอกสาร หรืออาจจะเรียกได้ว่าการหาค่าน้ำหนัก (Term weighting) การสร้างดัชนี โดยทั่วไปที่นิยมใช้กัน จะเริ่มจากการสร้างเวกเตอร์ตัวแทนเอกสาร จากนั้นจะสร้างเมตริกซ์ของกลุ่มเอกสารขึ้นจากเวกเตอร์เอกสารทั้งหมดในกลุ่ม [3,6] ซึ่งงานวิจัยนี้ใช้การทดลองคำนวณค่าน้ำหนักให้กับดัชนีดังต่อไปนี้ โดยให้

f_{ik} = เป็นความถี่ของคำ i ในเอกสาร k

N = จำนวนเอกสารทั้งหมดรวมของทุกกลุ่ม

M = จำนวนของคำทั้งหมดรวมในทุกกลุ่ม

n_i = จำนวนเอกสารทั้งหมดที่มีคำ i เกิดขึ้น

2.5.1 Boolean-weighting (binary)

ค่านี้จะพิจารณาจากการมีอยู่ของคำในแต่ละเอกสาร ถ้ามีคำปรากฏความถี่มากกว่า หรือเท่ากับ 1 ก็จะได้ค่าเป็น 1 แต่ถ้าไม่มีคำนั้น ๆ ปรากฏอยู่เลยจะมีค่าเท่ากับ 0 เรียกอีกอย่างหนึ่งว่าค่าคุณลักษณะความจริง (Boolean Feature) [3]

$$a_{ik} = \begin{cases} 1 & \text{if } f_{ik} > 0 \\ 0 & \text{otherwise} \end{cases}$$

2.5.2 Term frequency-weighting (tf)

ค่านี้จะพิจารณาจากความถี่ของคำที่ปรากฏในแต่ละเอกสารโดยตรง ถ้าคำใดมีความถี่มากก็จะได้ค่าน้ำหนักที่มีค่าสูง[3]

$$a_{ik} = f_{ik}$$

2.5.3 Term frequency–inverse document frequency -weighting (tfidf)

ค่านี้จะพิจารณาจากความถี่ของคำในเอกสาร คูณกับฟังก์ชัน log ของเอกสารทั้งหมด หารด้วยจำนวนเอกสารที่ปรากฏคำนั้นอยู่ กล่าวคือถ้าคำไหนมีอยู่ในทุกเอกสาร ก็จะทำให้ค่า log ของจำนวนเอกสารทั้งหมดหารด้วยจำนวนเอกสารที่ปรากฏคำนั้นอยู่มีค่าเท่ากับ 0 ทำให้ได้ค่าน้ำหนักเท่ากับ 0 ซึ่งเป็นวิธีการให้น้ำหนักแบบมาตรฐานที่ได้รับความนิยม [10]

$$a_{ik} = f_{ik} * \log\left(\frac{N}{n_i}\right)$$

2.5.4 Term frequency component-weighting (tfc)

ค่านี้จะพิจารณาถึงความแตกต่างในเรื่องของความยาวของเอกสารร่วมการให้น้ำหนักด้วย ดังนั้น ค่า tfc นี้จึงได้ปรับบรรทัดฐานในเรื่องของความยาวของเอกสารที่แตกต่างกันด้วย [11]

$$a_{ik} = \frac{f_{ik} * \log\left(\frac{N}{n_i}\right)}{\sqrt{\sum_{j=1}^{M}\left[f_{jk} * \log\left(\frac{N}{n_j}\right)\right]^2}}$$

2.5.5 Log-weighted term frequency-weighting (ltc)

ค่านี้จะพิจารณาถึงการเพิ่มฟังก์ชันlog เข้ามากับค่าความถี่ เพื่อลดความแตกต่างกันของความถี่ของคำ ในกรณีที่ความถี่มีความแตกต่างกันมาก [12]

$$a_{ik} = \frac{\log(f_{ik} + 1.0) * \log\left(\frac{N}{n_i}\right)}{\sqrt{\sum_{j=1}^{M}\left[\log(f_{jk} + 1.0) * \log\left(\frac{N}{n_j}\right)\right]^2}}$$

2.5.6 Entropy weighting (entropy)

ค่านี้เป็นการวัดค่าเอนโทรปีของคำ หรือเป็นความไม่แน่นอนของคำ โดยถ้าคำไหนปรากฏอยู่บนทุกเอกสาร ค่าความไม่แน่นอนที่ได้จะได้เท่ากับ -1 แต่ถ้าคำนั้นปรากฏอยู่เพียงแค่หนึ่งเอกสาร ค่าความไม่แน่นอนจะเท่ากับ 0 [13]

$$a_{ik} = \log(f_{ik} + 1.0) * \left(1 + \frac{1}{\log(N)} \sum_{j=1}^{N} \left[\frac{f_{ij}}{n_i} \log\left(\frac{f_{ij}}{n_i}\right)\right]\right)$$

2.6 การเลือกคุณลักษณะ (Feature Selection)

จากการศึกษาพบว่าคุณลักษณะที่สกัดมาได้จากเอกสารภาษาไทยนั้น มีจำนวนคุณลักษณะจำนวนมาก ทำให้วิธีลดขนาดของคุณลักษณะเบื้องต้นด้วยวิธีนำคำที่ไม่มีนัยสำคัญออก กับการทำรากศัพท์แล้วยังไม่เพียงพอ ซึ่งจำนวนคุณลักษณะที่มากนั้นส่งผลต่อประสิทธิภาพของการจำแนกหมวดหมู่เอกสาร เนื่องจากอัลกอริทึมที่ใช้ในการเรียนรู้เพื่อสร้างตัวจำแนกหมวดหมู่ โดยทั่วไปไม่สามารถรองรับการทำงานกับจำนวนคุณลักษณะของเอกสารที่จำนวนมากได้ดี จึงเกิดกระบวนการลดขนาดเอกสารโดยการเลือกคุณลักษณะที่ดีและสัมพันธ์กับกลุ่ม มาทำการสร้างตัวจำแนกเอกสาร งานวิจัยนี้ใช้ค่าสารสนเทศ (IG: Information Gain) ในการคัดเลือกคุณลักษณะของเอกสาร ซึ่งค่า *IG* จะคำนวณจากจำนวนบิตที่ได้รับสำหรับการทำนายกลุ่มโดยการดูจากการมีอยู่หรือไม่มีอยู่ของคำในเอกสารให้ $C_1, ..., C_K$ แทนเซ็ตที่เป็นไปได้ของกลุ่ม ค่า *IG* ของคำ w นิยามโดย [6,9]

$$IG(w) = -\sum_{j=1}^{K} P(c_j)\log P(c_j) + P(w)\sum_{j=1}^{K} P(c_j \mid w)\log P(c_j \mid w) + P(\overline{w})\sum_{j=1}^{K} P(c_j \mid \overline{w})\log P(c_j \mid \overline{w})$$

ค่า $P(C_j)$ คำนวณได้จาก เศษส่วนของจำนวนเอกสารที่อยู่กลุ่ม C_j กับ จำนวนเอกสารทั้งหมด

ค่า $P(w)$ คำนวณได้จาก เศษส่วนของจำนวนเอกสารที่มีคำ w กับจำนวนเอกสารทั้งหมด

ค่า $P(C_j \mid w)$ คำนวณได้จาก เศษส่วนของคำจำนวนเอกกลุ่ม C_j ที่มีคำ w กับเอกสารทั้งหมด

ค่า $P(C_j \mid \overline{w})$ คำนวณได้จาก เศษส่วนของคำจำนวนเอกกลุ่ม C_j ที่ไม่มีคำ w กับเอกสารทั้งหมด

2.7 อัลกอริทึมการจัดหมวดหมู่ (Classifier Algorithm)

อัลกอริทึมในการจัดหมวดหมู่การเรียนรู้แบบมีผลเฉลย (Supervised Learning) สามารถแบ่งขั้นตอนวิธีการจัดหมวดหมู่เอกสารแบ่งได้เป็น 2 ขั้นตอนคือ การเรียนรู้เพื่อสร้างกลุ่มเอกสารต้นแบบและจัดหมวดหมู่ของเอกสารที่สนใจ โดยการตรวจสอบหาความคล้ายกับกลุ่มเอกสารต้นแบบ [1-3,14]

2.7.1 ต้นไม้ตัดสินใจ (Decision Tree) ต้นไม้จะประกอบด้วยโหนดแทนคุณลักษณะ และโหนดล่างสุดแทนหมวดหมู่ การสร้างกิ่งสาขาจะพิจารณาจากค่าความจริงของคุณลักษณะ โดยค่าที่ใช้จะมาจากการคำนวณจากค่า Information Gain การสร้างต้นไม้ตัดสินใจ จะใช้ค่ามาตรฐาน

อัตราส่วนเกน (Gain Ratio) เพื่อเลือกคุณลักษณะที่จะใช้เป็นรากหรือโหนด ถ้าให้ชุดข้อมูล M ประกอบด้วยค่าที่เป็นไปได้ คือ $\{m_1, m_2, \ldots, m_n\}$ และให้ความน่าจะเป็นที่จะเกิดค่า m_i มีค่าเท่ากับ $P(m_i)$ จะได้ว่าค่าเกนสารสนเทศ (Information Gain) ของ M เขียนแทนด้วย I(M) คำนวณได้ดังสมการ

$$I(M) = \sum_{i=1}^{n} -P(m_i)\log_2 P(m_i)$$

ถ้าให้ข้อมูลสอน คือ T และคุณลักษณะที่เป็นโหนด คือ x และมีค่าทั้งหมดที่เป็นไปได้ n ค่า โหนดปัจจุบันจะแบ่งตัวอย่าง T ออกตามกิ่งเป็น $\{t_1, t_2, \ldots, t_n\}$ ตามค่าที่เป็นไปได้ของ x ดังนั้นจึงสามารถคำนวณค่าเกนสารสนเทศหลังจากแบ่งตามคุณลักษณะ x ได้ดังสมการ

$$I_x(T) = \sum_{i=1}^{n} \frac{|t_i|}{|T|} I(t_i)$$

ค่ามาตรฐานเกน (GAIN) ของคุณลักษณะ x ได้ดังสมการ

$$Gain(x) = I(T) - I_x(T)$$

จากนั้นคำนวณค่าสารสนเทศของการแบ่งแยก (Split Information) ของคุณลักษณะแต่ละตัว ถ้าให้ T คือ ชุดของตัวอย่าง เมื่อแบ่งตัวอย่างนี้ตามคุณลักษณะ x จะได้ชุดของตัวอย่างย่อยในแต่ละกิ่ง คำนวนค่ามาตรฐานอัตราส่วนเกน (Gain ratio) ท้ายสุดจึงเลือกค่า Gain ratio สูงสุดเป็นคุณลักษณะเริ่มต้น และเลือกคุณสมบัติถัดไปตามค่า Gain ratio น้อยลงตามลำดับ

2.7.2 ซัพพอร์ทเวกเตอร์แมชชีน (Support Vector Machine) แนวคิดหลักของวิธีการนี้ ใช้เพื่อหาระนาบการตัดสินใจในการแบ่งข้อมูลออกเป็นสองส่วน โดยใช้สมการเส้นตรงเพื่อแบ่งเขตข้อมูล 2 กลุ่มออกจากกัน โดยจะพยายามสร้างเส้นแบ่งตรงกึ่งกลางระหว่างกลุ่มให้มี ระยะห่างระหว่างขอบเขตของทั้งสองกลุ่มมากที่สุด SVM จะใช้ฟังก์ชั่นแม็ปสำหรับย้ายข้อมูลจาก Input Space ไปยัง Feature Space และสร้างฟังก์ชั่นวัดความคล้ายที่เรียกว่าเคอร์เนลฟังก์ชั่น (Kernel Function) บน Feature Space เหมาะใช้สำหรับข้อมูลที่มีมิติของข้อมูลสูง กำหนดให้ $(x_i,y_i),\ldots,(x_n,y_n)$ เป็นตัวอย่างที่ใช้สำหรับการสอน n คือ จำนวนข้อมูลตัวอย่าง m คือ จำนวนมิติข้อมูลเข้า และ y คือ ผลลัพธ์มีค่า + 1 หรือ -1 ดังสมการ

$$(x_i,y_i),\ldots,(x_n,y_n) \text{ เมื่อ } x \in R^m, y \in \{+1,-1\}$$

สำหรับปัญหาเชิงเส้น มิติข้อมูลขนาดสูงได้ถูกแบ่งเป็น 2 กลุ่ม โดยระนาบตัดสินใจ ได้ดังสมการ

$$(w*x)+b = 0$$

เมื่อ w คือ ค่าน้ำหนักและ b คือค่า bias สมการ ใช้สำหรับจำแนกประเภทของข้อมูล

$(w*x)+b>0$ ถ้า $y_i = +1$ และ $(w*x)+b<0$ ถ้า $y_i = -1$

ซึ่ง SVM มีเคอร์เนลฟังก์ชั่น (Kernel Function) ให้ผู้ใช้สามารถประยุกต์ใช้ในการแก้ปัญหาได้หลายวิธี เช่น Linear, Polynomial, Radial Basis Function เป็นต้น

2.7.3 เนอีฟเบย์ (Naïve-Bayes) หลักการของวิธีการนี้ใช้การคำนวณความน่าจะเป็นซึ่งถูกใช้ใน การทำนายผล Naïve-Bayes เป็นเทคนิคในการแก้ปัญหาแบบ classification ที่สามารถคาดการณ์ผลลัพธ์ได้และ สามารถอธิบายได้ด้วย มันจะทำการวิเคราะห์ความสัมพันธ์ระหว่างตัวแปร เพื่อใช้ในการสร้างเงื่อนไขความน่าจะเป็น สำหรับแต่ละความสัมพันธ์ การเรียนรู้เบย์อย่างง่ายเป็นวิธีจำแนกประเภทข้อมูลที่มีประสิทธิภาพวิธีหนึ่ง โดยที่ใช้ในงานจัดหมวดหมู่เอกสาร ข้อความ (Text Classification) ได้ดี อัลกอริทึมในการทำงานที่ไม่ซับซ้อน เหมาะกับกรณีของเซตตัวอย่างมีจำนวนมากและคุณสมบัติ (Attribute) ของตัวอย่างไม่ขึ้นต่อกัน โดยกำหนดให้ความน่าจะเป็นของข้อมูลที่จะเป็น

$$P(a_1, a_2, \cdots, a_n \mid v_j) = \prod_{i=1}^{n} P(a_i \mid v_j)$$

กลุ่ม v_j สำหรับข้อมูลที่มีคุณสมบัติ n ตัว X={ a_1, a_2, …, a_n } หรือ ใช้สัญลักษณ์ว่า P(a_1, a_2, …, a_n | v_j) โดยที่ $\prod$ หมายถึง ผลคูณของค่า P(a_i | v_j) ทั้งหมด i = 1,2,3,…,n และ j = 1,2,3,…,n ดังนั้นเราจะได้ ว่าวิธีการจำแนกประเภทแบบเบย์อย่างง่าย ดังสมการ

$$v_{NB} = \arg\max_{v_j \in V} P(v_j) \times \prod_{i=1}^{n} P(a_i \mid v_j)$$

3. วิธีการดำเนินการวิจัย

งานวิจัยนี้ทำการทดลอง เปรียบเทียบการวิธีการคำนวณค่าน้ำหนักให้กับดัชนีแบบต่างๆ ร่วมกับอัลกอริทึมการจัดหมวดหมู่เอกสารภาษาไทย โดยทดสอบกับเอกสารประเภทข่าวอิเล็กทรอนิกส์จากหนังสือพิมพ์ไทยรัฐ จำนวน 10 กลุ่ม ได้แก่ กลุ่มการศึกษา กลุ่มบันเทิง กลุ่มสังคม กลุ่มการเมือง กลุ่มเทคโนโลยี กลุ่มกีฬา กลุ่มข่าวต่างประเทศ กลุ่มเกษตร กลุ่มเศรษฐกิจ กลุ่มวัฒนธรรม โดยมีจำนวนกลุ่มตัวอย่างทั้งหมด 2000 เอกสาร เป็นกลุ่มตัวอย่างเรียนรู้ และ ทำการทดสอบด้วยวิธี 10-fold cross validation โดยแบบจำลองที่นำเสนอในงานวิจัยนี้ ใช้คุณลักษณะแบบคำเดียว (Single word) ที่ได้จากการตัดคำด้วยวิธีการตัดคำแบบยาวที่สุด (Longest Matching) โดยใช้พจนานุกรมฉบับ Lexitron เป็นตัวเปรียบเทียบและทำการกำจัดคำหยุดและทำรากศัพท์จากฐานข้อมูลที่กำหนดขึ้น [7] หลังจากนั้นลดขนาดคุณลักษณะ โดยทำการคัดเลือกคุณลักษณะที่มีค่า IG สูงสุดที่ระดับต่างๆ [9] มาส่งเข้าเครื่องจักรการเรียนรู้แบบมีผลเฉลยด้วยอัลกอริทึม ต้นไม้ตัดสินใจ (Decision Tree) เนอีฟเบย์ (Naïve-Bayes) ซัพพอร์ทเวกเตอร์แมชชีน (Support Vector Machine) เคอร์เนลฟังก์ชั่นแบบ Linear parameter C = 1 gamma = 0 (LibSVM Multiclass) โดยอัลกอริทึมทั้งหมดใช้ค่าพารามิเตอร์มาตรฐานของโปรแกรม Weka [15] มาทำการเรียนรู้ แล้วทำการทดสอบเปรียบเทียบ

ประสิทธิภาพด้านความถูกต้อง ความแม่นยำ โดยใช้วิธีการประเมินความสามารถของแบบจำลอง โดยวัดที่ประสิทธิผลของการจำแนกหมวดหมู่ตามแนวคิดทางด้านการค้นคืนสารสนเทศ ซึ่งก็คือการวัดค่า F-Measurement ซึ่งคำนวณได้ดังสมการ [8]

$$recall = \frac{a}{a+c} \qquad F\text{-}measure = \frac{2 \times (Precision \times Recall)}{Precision + Recall}$$

$$precision = \frac{a}{a+b}$$

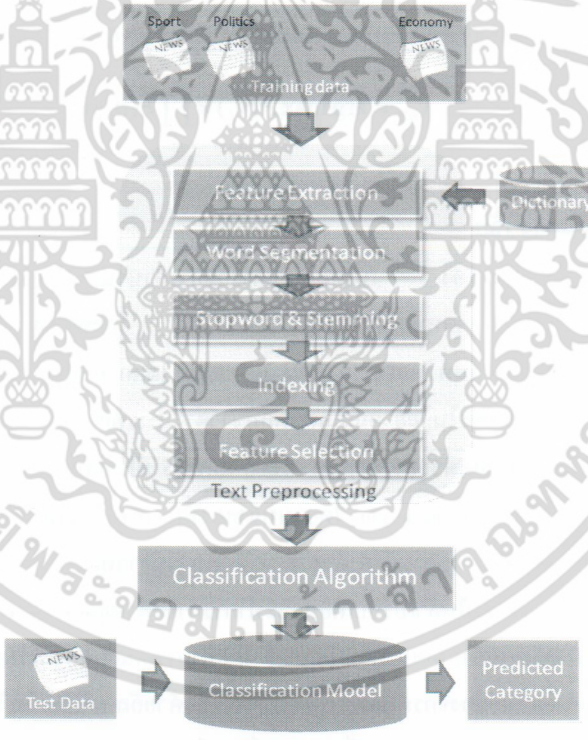
โดยให้ a = จำนวนเอกสารที่อยู่ในหมวดหมู่ C_j และตัวจำแนกทำนายว่าอยู่ในหมวดหมู่ C_j

b = จำนวนเอกสารที่ไม่อยู่ในหมวดหมู่ C_j และตัวจำแนกทำนายว่าอยู่ในหมวดหมู่ C_j

c = จำนวนเอกสารที่อยู่ในหมวดหมู่ C_j และตัวจำแนกทำนายว่าไม่อยู่ในหมวดหมู่ C_j

d = จำนวนเอกสารที่ไม่อยู่ในหมวดหมู่ C_j และตัวจำแนกทำนายว่าไม่อยู่ในหมวดหมู่ C_j

C_j = กลุ่มประเภทของเอกสารที่สนใจวัดประสิทธิภาพ



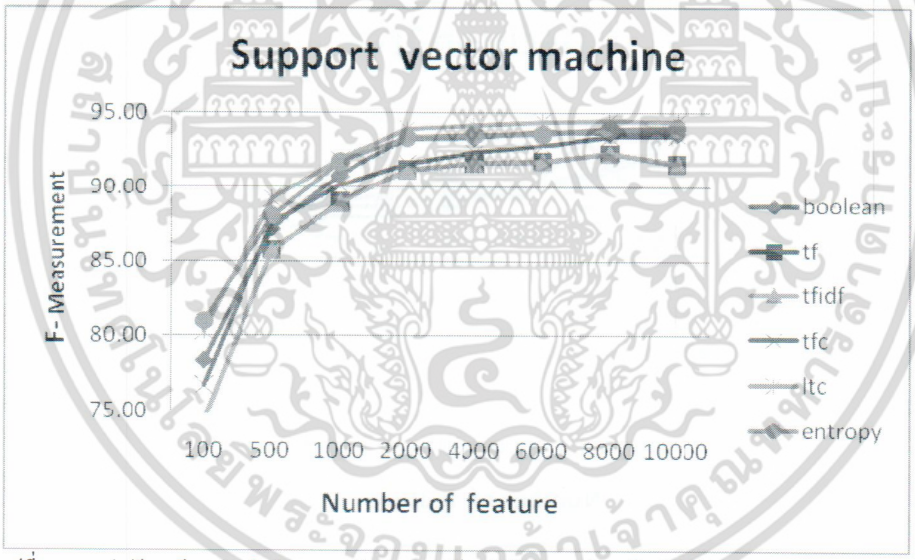
รูปที่ 3.1 แบบจำลองการจัดหมวดหมู่เอกสาร

4. ผลการทดลอง

การทดลองคำนวณค่าน้ำหนักให้กับดัชนีแบบต่างๆ ร่วมกับอัลกอริทึมการจัดหมวดหมู่เอกสารซึ่งประกอบด้วยอัลกอริทึมต้นไม้ตัดสินใจ (Decision Tree) เนอีฟเบย์ (Naïve-Bayes) ซัพพอร์ทเวกเตอร์แมชชีน (Support Vector Machine) โดยทดลองกับกลุ่มตัวอย่างเอกสารข่าวภาษาไทยจำนวน 10 ประเภท จำนวน 2000 เอกสาร ซึ่งมีการกระจายตัวของกลุ่มตัวอย่างเท่ากันคือประเภทละ 200 เอกสาร ซึ่งได้ผลการทดลองดังนี้

ตารางที่ 4.1 ผลการทดลองค่าน้ำหนักให้กับดัชนีแบบต่างๆ โดยใช้วิธี Support vector machine

	F-Measurement / Support Vector Machine					
feature	**boolean**	**tf**	**tfidf**	**tfc**	**ltc**	**entropy**
100	78.40	74.35	74.35	76.70	80.25	80.95
500	87.25	85.80	85.75	87.65	89.25	88.10
1000	90.85	89.00	88.95	90.15	91.85	91.70
2000	93.30	91.15	91.15	91.55	93.95	93.40
4000	93.40	91.65	91.65	92.35	94.20	93.55
6000	93.75	91.75	91.75	92.80	94.45	93.60
8000	93.70	92.30	92.30	93.45	***94.50***	93.95
10000	93.70	91.55	91.55	93.50	94.40	94.00
average	**90.54**	**88.44**	**88.43**	**89.77**	**91.61**	**91.16**



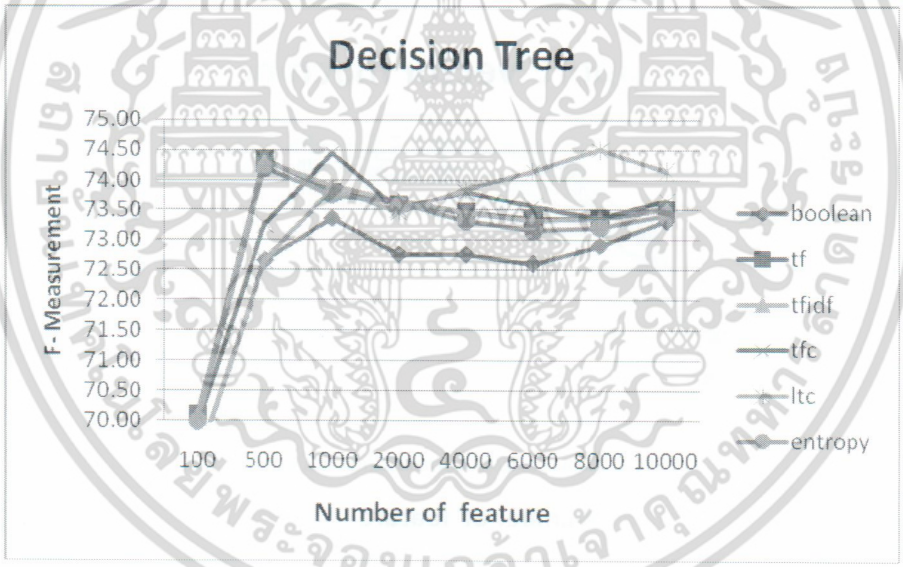
รูปที่ 4.1 กราฟเปรียบเทียบประสิทธิภาพการจัดหมวดหมู่ด้วยค่าดัชนีแบบต่างๆ โดยใช้วิธี Support vector machine

จากการทดลองให้ค่าน้ำหนักให้กับดัชนีแบบต่างๆ และทำการเรียนรู้ด้วยอัลกอริทึม Support Vector Machine โดยวัดจากค่า F-Measurement สามารถสรุปได้ว่า การคำนวณค่าน้ำหนักเพื่อสร้างดัชนีด้วยวิธี ltc weighting ให้ประสิทธิภาพการจัดหมวดหมู่โดยเฉลี่ยออกมาดีที่สุด คือ 91.61 %

รองลงมาเป็นวิธี entropy weighting ให้ประสิทธิภาพโดยเฉลี่ย 91.16 % และ boolean weighting ให้ประสิทธิภาพโดยเฉลี่ย 90.54 % เมื่อพิจารณาจากกราฟพบว่าวิธี ltc weighting ให้ประสิทธิภาพในการจัดหมวดหมู่เอกสารเหนือกว่าทุกวิธีที่ทำการทดลองจนถึงระดับ 500 คุณลักษณะ และเมื่อพิจารณาพารามิเตอร์ที่ส่งผลให้ประสิทธิภาพด้านการจัดหมวดหมู่เอกสารภาษาไทยดีที่สุด ของค่าน้ำหนักทุกวิธี ด้วยอัลกอริทึม Support Vector Machine พบว่า วิธี ltc weighting ที่จำนวน 8000 คุณลักษณะ ให้ประสิทธิภาพดีสุดที่ระดับ 94.50 %

ตารางที่ 4.2 ผลการทดลองค่าน้ำหนักให้กับดัชนีแบบต่างๆ โดยใช้วิธี Decision Tree

	F-Measurement / Decision Tree					
feature	**boolean**	**tf**	**tfidf**	**tfc**	**ltc**	**entropy**
100	69.30	70.10	70.05	69.95	69.35	70.00
500	72.65	74.35	74.30	73.25	72.65	74.20
1000	73.35	73.80	73.90	74.45	73.85	73.75
2000	72.75	73.60	73.65	73.50	73.45	73.60
4000	72.75	73.45	73.45	73.80	73.85	73.30
6000	72.60	73.35	73.30	73.55	74.15	73.15
8000	72.90	73.35	73.35	73.35	***74.50***	73.20
10000	73.30	73.50	73.50	73.65	74.15	73.40
average	**72.45**	**73.19**	**73.19**	**73.19**	**73.24**	**73.08**



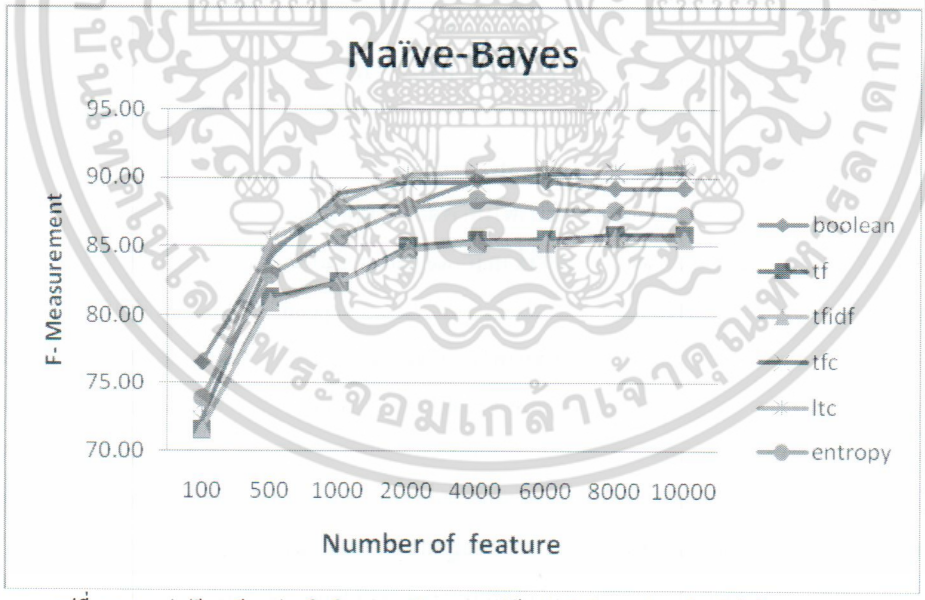
รูปที่ 4.2 กราฟเปรียบเทียบประสิทธิภาพการจัดหมวดหมู่ด้วยค่าดัชนีแบบต่างๆ โดยใช้วิธี **Decision Tree**

จากการทดลองให้ค่าน้ำหนักให้กับดัชนีแบบต่างๆ และทำการเรียนรู้ด้วยอัลกอริทึม Decision Tree โดยวัดจากค่า F-Measurement สามารถสรุปได้ว่า การคำนวณค่าน้ำหนักเพื่อสร้างดัชนีด้วยวิธี ltc weighting ให้ประสิทธิภาพการจัดหมวดหมู่โดยเฉลี่ยออกมาดีที่สุด คือ 73.24 % รองลงมาเป็นวิธี

tfc , tfidf ,tf weighting ให้ประสิทธิภาพโดยเฉลี่ย 73.19 % ทั้ง 3 วิธีเท่ากัน และถัดมาเป็น entropy weighting ให้ประสิทธิภาพโดยเฉลี่ย 73.08 % จากการสังเกตพบว่าขั้นตอนการคำนวณค่าน้ำหนักของวิธีต่างๆเหล่านี้ให้ผลด้านประสิทธิภาพการจัดหมวดหมู่กับอัลกอริทึม Decision Tree ไม่แตกต่างกันมากนัก เมื่อพิจารณาจากกราฟพบว่าวิธี ltc weighting ให้ประสิทธิภาพในการจัดหมวดหมู่เอกสารเหนือกว่าทุกวิธีที่ทำการทดลองทุกระดับเมื่อลดคุณลักษณะมาถึง 4000 คุณลักษณะ แต่ถ้าคุณลักษณะต่ำกว่านั้นวิธีอื่นจะให้ประสิทธิภาพที่ดีกว่า และเมื่อพิจารณาพารามิเตอร์ที่ส่งผลให้ประสิทธิภาพด้านการจัดหมวดหมู่เอกสารภาษาไทยดีที่สุดของค่าน้ำหนักทุกวิธี ด้วยอัลกอริทึม Decision Tree พบว่าวิธี ltc weighting ที่จำนวน 8000 คุณลักษณะ ให้ประสิทธิภาพดีสุดที่ระดับ 74.50 %

ตารางที่ 4.3 ผลการทดลองค่าน้ำหนักให้กับดัชนีแบบต่างๆ โดยใช้วิธี Naïve-Bayes

	F-Measurement / Naïve-Bayes					
feature	**boolean**	**tf**	**tfidf**	**tfc**	**ltc**	**entropy**
100	76.60	71.55	71.60	72.30	73.50	73.90
500	84.70	81.25	80.90	84.30	85.45	82.95
1000	87.85	82.45	82.40	88.85	88.15	85.65
2000	88.00	85.00	84.75	89.70	90.20	87.85
4000	89.85	85.45	85.25	89.65	90.50	88.45
6000	89.75	85.50	85.25	90.25	90.75	87.70
8000	89.25	85.85	85.60	90.45	90.40	87.65
10000	89.25	85.90	85.50	90.40	***90.85***	87.30
average	**86.91**	**82.87**	**82.66**	**86.99**	**87.48**	**85.18**



รูปที่ 4.3 กราฟเปรียบเทียบประสิทธิภาพการจัดหมวดหมู่ด้วยค่าดัชนีแบบต่างๆ โดยใช้วิธี **Naïve-Bayes**

จากการทดลองให้ค่าน้ำหนักให้กับดัชนีแบบต่างๆ และทำการเรียนรู้ด้วยอัลกอริทึม Naïve-Bayes โดยวัดจากค่า F-Measurement สามารถสรุปได้ว่า การคำนวณค่าน้ำหนักเพื่อสร้างดัชนีด้วยวิธี ltc weighting ให้ประสิทธิภาพการจัดหมวดหมู่โดยเฉลี่ยออกมาดีที่สุด คือ 87.48 % รองลงมาเป็นวิธี tfc weighting ให้ประสิทธิภาพโดยเฉลี่ย 86.99 % และ boolean weighting ให้ประสิทธิภาพโดยเฉลี่ย 86.91 % เมื่อพิจารณาจากกราฟพบว่าวิธี ltc weighting ให้ประสิทธิภาพในการจัดหมวดหมู่เอกสารเหนือกว่าทุกวิธีที่ทำการทดลองทุกระดับจนถึงระดับ 2000 คุณลักษณะ ยกเว้นที่ระดับ 8000 คุณลักษณะที่ tfc weighting ดีกว่าเล็กน้อย และเมื่อพิจารณาพารามิเตอร์ที่ส่งผลให้ประสิทธิภาพด้านการจัดหมวดหมู่เอกสารภาษาไทยดีที่สุดของค่าน้ำหนักทุกวิธี ด้วยอัลกอริทึม Naïve-Bayes พบว่าวิธี ltc weighting ที่จำนวน 10000 คุณลักษณะ ให้ประสิทธิภาพดีสุดที่ระดับ 90.85 %

5. สรุปผลและข้อเสนอแนะ

งานวิจัยนี้ได้นำเสนอวิธีการคำนวณค่าน้ำหนักให้กับดัชนีแบบต่างๆ และทำการลดคุณลักษณะโดยใช้วิธี Information Gain หลังจากนั้นทำการเรียนรู้ด้วยอัลกอริทึมเครื่องจักรการเรียนรู้ทั้ง 3 วิธี โดยทำการศึกษาเปรียบเทียบวิธีการคำนวณค่าน้ำหนักที่เหมาะสมและมีประสิทธิภาพดีที่สุดในการจัดหมู่เอกสารข่าวภาษาไทย จากการทดลองพบว่าการคำนวณค่าน้ำหนักให้กับดัชนีด้วยวิธี ltc weighting เมื่อพิจารณาจากค่าเฉลี่ยให้ประสิทธิภาพในการจัดหมวดหมู่เอกสารภาษาไทยดีที่สุด เมื่อเทียบกับการคำนวณน้ำหนักแบบอื่นๆในทุกอัลกอริทึม และเมื่อพิจารณาพารามิเตอร์ที่ส่งผลให้ประสิทธิภาพด้านการจัดหมวดหมู่เอกสารภาษาไทยดีที่สุด พบว่าอัลกอริทึม Support Vector Machine ร่วมกับวิธี ltc weighting ที่จำนวน 8000 คุณลักษณะ ให้ประสิทธิภาพดีสุดที่ระดับ 94.50 % รองลงมาเป็นอัลกอริทึม Naïve-Bayes ร่วมกับวิธี ltc weighting ที่จำนวน 10000 คุณลักษณะ ให้ประสิทธิภาพดีสุดที่ระดับ 90.85 % และสุดท้ายอัลกอริทึม Decision Tree ร่วมกับวิธี ltc weighting ที่จำนวน 8000 คุณลักษณะ ให้ประสิทธิภาพดีสุดที่ระดับ 74.50 % ตามลำดับ

เมื่อพิจารณาเปรียบเทียบประสิทธิภาพในการจัดหมวดหมู่เอกสาร ด้วยค่าเฉลี่ย F-Measurement ด้วยวิธี ltc weighting กับวิธี tfidf weighting ซึ่งเป็นวิธีมาตรฐานที่เป็นที่นิยมในการสร้างดัชนี [11] พบว่าการสร้างแบบจำลองด้วยอัลกอริทึม Support Vector Machine โดยใช้ค่าน้ำหนักวิธี ltc weighting ให้ประสิทธิภาพในการจัดหมวดหมู่เอกสารโดยเฉลี่ยสูงกว่าวิธี tfidf weighting สร้างดัชนีถึง 3.18 % เมื่อทดสอบกับแบบจำลองด้วยอัลกอริทึม Decision Tree วิธี ltc weighting ให้ประสิทธิภาพในการจัดหมวดหมู่เอกสารโดยเฉลี่ยสูงกว่าวิธี tfidf weighting เล็กน้อยที่ 0.05 % และเมื่อทดสอบกับแบบจำลองด้วยอัลกอริทึม Naïve-Bayes วิธี ltc weighting ให้ประสิทธิภาพในการจัดหมวดหมู่เอกสารโดยเฉลี่ยสูงกว่าวิธี tfidf weighting ถึง 4.82 %

เหตุผลที่งานวิจัยนี้ทำการศึกษาเปรียบเทียบการคำนวณค่าน้ำหนักให้กับดัชนีทั้ง 6 วิธีดังกล่าวมีแนวคิดมาจากการนำวิธีคำนวณค่าดัชนี ที่มีการใช้งานทางด้านการค้นคืนสารสนเทศ

(Information Retrieval) ในภาษาต่างประเทศ พบว่ามีประสิทธิภาพการค้นคืนในเกณฑ์ที่ดี มาประยุกต์ใช้ในการให้น้ำหนักดัชนีกับเอกสาร ในการสร้างแบบจำลองการจัดหมวดหมู่เอกสารภาษาไทย อีกทั้งวิธีการให้ค่าน้ำหนักดังกล่าวเป็นวิธีการที่เรียบง่าย ไม่ซับซ้อนและใช้เวลาการประมวลผลน้อย แต่สามารถจำแนกเอกสารได้เป็นอย่างดี ซึ่งจากเหตุผลดังกล่าว ทำให้ผู้วิจัยได้สร้างแบบจำลองเพื่อพัฒนาระบบจัดหมวดหมู่เอกสารภาษาไทยอัตโนมัติ (Thai Text Categorization) ที่มีประสิทธิภาพในอนาคต ทำให้ลดทรัพยากรแรงงานมนุษย์ในการแยกแยะเอกสารได้เป็นอย่างมาก นอกจากนั้นผลการทดลองที่ได้จากงานวิจัยนี้ยังสามารถนำวิธีการคำนวณค่าน้ำหนักให้กับดัชนีนี้ ไปประยุกต์กับงานด้านอื่นๆ เช่น การสร้างดัชนีให้กับการคัดกรองเอกสาร (Document Filtering) การจัดทำดัชนีอัตโนมัติเพื่อใช้ในการค้นคืนเอกสาร (Automatic Indexing for IR System) การสร้างดัชนีเพื่อจัดหมวดหมู่ของเว็บเพจ (Web Page Classification) เป็นต้น

กิตติกรรมประกาศ

งานวิจัยนี้ได้รับทุนอุดหนุนการทำวิจัยระดับบัณฑิตศึกษา มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ และขอขอบพระคุณ ดร.ชูชาติ หฤไชยะศักดิ์ ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ ที่อนุเคราะห์กลุ่มตัวอย่างที่ใช้ในการทดลองงานวิจัยนี้

เอกสารอ้างอิง

[1] Sebastiani, F. 2000. Machine Learning in Automated Text Categorization. ACM Computing Surveys (CSUR).

[2] Marquez, L. 2000. Machine learning and natural language processing. Technical Report Departament de Llenguatges Sistemes Informatics (LSI), Barcelona, Spain.

[3] Aas., E. 1999. Text Categorization: a Survey. Report Norwegian Computing Center.

[4] Charoenpornsawat, P. 1999. Feature-based Thai Word Segmentation. Master's Thesis. Computer Engineering. Chulalongkorn University, Bangkok, Thailand.

[5] Jaruskulchai, C. 1998. An Automatic Indexing for Thai Text Retrieval. PhD. Thesis, George Washington University. USA.

[6] วัลลภ อินทร์ฉ่ำ. 2548. ระบบการจัดหมวดหมู่เอกสารภาษาไทยอัตโนมัติโดยใช้ SVM ร่วมกับการประมวลผลภาษา, วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต มหาวิทยาลัยเกษตรศาสตร์.

[7] ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ. "ข้อมูลคำศัพท์ที่พบบ่อยจากฐานข้อมูลที่มาจากหนังสือพิมพ์" http://thailang.nectec.or.th/thaichar/word_thai.php

[8] Haruechaiyasak, C., Jitkrittum, W., Sangkeettrakarn, C., Damrongrat, C. 2008. Implementing News Article Category Browsing Based on Text Categorization Technique, International Conference on Web Intelligence and Intelligent Agent Technology.

[9] นิเวศ จิระวิชิตชัย ปริญญา สงวนสัตย์ และพยุง มีสัจ. 2552. การศึกษาทดลองเทคนิคการลดคุณลักษณะ และอัลกอริทึมการจัดหมวดหมู่ของเอกสารภาษาไทย . วารสารวิทยาศาสตร์ลาดกระบัง ปีที่ 18 ฉบับที่ 2

[10] Salton, G. and McGill, M.J. 1983. Introduction to Modern Information Retrieval, McGraw-Hill.

[11] Salton, G. and Buckley, C. 1988. Term-weighting approaches in automatic text retrieval, Information Processing and Management. 24(5), 513-523.

[12] Buckley, C., Salton, G., Allan, J. and Singhal, A. 1995. Automatic Query Expansion Using SMART : TREC 3, In Proc. of the Third Text Retrieval Conference. (TREC-3). NIST Special Publication, 500, 225.

[13] Dumais, S.T. 1991. Improving the retrieval of information from external sources. Behavior Research Methods, Instruments and Computers, 23, 229-236.

[14] พรพล ธรรมรงค์รัตน์, ลัดดา ปรีชาวีรกุล และวิภาดา เวทย์ประสิทธิ์. 2008. การจำแนกประเภทเว็บเพจโดยใช้ค่าความถี่เอกสารและซัพพอร์ตเวกเตอร์แมชชีน. The 12th National Computer Science and Engineering Conference.

[15] Witten, I.H. and Frank, E. 2005. Data Mining: Practical machine learning tools and techniques, 2nd Edition. Morgan Kaufmann, San Francisco.