บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในส่วนนี้ จะเป็นการอธิบายถึงแนวคิด ทฤษฎี และเทคนิคที่เกี่ยวข้อง ต่อการวิจัยและพัฒนากระบวนการของการจำแนกข้อความแสดงคิดเห็น โดยที่เอกสารเหล่านั้นมีลักษณะที่ขาดความสมดุลของเอกสาร และขาดความสมดุลคุณลักษณะในคลาส โดยแนวคิดและเทคนิคที่เกี่ยวข้องต่อไปนี้

2.1 ข้อมูลที่ไม่สมดุล (Imbalanced Data)

ข้อมูลที่ไม่สมดุล [2–5] หมายถึง ข้อมูลที่มีการกระจายตัวไม่เท่าเทียมกันในแต่ละกลุ่ม หรือข้อมูลซึ่งมีอัตราของสมาชิกกลุ่มหลัก (Majority) และกลุ่มรอง (Minority) มีจำนวนไม่เท่ากัน เช่น 1000:1 หรือ 10000:1 เป็นต้น ตัวอย่างเช่น สมมติว่ามีข้อมูลเกี่ยวกับผู้ป่วยโรคมะเร็งชนิดหนึ่ง โดยที่ข้อมูลผู้ป่วยที่ไม่เป็นมะเร็งเป็นข้อมูลกลุ่มหลัก อาจจะมีข้อมูลหลายหมื่นคน ในขณะที่ข้อมูลผู้ป่วยที่เป็นโรคมะเร็งเป็นกลุ่มข้อมูลกลุ่มรองอาจจะมีข้อมูลเพียงหลักร้อยคน

ในการจำแนกข้อมูล (Data Classification) หากนำข้อมูลทั้งสองมาสร้างโมเดลเพื่อการจำแนกเอกสาร จะมีความเป็นไปได้สูงว่าเมื่อสร้างโมเดสำหรับการจำแนกข้อมูลแล้ว ข้อมูลที่นำมาทดสอบมีโอกาสถูกทำนายเข้าเป็นกลุ่มไม่เป็นมะเร็ง เพราะจำนวนข้อมูลที่ใช้ในการสร้างโมเดลแล้ว ข้อมูลที่นำมาทดสอบมีโอกาสภูกทำนายยเข้าเป็นกลุ่มไม่เป็นมะเร็ง เพราะจำนวนนข้อมูลที่ใช้ในการสร้างโมเดลนั้นไม่สมดุล ดังนั้นในการทำนายกลุ่มจึงมีทิศทางถูกจำแนกไปยังกลุ่มที่มีข้อมูลมากกว่า

อย่างไรก็ตาม ความไม่สมดุลของข้อมูลในคลาส ไม่ได้หมายความถึงความแตกต่างของจำนวนข้อความ แต่รวมถึงขนาดของคลาส (Class Size) คลาสย่อย (Sub-Class) และคลาสที่มีการทับซ้อนของข้อมูล (Class Overlap) ซึ่งหมายถึงข้อมูลหนึ่งตัวสามารถปรากฏในหลายๆ คลาส เป็นต้น รายละเอียดแต่ละปัญหาสามารถอธิบายได้ดังนี้

(1) ปัญหาความไม่สมดุลอันเนื่องมาจากการกระจายข้อมูล (Data Distributed Imbalanced) [2] คือ จำนวนเอกสารข้อความในแต่ละกลุ่มมีจำนวนที่แตกต่างกันมาก กลายเป็นปัญหาของการจำแนกเอกสาร เพราะการกระจายตัวของเอกสารในแต่ละกลุ่มไม่เท่ากัน ดังนั้นในการจำแนกเอกสารเอกสารที่ถูกจำแนกจะมีโอกาสที่จะถูกทำนายไปอยู่ในกลุ่มที่มีเอกสารจำนวนมาก

(2) ปัญหาความไม่สมดุลอันเนื่องมาจากจำนวนเอกสารในแต่ละคลาสไม่เท่ากัน (Class Size Imbalanced) [2] นั่นคือ ขนาดของเอกสารในแต่ละกลุ่มไม่มีความสมดุลกัน

(3) ปัญหาการทับซ้อนของข้อมูล (Class Overlap) [2] คือ ปัญหาที่เกิดจากการที่เอกสารหนึ่งๆ มีโอกาสที่จะถูกจำแนกไปอยู่ได้ในหลายๆ กลุ่ม

(4) ปัญหากลุ่มย่อย (Sub-Class Problem) [2] คือ หลายๆ ปัญหาด้านการจำแนกเอกสาร อาจจะพบว่า ในกลุ่มๆ หนึ่งอาจจะมีหลายกลุ่มย่อย ซึ่งปัญหาดังกล่าวจะนำไปสู่ปัญหาความไม่สมดุลอันเนื่องมาจากจำนวนเอกสารในแต่ละคลาสไม่เท่ากันนั่นเอง

2.2 การจำแนกความรู้สึก (Sentiment Classification)

การจำแนกความรู้สึก [1] นั้น มีจุดประสงค์เพื่อวิเคราะห์เอกสารที่แสดงความรู้สึกออกเป็นความรู้สึกที่เป็นบวก (Positive) ความรู้สึกที่เป็นกลาง (Neutral) หรือความรู้สึกที่เป็นลบ (Negative) โดยทั่วไปเทคนิคที่ใช้ในการจำแนกความรู้สึกจะเป็นการผสมผสานระหว่างเทคนิคของการใช้การประมวลผลภาษาธรรมชาติ (Natural Language Processing : NLP) และเหมืองข้อความ (Text Mining)

ปัญหาอย่างหนึ่งที่พบในงานด้านการจำแนกเอกสารข้อความ รวมถึงการจำแนกความรู้สึกก็คือการที่จำนวนข้อมูลในแต่ละคลาสมีขนาดไม่เท่ากัน และเรียกปัญหานี้ที่พบในการจำแนกความรู้สึกว่า “การจำแนกความรู้สึกที่ไม่สมดุล (Imbalanced Sentiment Classification) [2]–[5]” โดยกลุ่มที่มีข้อมูลมากกว่าจะเรียกว่า “ข้อมูลกลุ่มหลัก (Majority Class)” และกลุ่มที่มีข้อมูลน้อยกว่าจะเรียกว่า “ข้อมูลกลุ่มรอง (Minority Class)” ในระหว่างการทำนายกลุ่ม ก็มักมีความเอนเอียงไปในทิศทางของข้อมูลกลุ่มหลัก เพราะมีข้อมูลที่มากกว่า

จากการศึกษาที่ผ่านมา พบว่ามีวิธีการในการจัดการกับปัญหาข้อมูลที่ไม่สมดุลหลายวิธี เช่น Re-Sampling [4], One-class Classification [10] และ Cost-Sensitive Learning [10] อย่างไรก็ตามวิธีการที่นำเสนอที่ผ่านมาก็ยังไม่สามารถจัดการปัญหาปัญหาข้อมูลที่ไม่สมดุลได้ทั้งหมด เพราะบริบทของข้อมูลในการศึกษามีความหลากหลาย วิธีการที่ใช้ได้ดีกับชุดข้อมูลหนึ่ง ก็ไม่ได้หมายความว่าจะใช้จัดการปัญหาข้อมูลที่ไม่สมดุลที่เกิดในข้อมูลชุดอื่นๆ ได้ดี

2.3 เทคนิคและอัลกอริทึมที่เกี่ยวข้อง

2.3.1 การจำแนกหมวดหมู่เอกสาร (Text Classification)

การจำแนกหมวดหมู่เอกสาร [10–12] เป็นการนำอัลกอริทึมการเรียนรู้ของเครื่องแบบมีผู้สอน (Supervised Machine Learning) มาประยุกต์รวมกับการประมวลผลภาษาธรรมชาติ เพื่อการจำแนกกลุ่มเอกสารแบบอัตโนมัติ โดยอาศัยการวิเคราะห์เนื้อหาของเอกสาร

โดยในการจำแนกเอกสารข้อความแบบอัตโนมัตินั้น จะใช้อัลกอริทึมการเรียนรู้ของเครื่องแบบมีผู้สอนในการสร้างตัวจำแนกเอกสาร (Text Classifier) จากเอกสารชุดสอน (Training Set) ที่เอกสารแต่ละฉบับต้องมีลาเบล (Label) ของคลาสกำกับ

กำหนดให้ D เป็นเซตของเอกสาร ขณะที่ C เป็นเป็นเซตของคลาสสที่เป็นไปได้ นั่นคือ $\{c_1,\ c_2, \ldots, c_{|c|}\}$ และกำหนดให้ T เป็นคู่ลำดับ $(d_j,\ c_i)$ ที่จะบ่งบอกว่าเอกสาร d_j อยู่ภายใต้กลุ่มหรือหมวดหมู่ c_i โดยให้ *F* เป็นฟังก์ชันที่กำหนดให้คู่ลำดับ $(d_j,\ c_i)$ เพื่อบอกว่าเอกสาร d_j ควรอยู่ภายใต้กลุ่มหรือหมวดหมู่ c_i หรือไม่ ดังนั้น การประมวลค่าของฟังก์ชันเป้าหมายสามารถแสดงได้คือ $F : D \times C \rightarrow \{T, F\}$ ซึ่งฟังก์ชันเป้าหมายจะแทนตัวจำแนกเอกสารนั่นเอง

2.3.2 ขั้นตอนการเตรียมเอกสาร (Document Pre-processing)

ในขั้นตอนนี้ จะเป็นการการเตรียมเอกสารหรือบทความให้อยู่ในรูปแบบที่พร้อมก่อนที่จะนำเข้าไปประมวลผลในขั้นตอนถัดไป ซึ่งจะมีขั้นตอนดังต่อไปนี้

2.3.2.1 การตัดคำ (Word Segmentation)

การตัดคำเป็นขั้นตอนแรกที่จะถูกดำเนินการในการประมวลผลภาษาธรรมติ ซึ่งเป็นการแบ่งข้อความ (String) ออกเป็นหน่วยย่อยที่มีความหมายทางภาษา โดยทั่วไปมักนิยมแบ่งข้อความออกมาเป็น “คำ (Word)” โดยในภาษาอังกฤษ การแบ่งข้อความออกเป็น “คำ” จะใช้ช่องว่าง (White Space) หรือเครื่องหมายวรรคตอน

2.3.2.2 การตัดคำหยุด (Stop-word Removal)

การตัดคำหยุด [15] คือ กระบวนการ การตัดคำหรือสัญลักษณ์ที่พบบ่อยในเอกสาร แต่คำเหล่านั้นไม่มีนัยสำคัญ ในที่นี้หมายถึงคำที่ใช้กันโดยทั่วไปไม่มีความสำคัญต่อเอกสารเมื่อตัดออกจากเอกสารแล้วไม่ทำให้ใจความสำคัญของเอกสารเปลี่ยนแปลง

ดังนั้น การตัดคำหยุด จึงมีความจำเป็นอย่างมากในการจัดกลุ่มเอกสารแบบอัตโนมัติ เพราะจะช่วยลดระยะเวลาในการประมวลผลได้เป็นอย่างมาก เนื่องจากระบบฯไม่ต้องนำคำเหล่านั้นไปประมวลผล เช่น คำว่า “is”, “the”, “are”, “and” แต่จะมีการนำคำที่มีผลต่องานออกจากพจนานุกรมคำหยุด เช่นคำว่า “not”, “very”, “much” เป็นต้น

ตารางที่ 2.1 แสดงการตัดคำหยุด

เอกสารที่ผ่านการทำ Lemmatization	เอกสารที่ผ่านการตัดคำหยุด
Black / space / is / great / song	Black / space / great / song
Possible / the / worst / music / of / the / year	Worst / music / year

2.3.2.3 การเปลี่ยนรูปแบบคำให้อยู่ในรูปแบบดั้งเดิม (Lemmatization)

การทำ Lemmatization คือ การเปลี่ยนคำให้มาอยู่ในรูปแบบดั้งเดิม (Lemma) เนื่องจากคำในภาษาอังกฤษนั้น มีการใช้งานคำที่มีความหมายเหมือนกันในลักษณะ [14] เช่น คำว่า “is”, “am”, “are”, “was” จะถูกเปลี่ยนเป็นคำว่า “be” หรือ “saw”, “seen” จะถูกเปลี่ยนเป็นคำว่า “see” ดังนั้นจึงจำเป็นที่ต้องมีการเปลี่ยนรูปแบบคำเหล่านั้นให้อยู่ในรูปแบบดั้งเดิม ในโครงงานนปริญญานิพนธ์นี้จะเลือกใช้วิธีการ Lemmatization Tagging โดยใช้ API จาก Stanford ซึ่งมีขั้นตอนการทำ Lemmatization ดังนี้

1. TokenizerAnnotator เป็นการตัดคำโดยใช้หลักการเดียวกับ *Penn Treebank* เช่น isn’t จะได้เป็น is, n’t ตัวอย่างเอกสาร

Yummy’s great song จะได้ song / Yummy / ‘s / great

2. ssplit เป็นการนำคำที่ผ่านกระบวนการตัดคำแล้ว มาเรียงลำดับตามประโยคเดิม

Yummy’s great song จะได้ Yummy / ‘s / great / song

3. POS (Part-Of-Speech Tagging) เป็นการติด tag ให้แต่ละคำเพื่อบอกว่าคำๆ อยู่ในบริบทใด เช่น bigger จะถูกกำหนด tag เป็น JJR (adj., comparative) เพื่อนำไปใช้ในการหาคำที่อยู่ในรูปแบบดั้งเดิม จากตัวอย่างเอกสารข้างต้นในขั้นตอนการติด tag จะได้

Yummy (NNP) | ‘s (POS) | great (NNP) | Song (NN)

4. Lemma จะเป็นการนำเอาคำที่ได้ภายหลังจากการติด tag มาทำ lemma โดยใช้ Wordnet ซึ่งจะมีการจัดกลุ่ม tag เป็น 5 กลุ่มคือ verbs (v), nouns (n), adjectives (a), satellite adjectives (s), adverbs (r)

Yummy	จะเป็นคำว่า	Yummy
‘s	จะเป็นคำว่า	is
great	จะเป็นคำว่า	great
song	จะเป็นคำว่า	song

2.3.3 การสร้างตัวแทนเอกสาร (Document Representation)

เนื่องจากคอมพิวเตอร์ไม่สามารถเรียนรู้ และจำแนกหมวดหมู่เอกสารที่เป็นภาษาธรรมชาติได้โดยตรง จึงจำเป็นต้องแปลเอกสารให้อยู่ในรูปแบบที่คอมพิวเตอร์สามารถใช้ในการเรียนรู้ได้ โดยขั้นตอนนี้เรียนกว่า การทำดัชนี (Indexing) [16] เพื่อสร้างตัวแทนเนื้อหาเอกสาร (Document Representation) สำหรับใช้ในกระบวนการเรียนรู้ วัตถุประสงค์ของการสร้างดัชนี คือ การคำนวณค่าที่จะนำมาใช้เป็นค่าคุณลักษณะของเอกสาร หรืออาจจะเรียกได้ว่าการหาน้ำหนัก (Term Weighting) การสร้างดัชนีโดยทั่วไปที่นิยมใช้กัน จะเริ่มจากการสร้างเวกเตอร์ตัวแทนเอกสารจากนั้นจะสร้างเมตริกซ์ของกลุ่มเอกสารขึ้นจากเวกเตอร์เอกสารทั้งหมดในกลุ่ม ซึ่งวิธีหาความถี่ของคำที่ปรากฏในเอกสารที่ผ่านการตัดคำมาเป็นค่าน้ำหนัก ถ้าคำใดผ่านการตัดคำมีปริมาณมาก ก็จะมีค่าความถี่มาก ซึ่งจะส่งผลให้ได้ค่าน้ำหนักที่มีค่าสูงมากตาม เมื่อถึงขั้นตอนนี้จะได้รูปแบบที่มีลักษณธของการแสดงความสัมพันธ์ระหว่างคำ (Words : *w*) และเอกสารทั้งหมด (Documents : *d*) ด้วย เวกเตอร์ 2 มิติ ซึ่งคำที่ได้นั้นต้องผ่านการทำดัชนีและการตัดคำหยุด (Stop-words) ออกไป และเอกสารทั้งหมดอยู่ในรูปแบบ Vector Space Model หากฟีเจอร์ (Feature) ที่ใช้เป็น “คำ” บางครั้งจึงเรียกรูปแบบนี้ว่า “ถุงคำ (Bag of Words: BOW)” [16] โดยสามารถแสดงได้ดังภาพประกอบที่ 2.1

	w_1	w_2	...	w_k	...	w_v
d_1	w_{11}	w_{12}	...	w_{1k}	...	w_{1v}
d_2	w_{21}	w_{22}	...	w_{2k}	...	w_{2v}
...	...	...	...	...	...	...
d_N	w_{N1}	w_{N2}	...	w_{Nk}	...	w_{Nv}

ภาพประกอบที่ 2.1 Bag of words

2.3.4 การเลือกคุณลักษณะ (Feature Selection)

ภายหลังจากการตัดคำ การตัดคำหยุด และการคัดเลือกคำด้วยพจนานุกรม คลังคำที่ได้จะถูกนำเข้าสู่ขั้นตอนของการคัดเลือกคุณลักษณะด้วย *Information Gain* สำหรับการเลือกคุณลักษณะจะเป็นวิธีเบื้องต้นในการลดขนาดเอกสาร [18, 19] เพราะการนำคำที่ไม่มีนัยสำคัญออกแล้วยังไม่เพียงพอ ซึ่งจำนวนคุณลักษณะมีผลต่อประสิทธิภาพของการจำแนกหมวดหมู่เอกสาร เนื่องจากอัลกอริทึมที่ใช้ในการเรียนรู้เพื่อสร้างตัวจำแนกหมวดหมู่ โดยทั่วไปไม่สามารถรองรับการทำงานกับจำนวนคุณลักษณะของเอกสารที่สูงมากได้ดี การลดขนาดของเอกสารจึงเป็นขั้นตอนหนึ่งที่จะต้อง

กระทำก่อน ในโครงงานปริญานิพนธ์นี้จะใช้ค่าเกนสารสนเทศ (IG: Information Gain) เป็นตัวลดคุณลักษณะของเอกสาร ซึ่งค่า *IG* จะคำนวณจากจำนวนบิตที่ได้รับสำหรับการทำนายกลุ่ม โดยการดูจากการมีอยู่หรือไม่มีอยู่ของคำในเอกสาร ให้ C_i , ..., C_K แทนเซตที่เป็นไปได้ของกลุ่ม ค่า *IG* ของคำ *w* นิยามโดย

$$IG(w) = -\sum P(c_j) \log P(c_j) + P(w) \sum P(c_j|w) \log P(c_j|w) + P(w) \sum P(c_j|w) \log P(c_j|w)$$

$P(C_j)$ คือความน่าจะเป็นของคลาสแต่ละคลาส

$P(w)$ คือความน่าจะเป็นของ “คำ” แต่ละคำที่พบ

$P(C_j|w)$ คือความน่าจะเป็นของ “คลาส” เพื่อพิจารณาจาก “คำ”

เมื่อทำการคำนวณค่า *IG* ของแต่ละคุณลักษณะที่ได้ จากนั้นทำการจัดเรียงคุณลักษณะที่มีค่า *IG* มากไปหาน้อยและทำการตัดคุณลักษณะที่มีค่าต่ำกว่าเกณฑ์ทิ้งไป ซึ่งจะช่วยลดระยะเวลาในการประมวลผล และยังคงความแม่นยำในการจัดกลุ่มเอกสาร

2.3.5 การให้น้ำหนักคำ (Term Weighting)

การให้น้ำหนักคำ [17] ถือว่าเป็นส่วนหนึ่งของการจัดการเอกสาร โดยรูปแบบการให้น้ำหนักสามารถแบ่งออกเป็นสองประเภทหลักตามการใช้งานข้อมูลชั้นเรียนในเอกสารการฝึกอบรม ดังนี้

1. การให้น้ำหนักคำแบบไม่มีผู้สอน (Unsupervised Term Weighting: UTW) [18] คือรูปแบบการให้น้ำหนักคำที่ซึ่งไม่ใช้ข้อมูลชั้นเรียนเพื่อสร้างน้ำหนัก รูปแบบที่ได้รับความนิยมมากที่สุด คือ Term Frequency - Inverse Document Frequency (TF-IDF) ซึ่งถูกใช้อย่างมีประสิทธิภาพในการศึกษาการดึงข้อมูล แต่อย่างไรก็ตามมันไม่เหมาะสำหรับงานการจัดหมวดหมู่ข้อความ เนื่องจากการให้น้ำหนักคำแบบ UTW เป็นการให้น้ำหนักคำกับเอกสารทั้งหมดโดยไม่แบ่งหมวดหมู่เอกสาร โดยหากใช้รูปแบบนี้จะทำให้ประสิทธิภาพในการจำแนกหมวดหมู่ข้อความลดลง

2. การให้น้ำหนักคำแบบมีผู้สอน (Supervised Term Weighting: STW) [11] ซึ่งได้รับการเสนอครั้งแรกโดย Debolc และ Sebastiani [11] การให้น้ำหนักคำแบบ STW จะใช้ชุดข้อมูลการฝึกอบรมของข้อมูลระดับชั้นเรียนเพื่อคำนวณน้ำหนักของคำศัพท์ โดยการให้น้ำหนักในแบบนี้จะใช้ประโยชน์จากข้อมูลระดับที่รู้จักในคลังข้อมูลการฝึกอบรม โดยจะทำให้การให้น้ำหนักมีประสิทธิภาพที่ดียิ่งขึ้น สำหรับการจำแนกหมวดหมู่ความรู้สึกของข้อความ การวิเคราะห์ความรู้สึก การจำแนกความไม่สมดุลของชุดเอกสาร และอื่นๆ โดยองค์ประกอบพื้นฐานของการกำหนดน้ำหนักมีดังตารางที่ 2.2

ตารางที่ 2.2 สัญลักษณ์สำหรับ Supervised Term Weighting (STW)

	c_k	$\bar{c_k}$
t_i	A	C
$\bar{t}_i$	B	D

โดยตัวแปรพื้นฐานมีดังต่อไปนี้

t_i คือ คำที่มีในเอกสาร

$\bar{t}_i$ คือ คำที่ไม่มีในเอกสาร

c_k คือ กลุ่มเอกสารกลุ่มหลัก

$\bar{c_k}$ คือ กลุ่มเอกสารกลุ่มรอง

A คือ จำนวนเอกสารใน c_k ที่คำว่า t_i เกิดขึ้นอย่างน้อยหนึ่งครั้ง

C คือ จำนวนเอกสารที่ไม่ได้เป็นของ c_k ที่คำว่า t_i เกิดขึ้นอย่างน้อยหนึ่งครั้ง

B คือ จำนวนเอกสารที่เป็นของ c_k โดยที่คำว่า t_i ไม่ได้เกิดขึ้น

D คือ จำนวนเอกสารที่ไม่ได้เป็นของ c_k โดยที่คำว่า t_i ไม่ได้เกิดขึ้น

N คือ จำนวนเอกสารทั้งหมดในคลังข้อมูล $N = A + B + C + D$

N_p คือ จำนวนเอกสารในชั้นบวก $N_p = A + B$

N_n คือ จำนวนเอกสารในชั้นเรียนที่เป็นลบ $N_n = C + D$

และตัวแปรพื้นฐานข้างต้นนำไปใช้ในอัลกอริทึมดังนี้

(1) Delta Term Frequency - Inverse Document Frequency (Delta TF-IDF)

Delta TF-IDF ถูกเสนอโดย Martineau และ Finin [19] มันคำนวณความแตกต่างของคะแนน TF-IDF ในคลาสที่เป็นบวกและลบเพื่อปรับปรุงความแม่นยำ ในฐานะที่เป็น STW จะพิจารณาการกระจายของคุณสมบัติระหว่างสองคลาสก่อนการจำแนกประเภทการรับรู้และการเพิ่มความสูงของผลคำที่แตกต่าง Delta TF-IDF ช่วยเพิ่มความสำคัญของคำที่กระจายอย่างไม่สม่ำเสมอระหว่างคลาสบวกและคลาสลบ โดยที่นี้ N_p และ N_n คือจำนวนของเอกสารในคลาสบวกและลบตามลำดับ ส่วน A และ C แสดงความถี่เอกสารของคำว่า t_i ในคลาสบวกและลบตามลำดับ ดัง (1)

$$w_{\&.TF.IDF}(t_i) = TF(t_i, d_j) \times log_2\left(\frac{N_p \times C + 1.5}{A \times N_n + 1.5}\right) \quad (1)$$

(2) Term Frequency - Inverse Document Frequency -Inverse Class Frequency (TF-IDF-ICF)

TF-IDF-ICF เป็นรูปแบบการควบคุมน้ำหนักตามแบบ TF-IDF แบบดั้งเดิม อย่างไรก็ตามมันเพิ่มปัจจัยความถี่ผกผันในคลาส (Inverse Class Frequency : ICF) [8] เพื่อให้ค่าน้ำหนักที่สูงขึ้นไปยังคำที่หายากที่เกิดขึ้นน้อยกว่าในเอกสาร (IDF) และ Class (ICF) และใน (2) M คือจำนวนคลาสในคอลเลกชันและ CF(t_i) สอดคล้องกับความถี่ของคลาสที่คำ t_i ปรากฏในคอลเลกชัน TF-IDF-ICF แสดงใน (2)

$$ICF(t_i) = (1 + log(\frac{M}{CF(t_i)})) \quad (2)$$

$$w_{TF.ICF}(t_i) = TF(t_i, d_j) \times IDF(t_i) \times ICF(t_i) \quad (3)$$

(3) Term Frequency - Relevance Frequency (TF-RF)

TF-RF [18] ได้รับการเสนอเช่นเดียวกับ Delta TF-IDF และ TF-RF คำนึงถึงการกระจายคำศัพท์ในชั้นเรียนทั้งบวกและลบ อย่างไรก็ตามมีการพิจารณาเฉพาะเอกสารที่มีคำดังกล่าวนั่นคือ ความเกี่ยวข้องของความถี่ (*RF)* ของข้อกำหนด TF-RF ถูกระบุใน (3) โดยที่ตัวหารน้อยที่สุดคือ *1* เพื่อหลีกเลี่ยงการหารด้วยศูนย์

$$w_{TF.RF}(t_i) = TF(t_i, d_j) \times log_2(2 + \frac{A}{max(1, C)}) \quad (4)$$

(4) Term Frequency - Inverse Gravity Moment (TF-IGM)

TF-IGM [20] ถูกนำเสนอให้วัดความไม่สม่ำเสมอหรือความเข้มข้นของการแจกแจงคำศัพท์ระหว่างคลาสซึ่งสะท้อนให้เห็นถึงอำนาจการจำแนกชั้นข้อตกลง

สมการ IGM มาตรฐานกำหนดอันดับ (r) ตามความเข้มข้นของการแจกแจงระหว่างคลาสของคำซึ่งคล้ายกับแนวคิดของ “แรงโน้มถ่วงโมเมนต์ (Gravity Moment: GM)” จากฟิสิกส์ *IGM* ถูกระบุใน (5) โดยที่ f_{ir} $(r = 1, 2, \ldots, M)$ ระบุจำนวนเอกสารที่มีคำว่า t_i ในคลาส $r - th$ ซึ่งส่วนโค้งเรียงตามลำดับจากมากไปน้อย ดังนั้น f_{il} จึงแสดงความถี่ของ t_i ในคลาสที่ปรากฏบ่อยที่สุด

$$IGM(t_i) = (\frac{f_{i1}}{\sum_{r=1}^{M} f_{ir} \times r}) \quad (5)$$

โดยน้ำหนักเทอม TF-IGM นั้นกำหนดตาม $IGM(t_i)$ ดังที่แสดงใน (6) ค่า λ คือ สัมประสิทธิ์แบบปรับได้ที่ใช้เพื่อรักษาสมดุลสัมพัทธ์ระหว่างปัจจัยทั่วโลก และท้องถิ่นในน้ำหนักของคำ สัมประสิทธิ์ λ มีค่าเริ่มต้นที่ 7.0 และสามารถตั้งเป็นค่าระหว่าง 5.0 ถึง 9.0 [20]

$$w_{TF.IGM}(t_i) = TF(t_i, d_j) \times (1 \times \lambda \times IGM(t_i)) \quad (6)$$

เพื่อแสดงให้เห็นถึงคุณสมบัติของการวัดน้ำหนักในระยะต่างๆ ได้ดีขึ้นให้พิจารณาองค์ประกอบพื้นฐานที่แสดงในตารางที่ 2.2 สมมติว่าชุดข้อมูลการฝึกอบรมมี 100 เอกสาร โดยพิจารณาการกระจายคำศัพท์ t_1 และ t_2 สำหรับสองคลาส c_p และ c_n ตามที่กำหนดไว้ใน

ตารางที่ 2.3 ตัวอย่างการแจกแจงเอกสารสองเทอม

	c_p	c_n			c_p	c_n
t_1	27	5		t_2	10	20
$\bar{t}_1$	3	65		$\bar{t}_2$	25	45

โดยคำนึงถึงการกระจาย t_1 ในตารางที่ 2.3 สามารถนำมาคำนวณการให้น้ำหนักได้ดังนี้

$$IDF(t_1) = log(100/(27+5)) = log(3.125) = 0.4949$$

$$IDF - ICF(t_1) = (1 + 0.4949) * (1 + log(2/2)) = 1.4949$$

$$Delta\ IDF(t_1, c_p) = log_2\left(\frac{30 * 5 + 0.5}{27 * 70 + 0.5}\right) = -3.6510$$

$$Delta\ IDF(t_1, c_n) = log_2\left(\frac{70 * 27 + 0.5}{5 * 30 + 0.5}\right) = 1.8445$$

$$RF(t_1, c_p) = log_2(2 + (27/5)) = 2.8875$$

$$RF(t_1, c_n) = log_2(2 + (3/65)) = 1.0329$$

$$IGM(t_1) = 27/((27 * 1) + (5 * 2)) = 0.7297$$

$$IGM.imp(t_1) = 27/((27 * 1) + (5 * 2) + 0.0458) = 0.7288$$

สามารถแสดงผลลัพธ์การคำนวณการกระจายน้ำหนักของ t_1 และ t_2 ได้ดังตารางที่ 2.4

ตารางที่ 2.4 ผลลัพธ์การคำนวณการกระจายน้ำหนัก

Weighting Scheme	$t_1\ c_p$	$t_1\ c_n$	$t_2 c_p$	$t_2 c_n$
IDF	0.4949	0.4949	0.5229	0.5229
$IDF - ICF$	2.9898	2.9898	3.0458	3.0458
$Delta\ IDF$	-3.6510	1.8445	-0.3782	-0.1069
RF	2.8875	1.0329	1.2630	1.3536
IGM	0.3333	0.3333	0.5000	0.5000

2.3.6 นาอีฟเบย์ (Naïve Bayes)

นาอีฟเบย์ (Naïve Bayes) เป็นการนำเอาหลักความน่าจะเป็นเข้ามาใช้ในการจำแนกข้อความ เนื่องจากนาอีฟเบย์นั้นเป็นอัลกอริทึมที่ง่ายไม่ซับซ้อน และมีความรวดเร็วในการใช้งาน ซึ่งในการคำนวณนาอีฟเบย์จะเริ่มคำนวณจากแต่ละตัวอย่าง จากตัวอย่างแรกไปยังตัวอย่างที่ *n* โดยค่าเป้าหมายที่ต้องการของแต่ละตัวอย่าง เป็นค่าใดๆ ภายในเซต *V* เมื่อ *V* มีสมาชิกเป็นค่าเป้าหมายที่ต้องการ ในที่นี้หมายถึงจำนวนกลุ่มของข้อมูล

นาอีฟเบย์เป็นการเรียนรู้อย่างง่าย เป็นวิธีการจำแนกประเภทของข้อมูลที่มีประสิทธิภาพวิธีหนึ่ง และเหมาะกับการนำมาใช้กับกรณีที่มีเซตตัวอย่างเป็นจำนวนมาก และแต่ละคุณสมบัติ (Attribute) ของตัวอย่างเป็นอิสระต่อกัน โดยนำการจำแนกประเภทนาอีฟเบย์มาประยุกต์ใช้ในการจำแนกประเภทของเอกสาร (Document Classification) พบว่ายังสามารถใช้งานได้ดีไม่ต่างจากการจำแนกวิธีการอื่นๆ และวิธีการไม่มีความซับซ้อน

การกำหนดความน่าจะเป็นของข้อมูลที่จะเป็นกลุ่ม V_j สำหรับข้อมูลที่มีคุณสมบัติ *n* ตัว $X = \{a_1,a_2,...a_n\}$ หรือใช้สัญลักษณ์ว่า $P(a_1,a_2,...a_n)$ คือ

$$P(v_j \mid a_1, a_2, \dots, a_n) = \prod_{i=1}^{n} P\,(a_i \mid v_j) \qquad (7)$$

โดยที่ $\prod$ หมายถึงผลคูณของค่า $P(a_i \mid v_j)$ เมื่อ *I* และ *j* มีค่าเท่ากับ *1, 2, 3, ..., n*

วิธีการเรียนรู้เบย์อย่างง่ายไปใช้มีวิธีดังต่อไปนี้คือ

(1) หาค่าความน่าจะเป็นของคำที่พบในแต่ละกลุ่มโดยนำค่า $P(a_1,a_2,...,a_n \mid v_j)$ จากสมการมาคูณกับค่าความน่าจะเป็นของกลุ่มนั้นๆ คือ $P(v_j)$ ได้เท่ากับ V_{NB}

(2) นำค่าที่ได้มาเปรียบเทียบกันกลุ่มที่มีความน่าจะเป็นสูงสุดคือกลุ่มที่ข้อมูลนั้นอยู่และจะถูกจัดเข้าไป เขียนเป็นสมการได้คือ

$$v_{NB} = argmax\ P(v_j) \times \prod_{i=1}^{n} P\ (a_i \mid v_j) \qquad : v_j\ \in V \tag{8}$$

2.3.7 วิธีการค้นหาเพื่อนบ้านใกล้ที่สุด (K-Nearest Neighbor: KNN)

วิธีการ KNN จะเป็นการจำแนกประเภทข้อมูลโดยขึ้นกับข้อมูลที่มีคุณสมบัติใกล้เคียงที่สุด *K* ตัวจากชุดข้อมูลตัวอย่าง แล้วเลือกคลาสที่สมาชิกส่วนใหญ่ที่อยู่ในกลุ่ม *K* ดังกล่าวสังกัดอยู่มากที่สุดให้กับ สมาชิกใหม่ การจำแนกประเภทข้อมูลโดยใช้ข้อมูลข้างเคียง *K* ตัวจะประกอบด้วยแอททริบิวต์หลายตัวแปร X_i ซึ่งจะนำมาใช้ในการแบงกลุม Y_i โดยระบุค่าตัวเลขจำนวนเต็มบวกให้กับ *K* ซึ่งค่านี้จะเป็นตัวบอกจำนวนของกรณี (Case) ที่จะต้องค้นหาในการทำนายกรณีใหม่ โดยในที่นี้จะกำหนด *1*-KNN หมายถึง อัลกอริทึมนี้จะค้นหา 1 กรณีที่มีลักษณะใกล้เคียงกับกรณีใหม่ (1 Nearest Cases) การนำระยะทางที่หาได้จากสมาชิกในข้อมูลตัวอย่างฝึกฝน มาเรียงลำดับจากน้อยไปหามากแล้วเลือกสมาชิกที่มีระยะทาง (Distance) ใกล้เคียงที่สุดออกมา *K* ตัว โดยใช้การวัดระยะทางแบบ Euclidean distance มีหลักการ คือ การวัดระยะทางระหว่างสองวัตถุ ถ้าวัตถุห่างกันมากแสดงว่าวัตถุนั้นมีความคล้ายคลึงกันน้อย ถ้ามีค่าน้อยก็แสดงว่ามีความคล้ายคลึงกันมาก โดยที่ คา p_i แทน คุณสมบัติจากฐานข้อมูล q_i แทนคุณสมบัติที่ผู้ใช้ระบุ

$$\sqrt{(\boldsymbol{p_1} - \boldsymbol{q_1})^2 + (\boldsymbol{p_2} - \boldsymbol{q_2})^2 + \cdots + (\boldsymbol{p_n} - \boldsymbol{q_n})^2} = \sqrt{\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} (\boldsymbol{p_i} - \boldsymbol{q_i})^2} \tag{9}$$

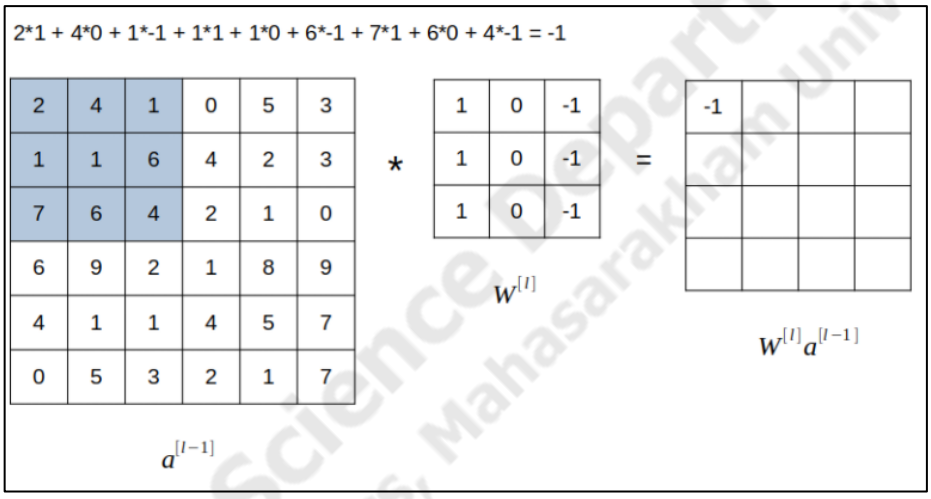
2.3.8 โครงข่ายประสาทแบบคอนโวลูชัน (Convolutional Neural Network: CNN)

CNN ได้รับการการนำเสนอ เพื่อให้ได้ผลลัพธ์ที่น่าประทับใจในภารกิจที่สำคัญในทางปฏิบัติของการจัดหมวดหมู่ประโยค ซึ่ง CNN สามารถใช้ประโยชน์จากการแทนคำแบบกระจายโดยการแปลงโทเค็น (Tokens) ที่ประกอบด้วยแต่ละประโยคเป็นเวกเตอร์ก่อนแล้วสร้างเมทริกซ์เพื่อใช้เป็นอินพุต

Convolutional Neural Network หรือ CNN ซึ่งเป็นโครงสร้าง Neural network แบบพิเศษ ที่มีความสามารถในการจำแนกข้อมูลได้ดีกว่า Neural network ทั่วไปมาก โดย CNN คือการที่

ใช้ Layer ชนิดพิเศษ ที่เรียกว่า Convolution layer ซึ่งทำหน้าที่สกัดเอาส่วนต่างๆ ของข้อมูลออกมา CNN จะใช้ Convolution layer มาประกอบกับ Layer ชนิดอื่น เช่น Pooling layer แล้วนำกลุ่ม Layer ดังกล่าวมาซ้อนต่อๆ กัน โดยอาจเปลี่ยน Hyperparameter บางอย่าง เช่นขนาดของ Filter layer (ซึ่งเป็นส่วนหนึ่งของ Convolution layer) และจำนวน Channel ของ layer วิธีการนำเอาส่วนต่างๆ มาประกอบกันนี้ เรียกว่าเป็นโครงสร้าง (Architecture) ของ CNN ซึ่งมีหลายแบบ เช่น LeNet, AlexNet, VGG, ResNet, Inception Network เป็นต้น ส่วนประกอบต่างๆ ของ CNN ซึ่งเป็นพื้นฐานที่เป็นส่วนสำคัญในการทำงานของ CNN ดังนี้

1) Convolution layer



ภาพประกอบที่ 2.2 ตัวอย่างการคำนวณ Convolution

จากภาพประกอบที่ 2.2 สมมุติเรามี Matrix ซ้ายมือ ขนาด 6x6 และมี Matrix ตรงกลาง ซึ่งเรียกว่า Filter หรือ Kernel ขนาด 3x3 เราจะนำเฉพาะ 3x3 ช่องแรกของ Matrix แรก มาคูณแบบ Element-wise กับ Filter matrix แล้วนำผลที่ได้แต่ละค่า (ซึ่งมีทั้งสิ้น 9 ค่า) มาบวกกัน แล้วนำไปสู่ในแถวแรกคอลัมน์แรกของ Matrix ที่สามซึ่งเป็นผลลัพธ์ โดยในภาพ ผลลัพธ์ที่ว่า เท่ากับ -1

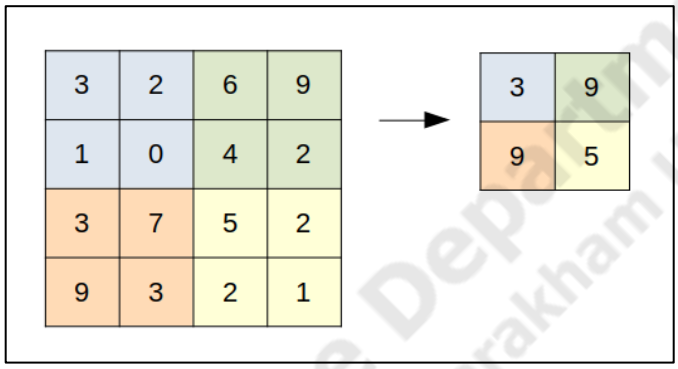
ถัดมา เราจะเลื่อนกรอบขนาด 3x3 ใน Matrix แรกไปทางขวา 1 ช่อง แล้วทำแบบเดิม ผลลัพธ์ที่ได้ นำไปใส่ในแถว 1 ช่อง 2 ของ Matrix ผลลัพธ์ ทำไปเรื่อยๆ จนสุดทาง แล้วเลื่อนกรอบ 3x3 ลงมาด้านล่าง 1 ช่อง (ชิดขอบด้านซ้ายมือ) แล้วทำแบบเดิม จนกระทั่งเติมค่าใน Matrix ผลลัพธ์จนเต็ม

กระบวนการนี้ เรียกว่า Convolution ซึ่งแสดงสัญลักษณ์ด้วย * ส่วน Neural network ที่มี Layer ที่ใช้กระบวนการ Convolution นี้อย่างน้อย 1 Layer เราก็เรียกว่า Convolutional neural network

2) Pooling layer

หลังจากที่ข้อมูลผ่าน Convolution layer แล้ว บ่อยครั้งที่จะถูกส่งเข้า Layer อีกแบบหนึ่งที่เรียกว่า Pooling layer

หน้าที่ของ Pooling layer คือการสกัดเอาส่วนที่สำคัญที่สุดของข้อมูล และเพิ่มประสิทธิภาพการประมวลผลให้รวดเร็วยิ่งขึ้น กลไกของ Pooling layer นั้นเรียบง่ายมาก คือการสกัดเอาเฉพาะค่าสูงสุดของ Grid เก็บไว้ใน Output เช่นจากภาพประกอบที่ 2.3 แสดง Pooling layer ขนาด 2x2 โดยมีค่า Stride s=2:

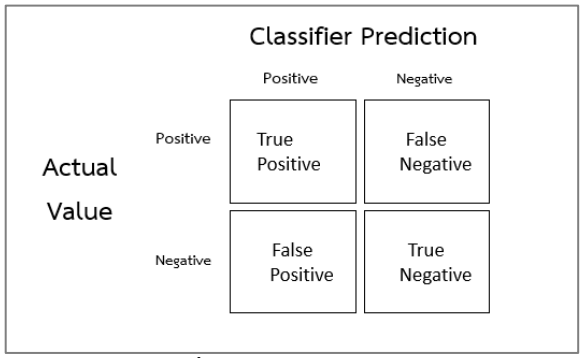


ภาพประกอบที่ 2.3 ตัวอย่างการทำ Pooling layer

Pooling layer ที่สกัดเอาเฉพาะค่าสูงสุดของ Grid เก็บไว้ เรียกว่า Max pooling ซึ่งเป็นรูปแบบที่ใช้บ่อยที่สุด นอกจากนั้นยังมี Average pooling ซึ่งหาค่าเฉลี่ยของ Grid เก็บไว้ แต่ใช้น้อยกว่า Max pooling มาก หลังจากที่ทำ Pooling layer เสร็จ ก็จะได้ feature map หรือ feature vector ที่จะนำไปทำเป็น model สำหรับทดสอบกับชุดข้อมูลอื่นๆ

2.3.9 การประเมิน (Evaluation)

ขั้นตอนการประเมินโมเดลเพื่อใช้ในการจัดการกลุ่มเอกสารก่อนนำไปใช้งานจริงที่โดยทั่วไปจะใช้เทคนิคมาตรฐาน [22] ที่เรียกว่า การวัดค่าความระลึก (Recall) การวัดค่าความแม่นยำ (Precision) และการวัดค่า F-measure



ภาพประกอบที่ 2.4 ตาราง Confusion Matrix

- True Positive (TP) คือ สิ่งที่โปรแกรมทำนายว่าจริง และคนบอกว่าจริง
- True Negative (TN) คือ สิ่งที่โปรแกรมทำนายว่าไม่จริง และคนบอกว่าไม่จริง
- False Positive (FP) คือ สิ่งที่โปรแกรมบอกว่าจริง แต่คนบอกว่าไม่จริง
- False Negative (FN) คือ สิ่งที่โปรแกรมบอกว่าไม่จริง แต่คนบอกว่าจริง

โดยจำนำค่าตาราง Confusion matrix มาใช้ในการคำนวณหสค่าความระลึก ค่าความแม่นยำ และค่า F-measure ได้ดังสมการต่อไปนี้

การวัดค่าความระลึก (Recall) [22] คือ เป็นอัตราส่วนของเอกสารที่จัดกลุ่มได้ จากเอกสารทั้งหมดที่มีอยู่ โดยจะนำค่าจากตาราง Confusion matrix มาใช้ในการคำนวณหาค่าความระลึกได้ดังนี้

$$Recall = \frac{tp}{tp + fn} \tag{10}$$

การวัดค่าความแม่นยำ (Precision) [22] คือ เป็นอัตาส่วนของเอกสารที่จัดกลุ่มได้และถูกต้อง ส่วนด้วยจำนวนเอกสารที่จัดกลุ่มได้

$$Precision = \frac{tp}{tp + fp} \tag{11}$$

การวัดค่า F-measure [22] เป็นการพิจารณาค่าความสัมพันธ์ระหว่างค่าความระลึกและค่าความแม่นยำ

$$F - measure = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{12}$$

โดยที่ค่า F จะมีค่าระหว่าง 0 ถึง 1 ซึ่งถ้าหากค่า F มีค่าเข้าใกล้ 1 มากเท่าไหร่ก็จำหมายถึงการจัดกลุ่มเอกสารมีประสิทธิภาพและมีความถูกต้องมากขึ้นเท่านั้น

2.4 งานวิจัยที่เกี่ยวข้อง (Related work)

ในการจำแนกความรู้สึกของเอกสารข้อความก็พบปัญหาของขจ้อมูลที่ไม่สมดุล ซึ่ง Li และคณะ [2] ได้ศึกษาเกี่ยวกับข้อมูลที่ไม่สมดุลหลายรูปแบบ เช่น จำนวนเอกสารที่ไม่สมดุล ขนาดของคลาสที่ไม่สมดุล รวมถึงความไม่สมดุลในคลาสย่อย จากการศึกษาที่ต่อเนื่องพบว่า ประเด็นที่หนึ่ง จำนวนเอกสารข้อความในสองคลาสจะเท่ากัน ความแตกต่างของจำนวนคำในเอกสารกลายเป็นปัจจัยสำคัญที่มีผลต่อความถูกต้องของการจำแนกเอกสาร ประเด็นที่สอง เพื่อปรับปรุงความถูกต้องของการจำแนกเอกสารด้วยการเพิ่มจำนวนของกลุ่มข้อมูบที่มีจำนวนน้อย และประเด็นที่สาม ในกรณีของข้อมูลที่ไม่สมดุล คำเดียวกันที่ปรากฏในสองคลาสมักจะเป็นสารสนเทศสำคัญของคลาส นั่นคือ คลาสทับซ้อนกันจะไม่ส่งผลกระทบต่อความถูกต้องของการจัดประเภท

Flavio Carvalho และ Gustavo Pai Guedes ได้นำเสนอการให้น้ำหนักคำแบบ Supervised Term Weighting ที่เหมาะสมต่อการจำแนกความรู้สึกที่ไม่สมดุล โดยได้นำเสนอการให้น้ำหนักคำที่ได้รับการควบคุมดูแลเจ็ดชุดและแผนการกำหนดน้ำหนัก ซึ่งวิธีนี้เป็นวิธีที่มีประสิทธิภาพมากกว่าการให้น้ำหนักคำในแบบ Unsupervised Term Weighting เนื่องจากการให้น้ำหนักคำในรูปแบบนี้เป็นใช้ประโยชน์จากข้อมูลที่อยู่ในคลังข้อมูลการฝึกอบรม

ในปี ค.ศ. 2011 Shoushan Li และคณะได้ทำงานวิจัย Imbalance Sentiment Classification [23] เพราะเล็งเห็นปัญหาในการจำแนกความรู้สึกที่ไม่สมดุลของข้อมูล เนื่องจากวิธีก่อนหน้านี้มีปัญหาในการทำงานค่อนข้างมาก จึงได้นำเสนอ วิธีการจำแนกความรู้สึกที่ไม่สมดุล โดยเสนอโครงร่างการจัดกลุ่มแบบ under-sampling ด้วยการแบ่งเป็นกลุ่มเพื่อเอาชนะปัญหาการกระจายระดับความไม่สมดุลในการจำแนกความรู้สึกไม่สมดุล ภายใต้กรอบงานนี้ กลุ่มตัวอย่างในกลุ่มเสียงส่วนใหญ่จะถูกจัดกลุ่มเป็นกลุ่มแรก จากนั้นเลือกกลุ่มตัวอย่างจำนวนที่เหมาะสมจากแต่ละกลุ่มจากตัวอย่างการฝึกอบรมของข้อมูลส่วนใหญ่

ในงานวิจัยของ Ah-Pine และ Pavel Soriano Morales [6] ศึกษาแก้ปัญหาความไม่สมดุลของข้อมูลในการวิเคราะห์ความรู้สึก (Sentiment Classification) ที่ใช้ข้อมูลจาก twitter ที่พบว่าการกระจายกลุ่มของข้อมูลมีความเอนเอียงไปกลุ่มใดกลุ่มหนึ่ง นั่นคือจำนวนข้อมูลในแต่ละกลุ่มขาดความสมดุล ดังนั้นนักวิจัยจึงนำเสนอการทำเทคนิคการสุ่มตัวอย่างแบบสังเคราะห์ (Synthetic Oversampling Techniques) สำหรับการจำแนกกลุ่มข้อความ Twitter

อย่างไรก็ตาม งานวิจัยส่วนใหญ่ที่ใช้ในการแก้ปัญหาข้อมูลไม่สมดุลในการจำแนกเอกสารมันทำผ่านการคัดเลือกคุณลักษณะที่เหมาะสม (Feature Selection) เช่น งาน Zheng และคณะ นำเสนอการศึกษาเรื่องการคัดเลือกเอกสารที่เหมาะสม เพื่อเพิ่มประสิทธิภาพในการจำแนกเอกสารข้อความที่มีประสิทธิภาพ โดยทั่วไป information gain (IG), chi-square (CHI), correlation coefficient (CC) และ odds ratios (OR) ล้วนเป็นเทคนิคในการคัดเลือกคุณลักษณะที่มีประสิทธิภาพ CC และ OR เป็นตัวชี้วัดด้านเดียว (one-sided metrics) ในขณะที่ IG และ CHI เป็นแบบสองด้าน (two-sided metrics) การเลือกคุณสมบัติโดยใช้การวัดด้านเดียวเลือกคุณบักษณะที่บ่งบอกถึงการเป็นสมาชิก (membership) มากที่สุดเท่านั้นในขณะที่การเลือกคุณลักษณะโดยใช้การวัดสองด้านโดยนัยรวมคุณลักษณะที่บ่งบอกถึงการเป็นสมาชิกมากที่สุด (เช่น คุณสมบัติเชิงบวก) ด้วยการไม่สนใจร่องรอยหรือเครื่องหมายของคุณลักษณะ

ซึ่งในการศึกษาที่ผ่านมาจะไม่ให้ความสำคัญกับคุณลักษณะเชิงลบ (negative features) ที่ค่อนข้างมีความสำคัญ ในขณะที่ต่อมา พบว่าการผสมผสานคุณสมบัติทั้งเชิงบวกและเชิงลบจะสามารถเพิ่มประสิทธิภาพในการจำแนกเอกสาร โดยเฉพาะอย่างยิ่งกับข้อมูลที่ไม่สมดุล ในงานวิจัยนี้ นักวิจัยได้ศึกษาเกี่ยวกับกระบวนการในการคัดเลือกเอกสารที่มีการควบคุมคุณสมบัติทั้งเชิงบวกและเชิงลบอย่างเหมาะสม ขณะที่มีการใช้ multinomial naïve Bayes และ regularized logistic regression ในการ

สร้างตัวจำแนกเอกสาร ผลลัพธ์ที่ได้จากการทดสอบแสดงให้เห็นกระบวนการคัดเลือกคุณลักษณะในการรวมคุณสมบัติบวกและลบในการแก้ปัญหาข้อมูลที่ไม่สมดุลได้ให้ประสิทธิภาพที่ดี