KKU SCIENCE JOURNAL
Journal Home Page : https://ph01.tci-thaijo.org/index.php/KKUSciJ
Published by the Faculty of Science, Khon Kaen University, Thailand

การพัฒนาพจนานุกรมคำไม่สุภาพภาษาไทยและขั้นตอนวิธีการตรวจจับคำไม่สุภาพภาษาไทยด้วยรายการผกผันพหุแบบ

Development of a Thai Impolite Word Dictionary and an Algorithm for Detecting Thai Impolite Words using Multiple Patterns Inverted Lists

ขณิดา จรุงจิตต์[1] ณัฐวุฒิ แก้วศิริ[1*] และ เชาวลิต ขันคำ[2]

Kanida Charungchit[1], Nattawut Kaewsiri[1*] and Chouvalit Khancome[2]

[1]สาขาวิชาเทคโนโลยีสารสนเทศ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏราชนครินทร์ จังหวัดฉะเชิงเทรา 24000

[2]ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยรามคำแหง กรุงเทพมหานคร 10240

[1]Information Technology, Faculty of Science and Technology, Rajabhat Rajanagarindra University, Chachoengsao, 24000, Thailand

[2]Computer Science Department, Faculty of Science, Ramkhamhaeng University, Bangkok, 10240, Thailand

บทคัดย่อ

บทความวิจัยนี้นำเสนอแนวคิดใหม่สำหรับตรวจจับคำไม่สุภาพภาษาไทยด้วยการออกแบบโครงสร้างพจนานุกรมจัดเก็บคำไม่สุภาพแบบใหม่ โดยใช้ตำแหน่งอักขระแทนการจัดเก็บคำในพจนานุกรมแบบเดิม จำแนกส่วนประกอบของแต่ละคำออกเป็นรายการผกผัน จัดเก็บในพจนานุกรมด้วยตาราง โดยอาศัยหลักการแฮชเพื่อการเข้าถึงที่รวดเร็ว จากนั้นพัฒนาขั้นตอนวิธีสำหรับตรวจจับคำไม่สุภาพได้แบบทันทีทันใดในระหว่างที่อักขระถูกป้อนเข้ามาในระบบ โดยไม่จำเป็นต้องป้อนคำจนสำเร็จเช่นขั้นตอนวิธีเดิมที่เคยมีมา พจนานุกรมใหม่ถูกสร้างด้วยค่าความซับซ้อนด้านเวลา O(W) และค่าความซับซ้อนด้านเนื้อที่จัดเก็บ O(|Σ|+|W|) โดยที่ W คือความยาวของจำนวนคำรวมกันทั้งหมดที่บรรจุในพจนานุกรม ในขณะที่ความซับซ้อนด้านเวลาการตรวจจับคือ O(n) เมื่อ n คือความยาวของข้อความที่ป้อนเข้าสู่ระบบ ผลการทดลองด้วยการพัฒนาโปรแกรมตรวจจับคำไม่สุภาพโดยอาศัยพจนานุกรมแบบใหม่ มีความรวดเร็วในการตรวจจับมากกว่าการตรวจจับโดยอาศัยพจนานุกรมแบบเดิมอย่างมีนัยสำคัญ ขั้นตอนวิธีใหม่สามารถตรวจจับคำไม่สุภาพที่มีในพจนานุกรมได้ค่าความถูกต้องร้อยเปอร์เซ็นต์ โดยไม่มีค่าความคลาดเคลื่อน

*Corresponding Author, E-mail: Nattawut.kae@csit.rru.ac.th

Received: date: 6 March 2024 | Revised date: 17 March 2024 | Accepted date: 18 March 2024

ABSTRACT

This research article presents a new concept for detecting impolite Thai words by designing a new structure for a dictionary that stores impolite words. It uses character positions instead of storing words in the traditional dictionary format. The components of each word are classified into an inversion list and stored in the dictionary using a table based on hashing principles for rapid access. The article then develops an algorithm for immediate detection of impolite words as characters are input into the system, eliminating the need for complete entry as required by previous algorithms. The new dictionary is created with a time complexity of O(W) and a space complexity of O(|Σ|+|W|), where W is the total length of all words contained in the dictionary. Meanwhile, the time complexity for detection is O(n), where n is the length of the text input into the system. Experimental results with the developed program for detecting impolite words, relying on the new dictionary model, show significantly faster detection compared to the traditional dictionary-based approach. The new algorithm can detect impolite words in the dictionary with 100% accuracy without any errors.

คำสำคัญ: คำไม่สุภาพ ขั้นตอนวิธี การตรวจจับคำหยาบ รายการผกผัน โครงสร้างข้อมูล
Keywords: Impolite Words, Algorithm, Rude Words Detection, Inverted List, Data Structure

บทนำ

การตรวจจับคำที่ไม่สุภาพ (Impolite word detection) หรือ การตรวจจับคำหยาบ (Rude word detection) คือ กระบวนการประมวลผลภาษาธรรมชาติ (Natural language processing) เพื่อตรวจจับคำไม่สุภาพหรือภาษาที่ไม่เหมาะสมที่ปรากฏในข้อความหรือสื่อต่างๆ หลักการนี้ถูกพัฒนาขึ้นเพื่อตรวจจับป้องกันการใช้ภาษาที่ไม่เหมาะสม หรือทำให้รู้สึกไม่พึงประสงค์ในสังคมออนไลน์หรือในแอปพลิเคชันที่ใช้ในการสื่อสารในปัจจุบัน เช่น Google Safe Browsing Facebook's AI และ Twitter's AI เป็นต้น ตามแนวคิดดังกล่าว มีงานวิจัยที่น่าสนใจในต่างประเทศ แสดงไว้ในบทความเป็นจำนวนมาก (Chakraborty *et al*., 2023; Chakravarthi *et al*., 2023; Chen and Feng, 2013; Li, 2020; Lusiana *et al*., 2018; Magami and Digiampietri, 2020; Modha *et al*., 2020; Moin *et al*., 2021; Novitasari *et al*., 2018; Priya *et al*., 2023; Tuarob *et al*., 2023) โดยแต่ละกลไกที่สำคัญและเทคนิคในการตรวจจับ พบว่าจะอาศัยทั้งหลักการทางสถิติ หรือพิจารณาความหมาย หรือไม่ก็อาศัยพจนานุกรมเพื่ออ้างอิงคำไม่สุภาพดังกล่าว

สำหรับในประเทศไทยมีการวิจัยทางด้านนี้ที่น่าสนใจ คือ สร้างโมเดลตรวจจับคำหยาบภาษาไทยบนสื่อออนไลน์ (ณัฐาศิริและสมชาย, 2560) และวิจัยเกี่ยวกับการหลีกเลี่ยงการใช้คำไม่สุภาพของผู้ใช้อินเทอร์เน็ตบนกระดานสนทนา (วิภาพรรณ, 2549) เป็นต้น ในการดำเนินการด้วยกลไกการตรวจจับต่างๆ เหล่านั้น พบว่ากลไกที่สำคัญคือการใช้ฐานพจนานุกรม (Dictionary-based algorithm) สำหรับตรวจจับ โดยกลไกนี้จะใช้พจนานุกรมของคำศัพท์หรือวลีที่เป็นที่รู้จักว่าเป็นคำไม่สุภาพหรือคำหยาบ การรวบรวมคำศัพท์ จัดเก็บไว้ในพจนานุกรม (ณัฐาศิริและสมชาย, 2560; Khancome and Boonjing, 2009) จากนั้นพัฒนาขั้นตอนวิธีตรวจสอบข้อความว่ามีคำศัพท์หรือวลีเหล่านี้ปรากฏอยู่ในพจนานุกรมหรือไม่ หากพบก็จะถือว่าข้อความนั้นไม่เหมาะสม หลักการตรวจจับคำไม่สุภาพแบบฐานพจนานุกรมมีความแม่นยำสูง

งานวิจัยนี้ศึกษาจาก (ณัฐาศิริและสมชาย, 2560; วิภาพรรณ, 2549) ร่วมกับการศึกษาหลักการออกแบบและการพิจารณาคำไม่สุภาพจากเถกิง (2558) นันทวัฒน์และสุวัฒนา (2558) นภัทรและปนันดา (2562) ยุวดี (2549) โดยเฉพาะอย่างยิ่งในงานวิจัยวุฒิชัย (2549) ที่มีประสิทธิภาพได้สร้างพจนานุกรมด้วยการตัดคำ และจำเป็นต้องเก็บพจนานุกรมดังกล่าวแบบเต็ม และต้องพิมพ์หรือป้อนข้อความทั้งหมดก่อนการตรวจจับ ขณะที่ณัฐาศิริและสมชาย (2560) แสดงโมเดลที่ได้จากการสอน (Train) ด้วยข้อมูลด้วยพจนานุกรมเช่นกัน การป้อนหรือการตรวจจับจำเป็นต้องป้อนข้อมูลครบก่อนการตรวจจับ ทั้งสองขั้นตอนวิธีจำเป็นต้องมีข้อมูลทดสอบด้วยการป้อนให้ครบทั้งหมด เนื่องจากพจนานุกรมดังกล่าวที่ใช้เป็นแบบเดิมที่จัดเก็บเป็นคำสำหรับตรวจสอบและเรียนรู้ คณะผู้วิจัยจึงให้ความสนใจและสร้างงานวิจัยในฐานพจนานุกรมโดยมุ่งเน้นขจัดข้อจำกัดของพจนานุกรมแบบเดิม ที่การตรวจจับจำเป็นต้องป้อนคำศัพท์หรือข้อความไม่สุภาพทั้งหมดก่อนจึงเข้าไปตรวจจับได้ ซึ่งส่งผลให้ก่อนการตรวจจับจะทำให้มีคำไม่สุภาพถูกนำเข้าสู่ระบบได้เรียบร้อยแล้วจึงตรวจพบในภายหลัง

ดังนั้น งานวิจัยนี้จึงต้องการสร้างพจนานุกรมแบบใหม่ สามารถตรวจจับและทำนายข้อความได้ทันทีทันใดตั้งแต่ตอนป้อนทีละอักขระ สามารถทำนายแนวโน้มของคำขณะป้อนเข้าสู่ระบบได้แบบทันทีทันใด คณะผู้วิจัยได้ออกแบบแนวคิดใหม่สำหรับตรวจจับคำไม่สุภาพ ด้วยการพัฒนาโครงสร้างข้อมูลพจนานุกรมจัดเก็บคำไม่สุภาพแบบใหม่ สร้างการจัดเก็บข้อมูลในรูปแบบอักขระ ใช้ตำแหน่งอักขระแทนการจัดเก็บคำในพจนานุกรมแบบเดิม จำแนกรายละเอียดของแต่ละคำออกเป็นรายการผกผัน (Inverted lists) จัดเก็บในพจนานุกรมเฉพาะที่ใช้หลักการแฮช (Hashing) สำหรับการเข้าถึงที่รวดเร็ว จากนั้นพัฒนาขั้นตอนวิธีสำหรับตรวจจับคำดังกล่าว ซึ่งสามารถตรวจจับคำไม่สุภาพได้แบบทันทีทันใดที่อักขระถูกป้อนเข้ามาในระบบโดยไม่จำเป็นต้องให้เกิดการป้อนจนสำเร็จเสียก่อนจึงจะตรวจพบเช่นขั้นตอนวิธีเดิมที่เคยมีมา

แนวคิดใหม่สร้างพจนานุกรมด้วยค่าความซับซ้อนด้านเวลา $O(|W|)$ และค่าความซับซ้อนด้านเนื้อที่จัดเก็บ $O(|\Sigma|+|W|)$ โดยที่ W คือจำนวนคำที่บรรจุในพจนานุกรม ในขณะที่ความซับซ้อนด้านเวลาการค้นหาคือ $O(n)$ เมื่อ n คือความยาวของข้อความที่ป้อนเข้าสู่ระบบ ผลการทดลองพัฒนาโปรแกรมเพื่อเปรียบเทียบการตรวจนับกับเทคนิคการตรวจนับโดยอาศัยพจนานุกรมแบบเดิม พบว่าพจนานุกรมแบบใหม่สามารถตรวจนับได้รวดเร็วกว่ามากกว่าอย่างมีนัยสำคัญที่การนำเข้าข้อมูลไม่จำเป็นต้องอาศัยการตัดคำก่อนการค้นหาเช่นขั้นตอนวิธีเดิม รวมถึงขั้นตอนวิธีใหม่มีความถูกต้อง (Accuracy) สูง และไม่มีความคลาดเคลื่อนอีกด้วย นอกจากนั้นพจนานุกรมใหม่นี้ยังเอื้อประโยชน์ต่อความเร็วในการตรวจจับและในอนาคตสามารถเพิ่มคำใหม่ๆ เข้าไปในพจนานุกรมคำไม่สุภาพ ซึ่งแนวคิดใหม่นี้สามารถนำไปใช้แบบพลวัตได้อีกด้วย รวมถึงสามารถขยายความสามารถในอนาคตให้พิจารณาได้ถึงบริบทของคำหยาบตามรูปแบบของบริบทได้ด้วย อีกทั้งยังได้พัฒนาขั้นตอนวิธีการตรวจจับคำไม่สุภาพใหม่ที่สัมพันธ์และเชื่อมโยงในมิติรูปภาพหรือภาษาถิ่นในอนาคตได้อีกด้วย

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ทฤษฎีที่นำมาใช้งาน ได้แก่ รายการผกผันแบบพหุแบบ ตารางการแฮช และงานวิจัยที่เกี่ยวข้องที่ได้ศึกษาเพื่อนำมาพัฒนาออกแบบงานวิจัยนี้

รายการผกผันพหุแบบ

แนวคิดของการสร้างรายการผกผัน (Khuncome and Boonjing, 2009) เกิดจากการแทนเอกสาร D ด้วยเซตอักขระแบบ W โดยที่ $W = \{w_1, w_2, ..., w_r\}$ เมื่อ w_i คือ สายอักขระจาก $c_1, c_2, ..., c_m$ ที่อยู่ภายใต้ Σ ซึ่ง Σ คือ เซตของอักขระ (Character set) ที่ปรากฏใน W

แสดงตัวอย่างของ เซต W = {aab,aabc,aade}

w_1 = aab, w_2 = aabc, w_3 = aade

เมื่อพิจารณา w_1 = aab ซึ่งจะได้ c_1 คือ a, c_2 คือ a, c_3 คือ b เป็นต้น

จากแนวคิดที่จะแทนรายละเอียดของเอกสาร D ด้วยเซตอักขระแบบ W นำมาสร้างดัชนีของรายการผกผันของแต่ละอักขระ ในรูปแบบของ อักขระ : <ตำแหน่งที่ปรากฏในแต่ละอักขระแบบ : สถานะของการเป็นตัวสุดท้ายหรือไม่ : {หมายเลขของอักขระแบบที่มีอักขระดังกล่าวปรากฏ}>

ดังนั้นสามารถแสดงดัชนีของรายการผกผันของ เซต W = {aab,aabc,aade} ดังนี้

a: <1:0:{1,2,3}>, <2:0:{1,2,3}>

b: <3:1:{1}>, <3:0:{2}>

c: <4:1:{2}>

d: <3:0:{3}>

e: <4:1:{3}>

เมื่อแทนค่าในรูปแบบดังกล่าวแล้วจึงนำเอาดัชนีรายการผกผันมาสร้างเป็นตารางดังนี้

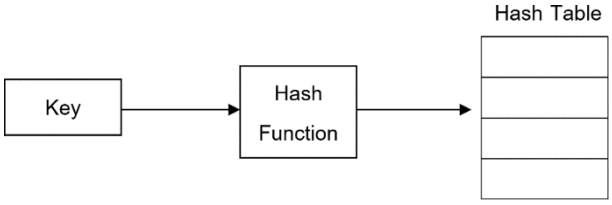
ตารางที่ 1 แสดงรายการผกผันแบบพหุ W = {aab,aabc,aade}

อักขระ	รายการผกผันก่อนการแทรก
A	<1:0:{1,2,3}><2:0:{1,2,3}>
B	<3:1:{1}><3:0:{2}>
C	<4:1:{2}>
D	<3:0:{3}>
e	<4:1:{3}>

จากนั้นนำเอารายการผกผันดังกล่าว ไปสร้างและนำเสนอในรูปแบบของตารางการแฮชเพื่อใช้เป็นโครงสร้างข้อมูลสำหรับการค้นหาต่อไป

ตารางการแฮช

ตารางการแฮช (Hashing table) (Dinh, 2020; Loudon, 2020, Wikipedia, 2020) คือ โครงสร้างข้อมูลที่ออกแบบเพื่อการจัดเก็บในรูปแบบตาราง มีกลไกการทำงาน 3 ส่วน คือ คีย์ (Key) แฮชฟังก์ชัน (Hash Function) และส่วนของตารางการเก็บข้อมูล สนับสนุนค้นหาข้อมูลในโครงสร้างด้วยความซับซ้อนต่ำสุดคือ O(1) โดยปกติจะสร้างตารางไว้เก็บข้อมูล (Bucket of Data) และเข้าถึงข้อมูลโดยใช้การคำนวณเพื่อหาตำแหน่งในการเก็บข้อมูลด้วยฟังก์ชันการแฮช รูปที่ 1 แสดงแนวคิดดังกล่าว



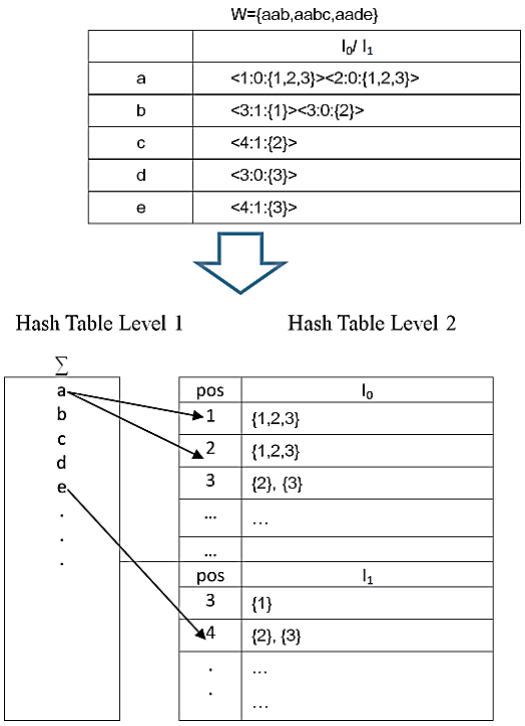
รูปที่ 1 แนวคิดการสร้างตารางการแฮช

ตารางแฮชพื้นฐานดังในรูปที่ 1 อาจมีปัญหาเรื่องของการชนกันของค่าคีย์ (Key collision) และเนื้อที่จัดเก็บที่ไม่ยืดหยุ่นและทำให้การเข้าถึงอาจต้องใช้ความซับซ้อนมากกว่า O(1) แต่ในปัจจุบันมีการพัฒนาเป็นตารางการแฮชสมบูรณ์แบบ (Perfect hashing) ซึ่งใช้เนื้อที่จัดเก็บข้อมูลเพียง O(n) และเวลาเข้าถึงเป็น O(1) เท่านั้น กลไกนี้จะพัฒนาตารางการแฮชเป็นสองระดับ และเข้าถึงโดยอาศัยคีย์เข้าถึง 2 คีย์ คือ

1) คีย์สากล U สำหรับการเข้าถึงข้อมูลในตารางระดับแรกด้วยค่าฟังก์ชันแฮช f(U)

2) คีย์ k สำหรับเข้าถึงระดับที่สองโดยฟังก์ชัน f(k) และข้อมูลที่เข้าถึงในระดับที่สอง คือ รายการข้อมูลที่เกี่ยวข้องกับคีย์ที่เกี่ยวข้องของ k ที่สัมพันธ์มาจากค่าของข้อมูลในตารางการแฮชระดับแรก

งานวิจัยนี้นำแนวคิดของตารางการแฮชสมบูรณ์แบบ (Dinh, 2020; Loudon, 2020, Wikipedia, 2020) มาประยุกต์ใช้สำหรับสร้างตารางรายการผกผันแบบพหุ โดยกำหนดคีย์สากล U คือ Σ เข้าถึงข้อมูลตารางระดับแรก คือ ตัวอักขระแบบทั้งหมด f(Σ) และหลังจากนั้นนำค่าตำแหน่งที่ปรากฏในแต่ละอักขระแบบที่ได้จาก f(Σ) เป็นค่าคีย์ k และ เข้าถึงรายการผกผันที่สร้างจัดเก็บไว้ในตารางแฮชระดับที่สองด้วย f(k) เพื่อเข้าถึงรายการผกผันในตารางการแฮชระดับที่ 2 และนำมาวิเคราะห์หาการตรวจจับที่ถูกต้อง แสดงแนวคิดการสร้างตารางการแฮชจากรายการผกผันในตารางที่ 1 ดังรูปที่ 2



รูปที่ 2 แนวคิดการสร้างตารางการแฮชสมบูรณ์แบบ

นิยามและการแสดงตัวอย่างต่อไปนี้นำแนวคิดจาก (Khancome and Boonjing, 2009) มาพัฒนาเพื่อสร้างพจนานุกรมคำไม่สุภาพภาษาไทย ดังนี้

นิยามที่ 1 กำหนดให้ W คือ ชุดของคำไม่สุภาพที่ประกอบด้วย $w_1, w_2, w_3, ..., w_n$ เขียนแทนด้วยเซตคำไม่สุภาพ

$$W = \{w_1, w_2, w_3, ..., w_r\}$$

ตัวอย่างที่ 1 แสดงตัวอย่างของ $W = <w_1, w_2, w_3, ..., w_r>$ เมื่อ r = 5 จากนิยามที่ 1 แสดงชุดของคำดังนี้

W = <ตอแหล, เฮงซวย, ไอ้ระยำ, ไอ้เบื๊อก, ไอ้หน้าโง่>

โดย w_1 คือ คำว่า “ตอแหล” และเรียงลำดับไปจนถึง w_5 คือคำว่า “ไอ้หน้าโง่” เป็นต้น

นิยามที่ 2 กำหนดให้แต่ละคำไม่สุภาพ w_i มีความยาว m_j เมื่อ $1 \le j \le r$ อักขระ ดังนั้น ความยาวรวมของอักขระที่บรรจุ อยู่ใน W คือ $m_1 + m_2 + ... + m_r$ เขียนแทน |W|

ตัวอย่างที่ 2 แสดงตัวอย่างของ W = <ตอแหล, เฮงซวย, ไอ้ระยำ, ไอ้เบื๊อก, ไอ้หน้าโง่> จะมีความยาว

|W| = 5 + 6 + 7 + 9 + 10 = 37 เป็นต้น

นิยามที่ 3 กำหนดให้ Σ คือ อักขระทั้งหมดที่ถูกบรรจุใน W เขียนแทนจำนวนอักขระทั้งหมดด้วย $|\Sigma|$

ตัวอย่างที่ 3 แสดงตัวอย่าง W = <ตอแหล, เฮงซวย, ไอ้ระยำ, ไอ้เบื๊อก, ไอ้หน้าโง่> จากนิยาม Σ คือ ต, อ, แ, ..., โ, ง่ เป็นต้น ซึ่งหากพิจารณาภาษาไทยตามนิยาม 3 คือ อักขระภาษาไทย สระ และวรรณยุกต์รวมกัน

นิยามที่ 4 กำหนด $\mathbf{c}$ คือ อักขระเดี่ยวที่ปรากฏใน wj ใดๆ โดยที่ $\mathbf{c} \in \Sigma$ สามารถเขียนดัชนีของ $\mathbf{c}$ ที่ปรากฏด้วย $\mathbf{c}$: <pos:w_j:terminated> โดยที่ pos คือ ตำแหน่งของอักขระที่พบในคำที่ w_j และ terminated คือ สถานะที่ระบุว่าเป็นอักขระสุดท้ายของคำไม่สุภาพ w_j หรือไม่ โดยกำหนดให้ terminated = 1 ถ้าเป็นอักขระสุดท้ายของคำ หรือ terminated = 0 หากไม่ใช่อักขระสุดท้ายของคำ

ตัวอย่างที่ 4 จากคำไม่สุภาพ W = <ตอแหล, เฮงซวย, ไอ้ระยำ, ไอ้เบื๊อก, ไอ้หน้าโง่> แสดงตัวอย่างดัชนีอักขระได้ดังตารางที่ 2

ตารางที่ 2 ตัวอย่างดัชนีอักขระของคำไม่สุภาพ

$\mathbf{c}$	<pos:w_j:terminated>	$\mathbf{c}$	<pos:w_j:terminated>
ต	<1:1:0>	ร	<4:3:0>
อ	<2:1:0> , <2:3:0>, <2:4:0>, <8:4:0>, <2:5:0>	ะ	<5:3:0>
		ย	<6:3:0>
แ	<3:1:0>	ำ	<7:3:1>
ห	<4:1:0>, <4:5:0>	เ	<4:4:0>
ล	<5:1:1>	บ	<5:4:0>
เ	<1:2:0>, <4:4:0>	ื	<6:4:0>
ฮ	<2:2:0>	ั	<7:4:0>
ง	<3:2:0>	ก	<9:4:1>
ซ	<4:2:0>	น	<5:5:0>
ว	<5:2:0>	า	<6:5:0>
ย	<6:2:1>	โ	<8:5:0>
ไ	<1:3:0>, 1:4:0>,1:5:0>	ง	<9:5:0>
้	<3:3:0>, 3:4:0>,3:5:0>,7:5:0>	่	<10:5:1>

นิยามที่ 5 ตารางรายการผกผัน (IVL Table) คือตารางที่นำดัชนีอักขระของนิยามที่ 4 มาสร้างในรูปแบบตารางสองคอลัมน์ คือ Σ และ รูปแบบของแต่ละ $\mathbf{c}$ ที่มีรูปแบบการแทนดัชนีในรูปแบบ $\mathbf{c}$: <pos : terminated : { I_0 และ/หรือ I_1}> โดยที่ I_0 และ I_1 คือเซตของรายการปรากฏของอักขระ $\mathbf{c}$ ใน w_j ใดๆ I_0 ที่ไม่ใช่อักขระสุดท้ายใน w_j ส่วน I_1 คือเซตของรายการปรากฏของอักขระ $\mathbf{c}$ ใน w_j ใดๆ ที่เป็นอักขระสุดท้ายใน w_j ใดๆ

ตัวอย่างที่ 5 จากชุดของคำไม่สุภาพในตัวอย่างที่ 1 W = <ตอแหล, เฮงซวย, ไอ้ระยำ, ไอ้เบื๊อก, ไอ้หน้าโง่ > และดัชนีอักขระคำไม่สุภาพในตัวอย่างที่ 1 เขียนด้วยตารางรายการผกผัน IVL Table ได้ดังตารางที่ 3

ตารางที่ 3 ตัวอย่างตารางผกผันของคำไม่สุภาพในตัวอย่างที่ 5

Σ	<pos:terminated:{l_0 และ/หรือ l_1}>	Σ	<pos:terminated:{l_0 และ/หรือ l_1}>
ต	<1:0:{1}>	ห	<4:0:{1, 5}>
อ	<2:0:{1, 3, 4, 5}>, <8:0:{4}>	ล	<5:1:{1}>
แ	<3:0:{1}>	เ	<1:0: {2}>, <4:0: {4}>
ฮ	<2:0:{2}>	เ	<4:0:{4}>
ง	<3:0:{2}>	บ	<5:0:{4}>
ซ	<4:0:{2}>	ี	<6:0:{4}>
ว	<5:0:{2}>	ั	<7:0:{4}>
ย	<6:1:{2}>	ก	<9:1:{4}>
ไ	<1:0:{3, 4, 5}>	น	<5:0:{5}>
้	<3:0: {3,,4, 5}>, <7:0: {5}>	า	<6:0:{5}>
ร	<4:0:{3}>	โ	<8:0:{5}>
ะ	<5:0:{3}>	ง	<9:0:{5}>
ย	<6:0:{3}>	่	<10:1:{5}>
ำ	<7:1:{3}>		

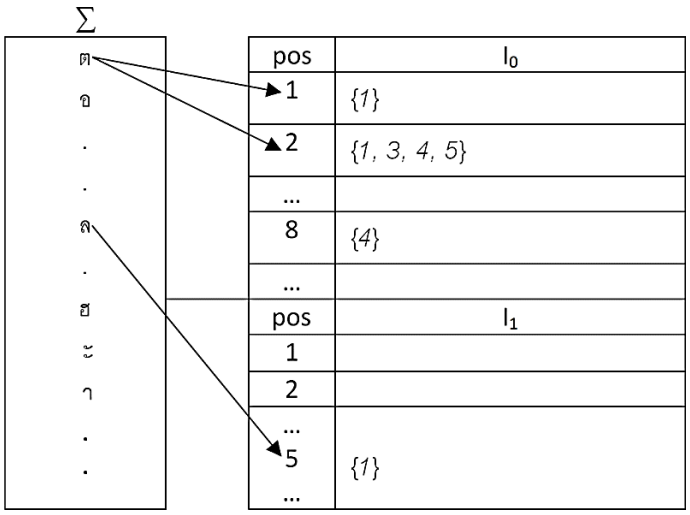
วิธีการดำเนินการวิจัย

ขั้นตอนวิธีการสร้างพจนานุกรม

เบื้องต้นก่อนการอธิบายถึงขั้นตอนการสร้างพจนานุกรม จำเป็นต้องกำหนดนิยามเฉพาะขึ้นเพื่อใช้อธิบายลำดับขั้นการสร้างในลำดับถัดไป โดยนิยามที่ 6 ต่อไปนี้แสดงรายละเอียดดังกล่าว

นิยามที่ 6 พจนานุกรม HT-Dictionary คือ พจนานุกรมการแฮชที่สร้างจากตารางการแฮชแบบสมบูรณ์ (Perfect hashing table) ใช้บรรจุรายการผกผันจากตารางรายการผกผัน (IVL Table) ประกอบด้วยสองระดับการแฮช โดยระดับแรก ใช้คีย์ f(**ב**) เข้าถึง pos และ ใช้ f(pos) เข้าถึง l_0 และ/หรือ l_1 ในระดับที่สองของตารางการแฮช

ตัวอย่างที่ 6 จากตัวอย่างที่ 5 สามารถเขียนให้อยู่ในรูปแบบพจนานุกรม HT-Dictionary ดังแนวทางดังรูปที่ 3 ต่อไปนี้



รูปที่ 3 แสดงตัวอย่างการสร้างพจนานุกรมคำไม่สุภาพแบบรายการผกผันพหุแบบ

ทฤษฎีบท 1 การเข้าถึง I_0 และ/หรือ I_1 ใดๆ ใน HT-Dictionary ทำได้ด้วยความซับซ้อน O(1)

พิสูจน์ กำหนด f(**ƈ**) เข้าถึง pos ในตารางการแฮชระดับที่ 1 และ ใช้ f(pos) เข้าถึง I_0 และ/หรือ I_1 ของตารางการแฮชในระดับที่ 2

เนื่องจากการเข้าถึงตารางการแฮชใดๆ ทำให้ f(**ƈ**) เข้าถึง pos ในตารางการแฮชระดับที่ 1 ด้วยความซับซ้อน O(1) และจากนั้น f(pos) เข้าถึง I_0 และ/หรือ I_1 ของตารางการแฮชในระดับที่ 2 ก็ยังคงใช้ความซับซ้อน O(1) ดังนั้นการเข้าถึง I_0 และ/หรือ I_1 ใดๆ ใน HT-Dictionary คือ O(1) + O(1) ซึ่งจะได้ค่าความซับซ้อน O(1)

ขั้นตอนวิธีสร้างพจนานุกรม HT-Dictionary แสดงดังนี้

Algorithm 1: Creating HT-Dictionary

Input: $W = \{w_1, w_2, w_3, ..., w_r\}$, **Σ**

Output: HT-Dictionary

1. Create empty HT-Dictionary
2. For I = 1 To r Do
3. For j = 1 to m of w_i Do
4. If **ƈ**$_j$ does not exist in HT-Dictionary Then
5. Store **ƈ** as a new key,and j as pos in HT-Dictionary Level 1
6. End of If
7. Create I_0 if j < m of w_i or Create I_1 if j = m of w_i
8. Store I_0 or I_1 to HT-Dictionary Level 2
9. End of If
10. End of For
11. End of For
12. Return HT-Dictionary

ทฤษฎีบท 2 การสร้างพจนานุกรม HT-Dictionary มีความซับซ้อนด้านเวลา O(|W|)

พิสูจน์ จาก $W = \{w_1, w_2, w_3, ..., w_r\}$ แต่ละ w_j มีความยาว m_i ซึ่ง $m_1 + m_2 + m_3 + ... + m_r = |W|$

การสร้างพจนานุกรมว่าง (บรรทัดที่ 1) ทำด้วยความซับซ้อนค่าคงที่ O(1)

ลูปนอก (บรรทัดที่ 2 - 11) จะวนเท่ากับ r รอบ ในขณะที่แต่ละ w_j จะวนเท่ากับจำนวนความยาวของแต่ละคำศัพท์ คือ m_j รอบ (บรรทัดที่ 3 – 10) ดังนั้น การสร้าง HT-Dictionary จะมีการดำเนินการด้วยค่าความซับซ้อนด้านเวลา $m_1 + m_2 + m_3 + ... + m_r = |P|$ ซึ่งความซับซ้อนเท่ากับ O(|P|)

นอกจากนั้น ระหว่างการดำเนินการในการสร้างคีย์ **ƈ** และตำแหน่งดัชนี pos (บรรทัดที่ 5) การสร้าง I_0 I_1 และการเพิ่มลงในพจนานุกรม (บรรทัดที่ 7 - 8) ต่างดำเนินการด้วยค่าความซับซ้อน O(1) ซึ่งเข้าถึงรายการผกผันด้วยความซับซ้อนค่าคงที่ O(1) ตามทฤษฎีบท 1

ดังนั้น ความซับซ้อนการสร้างพจนานุกรม HT-Dictionary เท่ากับ O(|P|)

ทฤษฎีบท 3 พจนานุกรม HT-Dictionary มีความซับซ้อนด้านเนื้อที่ O(|**Σ**|+|W|)

พิสูจน์ ด้วยการพิจารณา **Σ** ซึ่งบรรจุ **ƈ** ที่จำนวนมากที่สุดเท่ากับ **Σ** ในขณะที่มีการสร้าง I_0 I_1 ของ $w_1, w_2, w_3, ..., w_r$ เนื่องจาก **ƈ** ∈ **Σ** ซึ่งขนาดของการจัดเก็บในพจนานุกรม HT-Dictionary จำนวน **ƈ** ทั้งหมดที่จำเป็นจัดเก็บใน HT-Dictionary คือ |**Σ**|

ในขณะที่แต่ละ l_0 l_1 ที่ถูกสร้างขึ้นในแต่ละรายการจาก $w_1, w_2, w_3, ..., w_r$ ด้วยจำนวน l_0 l_1 เท่ากับ $m_1 + m_2 + m_3 + ... + m_r = |W|$ ซึ่งความซับซ้อนเท่ากับ O(|W|) ในขณะที่ตำแหน่ง pos ซึ่งเท่ากับความยาวมากที่สุดของคำศัพท์ซึ่งถือเป็นค่าคงที่

ดังนั้น ความซับซ้อนด้านเนื้อที่ของพจนานุกรม HT-Dictionary เท่ากับ O(|Σ|+|W|)

ขั้นตอนวิธีการตรวจจับ

ขั้นตอนวิธีการตรวจจับ ขั้นตอนวิธีเริ่มจากรับอักขระเข้ามาทีละอักขระจากข้อความ ($T = t_1, t_2, t_3 ... t_n$) นำอักขระนั้นไปเปรียบเทียบในพจนานุกรมตารางรายการผกผัน หากพบอักขระในพจนานุกรมตารางรายการผกผัน ขั้นตอนวิธีการตรวจจับคำไม่สุภาพ จะดำเนินการได้หลังจากสร้างพจนานุกรมเสร็จสิ้นแล้ว เพื่อให้เข้าใจขั้นตอนวิธีได้ชัดเจนมากขึ้น จึงกำหนดตัวแปรที่จำเป็นสำหรับขั้นตอนวิธีส่วนการค้นหา ได้แก่ N คือ ตัวนำทางเพื่อชี้ตำแหน่งการเปรียบคู่ปัจจุบัน pos คือ ตัวเก็บตำแหน่งของรายการผกผันที่ต้องการนำมาเปรียบคู่ TN คือ อักขระจากข้อความตำแหน่งที่เปรียบคู่ปัจจุบัน SET1 กับ SET2 คือ ตัวแปรชั่วคราวเก็บเซต l_1 l_0 ระหว่างการตรวจจับ คือ ฟังก์ชันที่หาผลการอินเตอร์เซกชันระหว่างเซตของ SET1 และ SET2

ขั้นตอนวิธีเริ่มจากกำหนดค่าเริ่มต้นให้กับ SET1 และ SET2 และกำหนดค่า min คือค่าความยาวของคำที่สั้นที่สุดใน W หลังจากนั้นขั้นตอนวิธีทำงานโดยกวาด (Scan) เพื่อเปรียบเทียบความต่อเนื่องของ l_1 l_0 ที่นำมาจาก HT-Dictionary ตามตำแหน่ง pos ของ $T[t_N]$ ซึ่งแต่ละครั้งของการตรวจจับจะเก็บรายการผกผัน l_1 หรือ l_0 ลงตัวแปรชั่วคราว SET1 หรือ SET2 แล้วนำรายการผกผันดังกล่าวมาอินเตอร์เซกชันกันเพื่อหาความต่อเนื่องของอักขระแบบและผลการตรวจจับสำเร็จ ดำเนินการกวาดไปแต่ละหน้าต่างการค้นหา จนกระทั่งถึงหน้าต่างที่ n-min นอกจากนั้น จำเป็นต้องกำหนดค่า d ให้เป็นค่าช่องว่าง (White space) ที่อนุญาตให้มีได้ในแต่ละคำระหว่างการตรวจจับ และ f คือ จำนวนช่องว่างที่ตรวจพบระหว่างตรวจจับ ซึ่งขั้นตอนวิธีการค้นหา แสดงดังนี้

Algorithm2: Impolite Detection

Input: HT-Dictionary, min, l_1, l_2, $T = t_1 t_2 t_3 ... t_n$, d

Output: Impolite word(s) is reported.

1. N = 1, pos = 1, SET1 = SET2 = null, RESULTS = {}, f = 0
2. SET1 ⟵ (l_0 or l_1 of $T[t_N]$ in HT-Dictionary with pos), N++
3. While (N <= n-min) Do
4. Report the matched position if SET1 contains l_1
5. If SET1 ≠ ∅ Then
6. pos + +
7. SET2 ⟵ (l_0 or l_1 of $T[t_N]$ in HT-Dictionary with pos)
8. SET1 ⟵ SET1 ∩ SET2
9. Else
10. pos = 1 or if $T[t_N]$ = whitespace and f<d then f++/else f = 0
11. SET1 ⟵ (l_0 or l_1 of $T[t_N]$ in HT-Dictionary with pos)
12. End of If
13. N++
14. End of While
15. Retur

ตัวอย่างที่ 7 การตรวจจับคำไม่สุภาพ เมื่อข้อความ T = ไอ้หน้าโง่ โดยใช้พจนานุกรมคำไม่สุภาพตารางที่ 3 ลำดับขั้นตอนตรวจจับ แสดงเป็นลำดับดังนี้

1. ค้นอักขระ โดยใช้ “ไ” แล้ว SET1 ไม่ใช่ ∅ ได้ผลลัพธ์คือ “ไ” = <{1 : 0 : {3, 4, 5}>

หมายความว่า “ไ” ตรงกับคำที่ไม่สุภาพ โดยอยู่ในตำแหน่งที่หนึ่งของคำ ยังไม่จบคำ (Terminated = 0) และพบในคำที่ 3, 4 และ 5 ดังนั้นสามารถค้นอักขระตัวถัดไป โดยเก็บผลลัพธ์ของ “ไ” ไว้ใน SET1

2. ค้นอักขระ โดยใช้ “อ” ซึ่งได้ผลลัพธ์คือ “อ” = <2 : 0 : {1, 3, 4, 5}> เก็บผลลัพธ์ไว้ใน SET2

โดย SET1 ⟵ SET1 ∩ SET2 = <2 : 0 : {3, 4, 5} หลังจากนั้น SET1 ⟵ SET2 ≠ ∅ และยังไม่จบคำ

3. ค้นอักขระ โดยใช้ “ ้ ” ซึ่งได้ผลลัพธ์คือ “ ้ ” = <{3 : 0 : {3, 4, 5}> เก็บผลลัพธ์ไว้ใน SET2

โดย SET1 ⟵ SET1 ∩ SET2 = <3 : 0 : {3, 4, 5} จะได้ว่า SET1 ⟵ SET2 ≠ ∅ และยังไม่จบคำ

4. ค้นอักขระ โดยใช้ “ห” ซึ่งได้ผลลัพธ์คือ “ห” = <{4 : 0 : {1, 5}> เก็บผลลัพธ์ไว้ใน SET2

โดย SET1 ⟵ SET1 ∩ SET2 = <4 : 0 : {5}> จะได้ว่า SET1 ⟵ SET2 ≠ ∅ และยังไม่จบคำ

5. ค้นอักขระ โดยใช้ “น” ซึ่งได้ผลลัพธ์คือ “น” = <{5 : 0 : {5}> เก็บผลลัพธ์ไว้ใน SET2

โดย SET1 ⟵ SET1 ∩ SET2 = <5 : 0 : {5} > จะได้ว่า SET1 ⟵ SET2 ≠ ∅ และยังไม่จบคำ

6. ค้นอักขระ โดยใช้ “า” ซึ่งได้ผลลัพธ์คือ “า” = <{6 : 0 : {5}> เก็บผลลัพธ์ไว้ใน SET2

โดย SET1 SET1 ⟵ SET1 ∩ SET2 = <6 : 0 : {5}> จะได้ว่า SET1 ⟵ SET2 ≠ ∅ และยังไม่จบคำ

7. ค้นอักขระ โดยใช้ “ ้ ” ซึ่งได้ผลลัพธ์คือ “ ้ ” = {7 : 0 : {5}> เก็บผลลัพธ์ไว้ใน SET2

โดย SET1 SET1 ⟵ SET1 ∩ SET2 = <7 : 0 : {5}> จะได้ว่า SET1 ⟵ SET2 ≠ ∅ และยังไม่จบคำ

8. ค้นอักขระ โดยใช้ “ โ ” ซึ่งได้ผลลัพธ์คือ “ โ” = {8 : 0 : {5}> เก็บผลลัพธ์ไว้ใน SET2

โดย SET1 SET1 ⟵ SET1 ∩ SET2 = <8 : 0 : {5}> จะได้ว่า SET1 ⟵ SET2 ≠ ∅ และยังไม่จบคำ

9.ค้นอักขระ โดยใช้ “ง” ซึ่งได้ผลลัพธ์คือ “ง” = {9 : 0 : {5}> เก็บผลลัพธ์ไว้ใน SET2

โดย SET1 ⟵ SET1 ∩ SET2 = <9 : 0 : {5}> จะได้ว่า SET1 ⟵ SET2 ≠ ∅ และยังไม่จบคำ

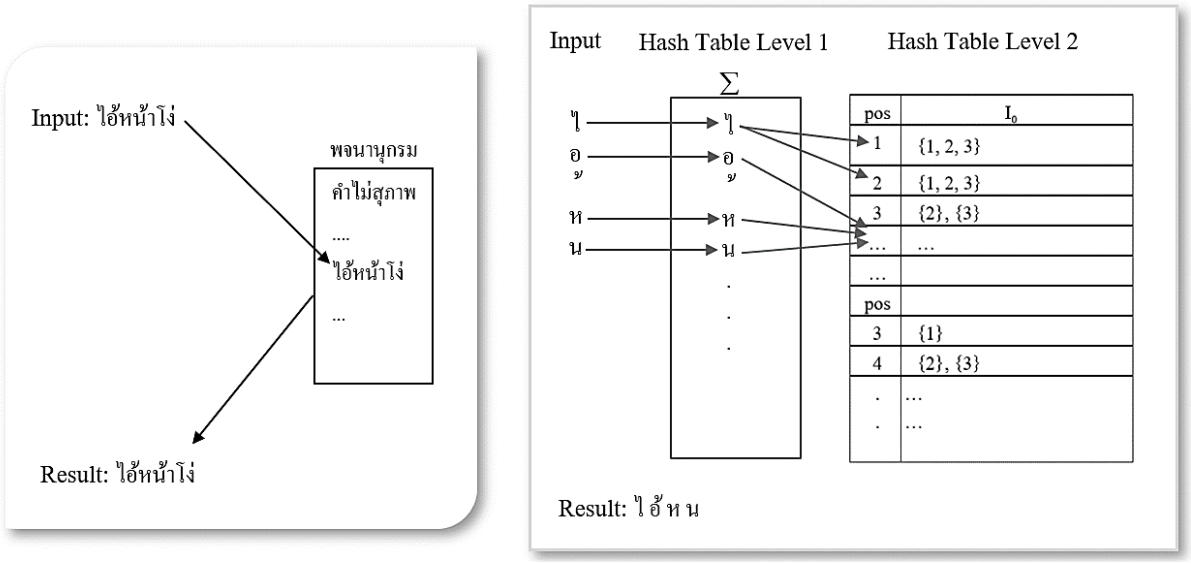
10. ค้นอักขระ โดยใช้ “ ่ ” ซึ่งได้ผลลัพธ์คือ “ ่ ” = {10 : 1 : {5}> ซึ่งค่า Terminated = 1 หมายความว่าจบคำ

ดังนั้น ค้นพบคำไม่สุภาพ คือ W5 = “ไอ้หน้าโง่”

ตัวอย่างที่ 8 แสดงข้อได้เปรียบของพจนานุกรมแบบใหม่นี้ ที่นำไปสู่การทำนายคำไม่สุภาพ และการตรวจจับแบบออนไลน์ที่ต้องมีการตรวจจับการป้อนข้อมูลแบบปัจจุบันทันด่วน เมื่อมีการป้อนข้อมูลคำว่า “ไอ้หน้าโง่”

ถ้าเป็นการตรวจจับแบบเดิม ผู้ป้อนข้อมูลจะอ่านจากข้อความที่ป้อนจนครบ แล้วตรวจจับด้วยการนำคำดังกล่าวไปค้นกับพจนานุกรมที่มีอยู่แบบคำต่อคำ แต่หากเป็นพจนานุกรมแบบใหม่ เมื่อผู้ป้อนข้อมูลป้อนแต่ละอักขระสารมารถตรวจจับได้ทีละอักขระ ทำให้สามารถตรวจจับได้ตั้งแต่ข้อความ “ไอ้” หรือ “ไอ้หน้” เปรียบเทียบกลไก การทำงานของขั้นตอนวิธีแบบเดิม และแบบใหม่ตามแนวทางของงานวิจัยนี้ ดังรูปที่ 4 เป็นต้น

แนวทางการตรวจจับแบบปกติ **แนวทางการตรวจจับจากงานวิจัยใหม่**



รูปที่ 4 เปรียบเทียบแนวคิดการตรวจจับตามแนวทางพจนานุกรมแบบเก่าและแบบใหม่ที่พัฒนาขึ้นจากงานวิจัย

สำหรับความซับซ้อนด้านเวลาของการค้นหามากสุดคือ O(n) เมื่อ n คือขนาดของข้อความที่ต้องการค้น เบื้องต้นก่อนการพิสูจน์ทฤษฎีบทจำเป็นต้องอธิบายบทแทรก เพื่อแสดงการเข้าถึง ตัวแปร SET1, SET2 และการอินเทอร์เซคชัน ดังนี้

บทแทรก 1 กำหนดให้ SET คือตารางการแฮชย่อยที่มี $f(t_k)$ มีความซับซ้อน O(1)

พิสูจน์ กำหนด SET เป็นตารางการแฮช เช่นเดียวกับ ทฤษฎีบท 1 ซึ่งมีคีย์ t_k และ I_0 หรือ I_1

ดังนั้นจะใช้ $f(t_k)$ เพื่อเข้าถึง I_0 หรือ I_1 ด้วยความซับซ้อน O(1) ตามทฤษฎีบท 1

บทแทรก 2 การนำรายการผกผันจาก HT-Dictionary ที่ตรงกับตำแหน่ง pos ของอักขระ $T[t_N]$ ลง SET ใดๆ ใช้ความซับซ้อน O(1)

พิสูจน์ กำหนดให้ t_N คือ อักขระจากข้อความ T ที่แปลงเป็นเป็นคีย์เพื่อเข้าถึง I_0 หรือ I_1 ได้

ดังนั้น การเข้าถึง I_0 หรือ I_1 ใน HT-Dictionary ด้วยตำแหน่ง pos จะใช้ความซับซ้อน O(1) ตามทฤษฎีบท 1 และนำ I_0 หรือ I_1 ลง SET ด้วย O(1) ตามบทแทรก 1

บทแทรก 3 การอินเตอร์เซกชันระหว่าง SET1 และ SET2 ใช้ความซับซ้อน O(1)

พิสูจน์ กำหนดให้ SET1 และ SET2 คือ SET ตามบทแทรก 1 โดยที่

SET1 บรรจุ I_0 และ/หรือ I_1

SET2 บรรจุ I_0 และ/หรือ I_1

การเข้าถึงเพื่อนำ I_0 หรือ I_1 มาเปรียบเทียบสำหรับการอินเทอร์เซกชัน ใช้ O(1) เป็นไปตามบทแทรก 1

ทฤษฎีบท 4 ขั้นตอนวิธีการตรวจจับคำไม่สุภาพด้วย Algorithm 2 มีความซับซ้อน O(n)

พิสูจน์ กำหนด n คือความยาวของข้อความนำเข้า T ซึ่ง $T = t_1, t_2, t_3, ..., t_n$

อ้างถึง Algorithm 2 เริ่มจากบรรทัดที่ 1 เป็นการกำหนดค่า ใช้ความซับซ้อน O(1)

บรรทัดที่ 2 7 และ 11 เข้าถึง I_0 หรือ I_1 นำเข้า SET1 ด้วยความซับซ้อน O(1) ตามทฤษฎีบท 1 และบทแทรก 2

การอินเทอร์เซกชันในบรรทัดที่ 8 ทำงานด้วยความซับซ้อน O(1) ตามบทแทรก 3 ในขณะที่ลูปหลัก (บรรทัดที่ 3 - 14) ที่วนดำเนินการมากที่สุด n ครั้ง O(n) ซึ่งทำให้ความซับซ้อนของส่วนการตรวจจับจะเท่ากับ O(n) ในกรณีแย่สุด (ตรวจจับพบ) หรือหากกรณีที่สุดจะทำงานที่ O(n-min) ในกรณีดีสุด (ข้อความไม่ใช่คำไม่สุภาพ)

การทดลองและการประเมินผล

การประเมินผลเชิงทฤษฎี

พจนานุกรมที่พัฒนาขึ้นนี้เป็นแนวคิดใหม่ที่แตกต่างจากรูปแบบการจัดเก็บพจนานุกรมคำไม่สุภาพแบบทั่วไปที่เคยมีมาก่อน พจนานุกรมแบบนี้ต้องการให้สามารถตอบสนองต่อสิ่งต่อไปนี้ได้เป็นประเด็นดังนี้

1) ด้านความซับซ้อน พจนานุกรมและขั้นตอนวิธีนี้สามารถประยุกต์สู่การตรวจจับคำไม่สุภาพได้แบบทันทีทันใดที่อักขระถูกป้อนเข้ามาในระบบ โดยไม่จำเป็นต้องให้เกิดการป้อนจนสำเร็จเสียก่อนสร้างพจนานุกรมได้รวดเร็ว ด้วยค่าความซับซ้อนด้านเวลา O(|W|) และค่าความซับซ้อนด้านเนื้อที่จัดเก็บ O(|Σ |+|W|) ขณะที่การตรวจจับมากที่สุดใช้เวลา O(n) ทั้งนี้แนวคิดการแทนตำแหน่งของคำนี้ยังสามารถตรวจจับคำที่มีขนาดยาวๆ โดยประยุกต์การเข้าถึงตำแหน่งต่างๆ ที่แตกต่างกันโดยไม่จำเป็นต้องพิจารณาตามลำดับก็ได้

2) จุดเด่นสามารถค้นหาได้แบบอักขระต่ออักขระที่ป้อนเข้ามา ตอบสนองการค้นหาแบบออนไลน์ได้แบบทันทีทันใดเข้าถึงแบบเป็นอิสระหรือสามารถนำไปสู่การเข้าถึงแบบขนานได้

3) พจนานุกรมแบบใหม่นี้นำไปสู่การทำนายคำไม่สุภาพได้ตั้งแต่ข้อความอักขระเริ่มต้น

4) ด้านความยืดหยุ่นของพจนานุกรม เนื่องจากสืบทอดแนวคิดมาจากพจนานุกรมแบบพลวัต ดังนั้นพจนานุกรมนี้จึงสามารถเพิ่ม ลบ แก้ไขคำต่างๆ ในพจนานุกรมได้แบบพลวัตได้อีกด้วย

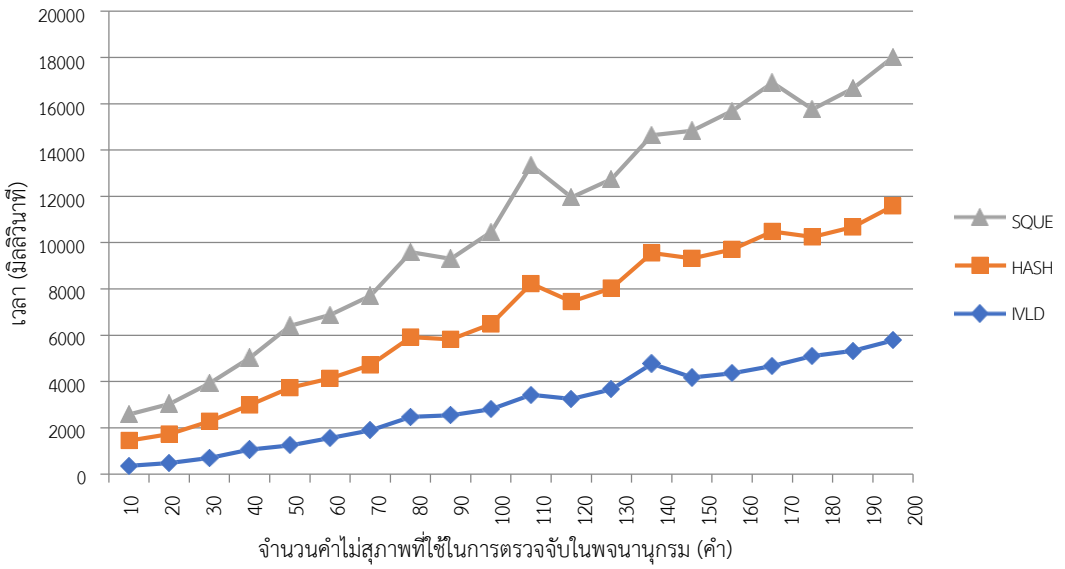
ชุดข้อมูลและการเตรียมการทดลอง

การทดลองวิเคราะห์คำไม่สุภาพจาก 1) กระทู้ในเว็บบอร์ดภาษาไทย 2) เว็บไซต์ที่ระบุคำศัพท์ไม่สุภาพ 3) เว็บกฎหมาย 4) บทความวิจัยและงานวิจัย 5) วิทยานิพนธ์ที่เกี่ยวกับคำไม่สุภาพ 7) เฟซบุ๊ก และอื่นๆ ด้วยขนาดรวมกัน 62.9 MB จากนั้นสกัดและคัดเลือกคำไม่สุภาพนำเข้าสู่พจนานุกรม จำนวน 233 คำ

จากนั้นพัฒนาโปรแกรมด้วยโปรแกรมภาษาจาวา jdk 21, Netbeans 20 ทดลองด้วยเครื่องคอมพิวเตอร์ Dell Vostro 5400 Windows 10 Pro Intel(R) Core(TM) i7-7700HQ RAM 16.0 GB พัฒนาโปรแกรมและนำเข้าข้อมูลสู่พจนานุกรม เขียนโปรแกรมสุ่มคำจากพจนานุกรมจำนวน 10 - 200 คำ นำเข้าสู่อินพุตโปรแกรมและเขียนโปรแกรมจากขั้นตอนวิธีที่ออกแบบไว้ และพัฒนาโปรแกรมที่ตรวจจับแบบตามลำดับและกลวิธีการแฮชพร้อมวัดผลความถูกต้องเปรียบเทียบกับเทคนิคการตรวจจับโดยใช้ไฟไนท์สเตทแมชชีน (วุฒิชัย, 2549) และ เทคนิคดาต้าไมน์นิง (ณัฐาศิริและสมชาย, 2560) จำนวน 5 ขั้นตอนวิธี

การประเมินผลเชิงการทดลอง

ผลการทดลองพัฒนาโปรแกรมสำหรบตรวจนับตามขั้นตอนวิธีและพจนานุกรมที่ออกแบบไว้ นำเข้าคลังจากคำศัพท์ 233 คำ และนำผลการทดลองมาพัฒนาโปรแกรมสำหรับวัดเวลาเข้าถึงและตรวจจับของพจนานุกรมแบบเก่า ด้วยเทคนิคการตรวจจับตามลำดับ (Sequential Search: SQUE) การตรวจจับแบบแฮชชิ่ง (Hashing Search: HASH) โดยรวมเวลาของการตัดคำภาษาไทยก่อนการตรวจจับ เทียบกับพจนานุกรมแบบใหม่ (IVLD) แสดงดังรูปที่ 5 โดยที่แกนตั้งแสดงเวลาในหน่วยมิลลิวินาที แกนนอนแสดงจำนวนคำไม่สุภาพที่ใช้ในการตรวจจับในพจนานุกรม



รูปที่ 5 ผลเปรียบเทียบเวลาตรวจจับด้วยพจนานุกรมแบบเก่าและพจนานุกรมใหม่ที่พัฒนาขึ้นจากงานวิจัย

จากกราฟในรูปที่ 5 แสดงให้เห็นว่าในทุกๆ ปริมาณของคำไม่สุภาพที่นำมาเปรียบเทียบการตรวจจับ ขั้นตอนวิธีโดยอาศัยพจนานุกรมแบบใหม่ สามารถทำงานได้รวดเร็วและมีประสิทธิภาพมากกว่าขั้นตอนการใช้พจนานุกรมที่จัดเก็บแบบเป็นคำอย่างชัดเจนและมีนัยสำคัญ

ผลการทดลองพบว่า ประสิทธิภาพตรวจจับความถูกต้อง ได้พัฒนาโปรแกรมสำหรับตรวจจับ โดยสุ่มคำไม่สุภาพจากพจนานุกรมและนำเข้าข้อมูลทดลองจากข้อมูลต้นฉบับที่นำมาสกัดคำไม่สุภาพ โดยจากการค้นสุ่มเอกสารข้อความที่นำมาใช้สกัดคำไม่สุภาพ และสุ่มและแทรกคำไม่สุภาพแทรกในพจนานุกรมเข้าไปในเอกสารตามปริมาณดังในรูปที่ 5 ผลการทดลองตรวจนับคำไม่สุภาพจะได้ค่าความถูกต้องร้อยละ 100 ของข้อมูลที่ใช้ทดลอง ผลการแทรกคำหยาบเข้าไปกับข้อมูลจริงที่นำมาทดลองพบว่า ตรวจจับได้ร้อยละ 100 มีค่าความคลาดเคลื่อนการทำนายคำอยู่ที่ร้อยละ 0.00 ทั้งยังได้เทียบกับเทคนิคแบบพจนานุกรมแบบเดิมที่ได้ค่าการตรวจจับไม่ผิดพลาด (SQUE และ HASH) อีกด้วย นอกจากนั้น เพื่อให้เห็นความแตกต่างกับ เทคนิคแบบอื่นที่เคยศึกษามา จึงได้นำผลการเปรียบเทียบผลการทดลองของ ณัฐาศิริและสมชาย (2560) วุฒิชัย (2549) มาเปรียบเทียบให้เห็น โดยกำหนดให้เทคนิคดาต้าไมน์นิ่ง (Data mining) ใน ณัฐาศิริและสมชาย (2560) ได้แก่ ต้นไม้ตัดสินใจ (Decision Tree) เทคนิคนาอีฟเบย์ (Naïve Bayes) และเทคนิคเคเนียร์เรสเนเบอร์ (K-Nearest Neighbors) ขณะที่ วุฒิชัย (2549) คือ เทคนิคการวิจัยด้วยไฟไนท์สเตทแมชชีน (Finite State Machine) แสดงดังตารางที่ 4

ตารางที่ 4 แสดงผลการทดลองจากความถูกต้อง (Accuracy) ของการตรวจจับ

ขั้นตอนวิธี	ร้อยละของความถูกต้อง	ร้อยละของความคลาดเคลื่อน
IVLD	100	0.00
SQUE	100	0.00
HASH	100	0.00
Decision Tree	96	0.43
Naïve Bayes	96	0.44
K-Nearest Neighbors	95	0.46
Finite State Machine	99.14	8.36

สรุปผลการวิจัยและข้อเสนอแนะ

งานวิจัยนี้จึงนำเสนอแนวคิดใหม่สำหรับตรวจจับคำไม่สุภาพและคำหยาบคายด้วยการพัฒนาโครงสร้างข้อมูลพจนานุกรมจัดเก็บคำไม่สุภาพแบบใหม่อาศัยรูปแบบอักขระเดี่ยวแทนการใช้คำแบบเดิมในพจนานุกรมแบบเดิม ใช้วิธีจำแนกรายละเอียดของแต่ละคำเป็นสำคัญออกเป็นรายการผกผันพหุแบบ เก็บในพจนานุกรมเฉพาะที่ใช้หลักการแฮชแบบสมบูรณ์สำหรับการเข้าถึง จากนั้นพัฒนาขั้นตอนวิธีตรวจจับคำดังกล่าวซึ่งสามารถตรวจจับคำไม่สุภาพได้แบบทันทีทันใดที่อักขระถูกป้อนเข้ามาในระบบ โดยไม่จำเป็นต้องให้เกิดการป้อนจนสำเร็จเสียก่อนจึงจะตรวจพบเช่นขั้นตอนวิธีเดิมที่เคยมีมา สร้างพจนานุกรมด้วยค่าความซับซ้อนด้านเวลา $O(|W|)$ และค่าความซับซ้อนด้านเนื้อที่จัดเก็บ $O(|\Sigma|+|W|)$ โดยที่ W คือจำนวนคำที่บรรจุในพจนานุกรม ในขณะที่ความซับซ้อนด้านเวลาการค้นหา คือ $O(n)$ เมื่อ n คือความยาวของข้อความที่ป้อนเข้าสู่ระบบ ผลการทดลองพัฒนาโปรแกรมสำหรับตรวจนับตามขั้นตอนวิธีและพจนานุกรมที่ออกแบบไว้ นำเข้าคลังจากคำศัพท์ 233 คำ ผลการทดลองพบว่า พจนานุกรมแบบใหม่สามารถตรวจจับคำไม่สุภาพโดยไม่จำเป็นต้องมีการตัดคำก่อนทำให้สามารถตรวจจับได้อย่างรวดเร็วกว่าการจัดเก็บและเข้าถึงข้อมูลคำไม่สุภาพแบบเดิมอย่างมีนัยสำคัญ ผลการทดลองวัดค่าความถูกต้องของการตรวจจับคำไม่สุภาพที่มีในพจนานุกรมได้ร้อยละ 100 ของข้อมูลที่ใช้ทดลอง ผลการแทรกคำหยาบเข้าไปกับข้อมูลจริงที่นำมาทดลองพบว่า ตรวจจับได้ร้อยละ 100 โดยไม่มีค่าความคลาดเคลื่อนในการตรวจจับ

ข้อเสนอแนะของการต่อยอดงานวิจัยนี้ คือ การสร้างให้พจนานุกรมสามารถเชื่อมโยงการตรวจจับได้ด้วยบริบทของข้อมูลหรือด้วยความสัมพันธ์และเชื่อมโยงในมิติรูปภาพหรือภาษาถิ่น โดยการขยายลำดับชั้นของตารางการแฮชในพจนานุกรมให้มีระดับเพิ่มขึ้น ในขณะที่พจนานุกรมยังเป็นแบบพลวัตได้ด้วยการเพิ่มคำไม่สุภาพใหม่เข้าไปในพจนานุกรมได้อีกด้วย

เอกสารอ้างอิง

ณัฐาศิริ เชาว์ประสิทธิ์ และสมชาย เล็กเจริญ. (2560). การพัฒนาโมเดลตรวจจับคำหยาบภาษาไทยบนสื่อออนไลน์ด้วยเทคนิคดาต้าไมน์นิง. ใน: การประชุมนำเสนอผลงานวิจัยระดับบัณฑิตศึกษา ครั้งที่ 12 ปีการศึกษา 2560. มหาวิทยาลัยรังสิต, กรุงเทพฯ. 1432 - 1441.

เถกิง พันธุ์เถกิงอมร. การใช้ภาษาในการเขียนเชิงวิชาการ. (2558). เอกสารประกอบการบรรยายในโครงการแลกเปลี่ยนเรียนรู้ด้านการเรียนการสอน ด้านการวิจัยและด้านการสนับสนุนวิชาการ สำหรับอาจารย์ คณะมนุษยศาสตร์และสังคมศาสตร์, มหาวิทยาลัยราชภัฏสงขลา. สงขลา.

นภัทร อังกูรสินธนา และปนันดา เลอเลิศยุติธรรม. (2562). กลวิธีความไม่สุภาพทางภาษาในภาษาไทย. วารสารปาริชาต 32(2): 63 - 74.

นันทวัฒน์ เนตรเจริญ และสุวัฒนา เลี่ยมประวัติ. (2558). การวิเคราะห์ข้อบกพร่องในการแปลถ้อยคำต้องห้ามของ คำระวี-ใบเตยในวรรณกรรมเรื่อง The Catcher in the Rye และแนวทางแก้ไข. ใน: การประชุมหาดใหญ่วิชาการระดับชาติ ครั้งที่ 6. มหาวิทยาลัยหาดใหญ่, สงขลา. 396 - 406.

ยุวดี พนมพรสุวรรณ. (2549). ปัญหานักเรียนพูดจาไม่ไพเราะเหมาะสม ระดับชั้น ปวช. 1/1 แผนกวิชาการตลาด. ใน: รายงานการวิจัยในชั้นเรียน, วิทยาลัยเทคนิคราชบุรี. ราชบุรี.

วิภาพรรณ แจ้งจร. (2549). วิธีการหลีกเลี่ยงการใช้คำไม่สุภาพของผู้ใช้อินเทอร์เน็ตบนกระดานสนทนา. มนุษยศาสตร์ปริทรรศน์ 28(1): 42 - 59.

วุฒิชัย วิเชียรไชย. (2549). การกรองคำหยาบในข้อความภาษาไทยโดยไฟไนท์สเตทแมชชีน. วิทยานิพนธ์วิทยาศาสตรมหาบัณฑิต (สถิติประยุกต์), สถาบันบัณฑิตพัฒนบริหารศาสตร์. กรุงเทพฯ: 127 หน้า.

Chakraborty, A., Joarda, S. and Sekh, A.A. (2023). Ensemble Classifier for Hindi Hostile Content Detection. ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP) 23(1): 11 – 17. doi: 10.1145/3591353.

Chakravarthi, B.R., Jagadeeshan, M.B., Palanikumar, V. and Priyadharshini R. (2023). Offensive language identification in dravidian languages using MPNet and CNN. International Journal of Information Management Data Insights 3(1): 1 - 18.

Chen, Z. and Feng, W. (2013). Detecting Impolite Crawler by Using Time Series Analysis. In: 2013 IEEE (25th International Conference on Tools with Artificial Intelligence. Herden, VA, United States. 123 - 126.

Dinh, V.H. (2020). Hash Table. Source: http://libetpan.sourceforge.net /doc/API/API/x161.html.Khancome. Retrieved from 15 October 2023.

Khancome, C. and Boonjing, V. (2009). Optimal Linear-time Multi-string Pattern Matching Algorithm. International Journal of Computational Science 3(6): 629 - 641.

Li, W. (2020). The language of bullying: Social issues on Chinese websites. Aggression and Violent Behavior 53(1): 101453.

Loudon K. (2020). Hash Tables. Source: www.oreilly.com/catalog/ masteralgoc/chapter/ch08.pdf. Retrieved from 15 October 2023.

Lusiana. Gemini, H., Efendi, Y. (2018). Filtering Impolite Words in Social Network Using Naïve Bayes Classifier. In: 2018 Third International Conference on Informatics and Computing (ICIC). Palembang, Indonesia. 1 - 5.

Magami, F. and Digiampietri, L.A. (2020). Automatic detection of depression from text data: A systematic literature review. In: SBSI 20: Proceedings of the XVI Brazilian Symposium on Information Systems. Association for Computing Machinery, New York, United States. 1 – 8.

Modha, S., Majumder, P., Mandl T. and Mandalia, C. (2020). Detecting and visualizing hate speech in social media: A cyber Watchdog for surveillance. Expert Systems with Applications 161: 1 - 11.

Moin, K., Shahzad, K. and Malik, M.K. (2021). Hate Speech Detection in Roman Urdu. ACM Transactions on Asian and Low -Resource Language Information Processing 20(1): 1 - 19.

Novitasari, S., Lestari, D.P., Sakti, S. and Purwarianti, A. (2018). Rude-Words Detection for Indonesian Speech Using Support Vector Machine. In: 2018 International Conference on Asian Language Processing (IALP), Bandung, Indonesia. 19 - 24. doi: 10.1109/IALP.2018.8629145.

Priya, P., Firdaus, M. and Ekbal, A. (2023). A multi-task learning framework for politeness and emotion detection in dialogues for mental health counselling and legal aid. Expert Systems with Applications 224: 1 - 17.

Tuarob, S., Satravisut, M., Sangtunchai, P., Nunthavanich, S. and Noraset, T. (2023) . FALCoN: Detecting and classifying abusive language in social networks using context features and unlabeled data. Information Processing & Management 60(4): 1 - 24.

Wikipedia. (2020). Hash table. Source: https://en.wikipedia.org/wiki/Hash_table. Retrieved from 15 October 2023.

❑❑❑❑❑