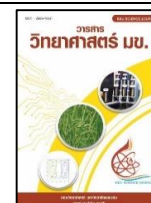




KKU SCIENCE JOURNAL

Journal Home Page : <https://ph01.tci-thaijo.org/index.php/KKUSciJ>

Published by the Faculty of Science, Khon Kaen University, Thailand



การประมวลผลภาษาธรรมชาติ เพื่อปรับปรุงคำผิดที่เกิดจากวิธีการรู้จำอักขระ Natural Language Processing to Improve the Errors Caused by the Optical Character Recognition

ทวีศักดิ์ คุ่มภักดี¹ สัจจาภรณ์ ไวจรรย์ยา^{1*} และ ณัฐโชติ พรหมฤทธิ์^{1*}Taweesak Khumphakdee¹, Sajjaporn Waijanya^{1*} and Nuttachot Promrit^{1*}¹ภาควิชาคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยศิลปากร วิทยาเขต พระราชวังสนามจันทร์ จังหวัดนครปฐม 73000¹Department of Computing, Faculty of Science, Silpakorn University, Nakhon Pathom, 73000, Thailand

บทคัดย่อ

บทความนี้นำเสนอการแก้ไขคำผิดที่เกิดขึ้นจากการรู้จำอักขระลายมือเขียนภาษาไทย การรู้จำอักขระลายมือเขียนภาษาไทยนั้นเป็นสิ่งที่ท้าทายเนื่องจากลายมือของคนแต่ละคนเขียนออกมาได้แตกต่างกัน ดังนั้นผลลัพธ์ที่ได้จากการรู้จำอักขระอาจเกิดผลลัพธ์ที่ผิดขึ้นได้แก่ คำที่อ่านออกเสียงไม่ได้หรือได้คำผิดที่ต้องมีการปรับให้ถูกต้อง โดยนำงานด้านการประมวลผลธรรมชาติมาปรับปรุงผลลัพธ์ที่ได้จากการรู้จำอักขระให้ดียิ่งขึ้น มีข้อมูลนำเข้า คือ ข้อความที่ได้จากการรู้จำอักขระลายมือเขียนภาษาไทยไปกรอกที่หน้าเว็บแอปพลิเคชันส่งไปแก้ไขโดยอาศัยความรู้จากหลักการออกเสียงพยางค์ในภาษาไทยมาใช้แก้ไขผลลัพธ์การรู้จำอักขระที่ผิด เมื่อแก้ไขแล้วข้อความนั้นจะถูกนำมารวมกันและหาคำที่มีความเหมือนกัน เริ่มที่พยางค์สูงสุดของคำในภาษาไทยคือ 7 พยางค์ไล่ไปจนถึง 1 พยางค์ โดยต้องมีความเหมือนกันอย่างน้อย 66% ของคำที่มี 1 พยางค์ 80% ของคำที่มี 2 - 3 พยางค์ และ 90% ของคำที่มี 4 พยางค์ขึ้นไปจนถึง 7 พยางค์ โดยใช้ไลบรารีภาษาไพธอน คือ difflib และวัดผลการแก้ไขข้อความโดยใช้ Bleu Score แบบ unigram มาวัดผลการแก้ไขจากข้อความตัวอย่างได้คะแนน 0.66 หลังจากเสร็จสิ้นกระบวนการจะนำผลลัพธ์จากการแก้ไขไปแสดงผลบนหน้าเว็บแอปพลิเคชัน

ABSTRACT

This article presents a solution for correcting errors that occur in recognizing handwritten Thai characters. Recognizing handwritten Thai characters is challenging because each person's handwriting is different. Therefore, the results obtained from recognition may contain errors, such as words that cannot be pronounced or incorrect words that need to be corrected. This article applies natural language processing techniques to improve the results obtained from character recognition. The input data is text obtained from recognizing handwritten Thai characters, which is then entered into a web application for correction. The correction process utilizes knowledge of Thai phonetics to rectify the inaccuracies in the character recognition results. After the correction, the texts are combined, and similar words are identified.

*Corresponding Author, E-mail: waijanya_s@silpakorn.edu, promrit_n@silpakorn.edu

Received: date: 6 June 2023 | Revised date: 20 June 2023 | Accepted date: 14 July 2023

The process starts with words with the highest number of syllables in the Thai language, which is seven syllables, and proceeds down to one syllable. The requirement is that the words have a minimum similarity of 66% for one-syllable words, 80% for two to three syllables, and 90% for four to seven syllables. The Python library "difflib" is used for this task, and the effectiveness of the correction is evaluated using the unigram BLEU score. The sample text achieved a score of 0.66. Upon completion of the process, the corrected results are displayed on the web application.

คำสำคัญ: การแก้ไขคำผิด การรู้จำอักขระ การออกเสียงพยางค์ภาษาไทย การวัดความเหมือนกันของสายอักขระด้วยบลูสกอร์ ลายมือเขียนภาษาไทย

Keywords: Fixing Incorrect Words, Optical Character Recognition, Thai Pronunciation of Syllables, Measuring the Similarity of Two Texts Using a BLEU Score, Thai Handwriting

บทนำ

การทำ OCR ซึ่งย่อมาจาก Optical Character Recognition เป็นตัวช่วยอ่านข้อความจากเอกสาร รูปภาพหรือแม้กระทั่งสิ่งพิมพ์อิเล็กทรอนิกส์ แปลงออกมาเป็นข้อความ ตัวอักษร และตัวเลข เทคโนโลยี OCR ช่วยได้มากในการจัดเก็บเอกสาร เพราะสามารถนำข้อมูลที่แปลงออกมาไปใช้งานได้ทันทีโดยไม่ต้องมาพิมพ์ใหม่เพื่อจัดเก็บ ในโลกยุคปัจจุบัน OCR รองรับหลากหลายภาษา ซึ่งรวมถึงภาษาไทยด้วย การทำ OCR ลายมือเขียนภาษาไทยนั้นมีความท้าทายอย่างมาก เนื่องจากลายมือของแต่ละคนนั้นแตกต่างกัน ในยามปกติปราศจากความเร่งรีบการเขียนข้อความอะไรสักอย่างหนึ่ง เราเขียนตัวหนังสือที่มองแล้วอ่านง่ายและทำให้สแกนข้อความแล้วแปลงออกมาเป็นข้อความง่ายขึ้นอีกด้วย แต่เมื่อเกิดเหตุการณ์ที่ต้องเร่งรีบจดเร่งเขียนจนเขียนหวัดหรือเขียนติดกัน การทำ OCR ก็ไม่ได้ทำได้ง่ายตายเข่นนั้นแล้ว เพราะอาจเจอเหตุการณ์ที่เมื่อทำ OCR แล้วได้ข้อความที่แปลงออกมาได้ไม่ตรงกับข้อความที่แท้จริงในเอกสาร

ข้อความที่มีลักษณะเป็นการเขียนด้วยลายมือ ในด้านการคิดวิเคราะห์ของมนุษย์เมื่อมองลายมือแล้วรู้ได้ว่าตัวภาษาของคำหรือประโยคที่มีการเขียนติดกันเป็นคำใดที่อยู่ในประโยคนั้น แต่ในมุมมองของคอมพิวเตอร์เมื่อเรานำเข้ารูปภาพไปสแกนข้อความจากลายมือที่เขียนติดกันหรือการเขียนหวัด ตัวระบบ OCR นั้นเรียนรู้รูปภาพที่เรานำเข้า จากนั้นก็เป็นกระบวนการตัดสินใจเลือกสิ่งที่คุ้นเคยและได้เรียนรู้มาก่อนแล้วแสดงผลลัพธ์ออกมา ซึ่งผลที่ได้อาจไม่ใช่ผลลัพธ์ที่ต้องการ ตัวอย่างปัญหาที่พบจากการทำ OCR จากฟังก์ชันสแกนข้อความของแอปพลิเคชันแชทยอดนิยมแอปพลิเคชันหนึ่ง คือ มีกลุ่มของวลีที่ไม่สามารถอ่านได้ หรือมีสระซ้ำซ้อนกันในคำ ทำให้อ่านออกเสียงไม่ได้ตามหลักของการออกเสียงพยางค์ในภาษาไทย

ตัวช่วยอำนวยความสะดวกในการแก้ไขข้อความในปัจจุบันมีมากมายโดยแต่ละตัวอาจเน้นไปที่ด้านใดด้านหนึ่งโดยเฉพาะเช่น ระบบแก้ไขคำผิดในบันทึกของคลินิก (Aiman *et al.*, 2018) การตรวจจับและแก้ไขการสะกดผิดในภาษาเบงกาลี (Pal, 2021) งานวิจัยด้านการประมวลผลภาษาธรรมชาติโดยคนไทยนั้นมีหลากหลายรูปแบบ เช่น การตรวจจับข่าวปลอมโดยใช้เทคนิคของปัญญาประดิษฐ์ (Meesad *et al.*, 2022) การพยากรณ์ทิศทางราคาหุ้นจากข่าวภาษาไทยโดยใช้วิธีการประมวลผลภาษาธรรมชาติ (Phaphan and Pimpisal, 2020) การสรุปข้อความอัตโนมัติโดยใช้ Keyword-Based Abstractive Method (Ngamcharoen *et al.*, 2022) การแก้ไขข้อความภาษาไทยใช้ไลบรารีจาก pythainlp ในการจัดการตัดคำหรือปรับปรุงข้อความ เช่น การแยกประเภทบทความการวิจัยไทยในฐานข้อมูลจากบทคัดย่อ (Puttipornchai *et al.*, 2022) การแก้ไขคำผิดมีหลากหลายวิธีโดยวิธีหนึ่งที่ได้ผลดีคือการใช้อัลกอริทึม difflib เช่น งานวิจัยการตรวจจับและแก้ไขคำกริยาที่เผยแพร่ในออนไลน์ ผลลัพธ์จากการทดสอบโดยใช้วิธีการ Jaro-winkler Distance และฟังก์ชัน difflib ได้ค่า accuracy 95.9% และ 95% ตามลำดับ การทดสอบกับคำที่มีตัวกำกับ ได้ค่า accuracy 93.29% และ 96% การทดสอบกับคำที่ไม่มีตัวกำกับ ได้ค่า F-score 93.22% และ 96.14% ตามลำดับ (Islam *et al.*, 2019) การสร้าง datasets ในการ

แก้ไขการพิมพ์ผิดภาษาญี่ปุ่นโดยอ้างอิงจากประวัติการแก้ไขของเว็บไซต์วิกิพีเดีย (Tanaka *et al.*, 2020) การประเมินคุณภาพของข้อความใช้ Bleu Score (Papineni *et al.*, 2002) การใช้ OCR เพื่อจัดเก็บข้อมูล เช่น การดึงข้อมูลของผู้ป่วยโดยใช้การ OCR (Chumwatana *et al.*, 2022)

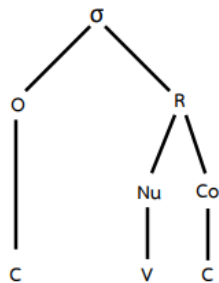
จากปัญหาข้างต้นเมื่อศึกษาวิธีการแก้ไขข้อความจากบทความจึงได้เกิดงานการประมวลผลภาษาธรรมชาติ เพื่อปรับปรุงคำผิดที่เกิดจากวิธีการรู้จำอักขระ ซึ่งช่วยแก้ไขข้อผิดพลาดการแปลงข้อความที่เขียนติดกันหรือเขียนหวัดแล้วแสดงผลเป็นตัวคำที่อ่านไม่ออก แล้วทำให้ตัวข้อความที่แสดงออกมามีความถูกต้องมากขึ้น

วิธีการดำเนินการวิจัย

การดำเนินงานวิจัยเริ่มจากการศึกษารูปแบบการเกิดพยางค์และนำมาวิเคราะห์ลักษณะคำผิดที่เกิดขึ้นจากวิธีการรู้จำอักขระ จากนั้นจึงสร้างระบบเพื่อแก้ไขคำผิดเหล่านั้น ซึ่งจะอธิบายรายละเอียด ในลำดับถัดไป

1. รูปแบบของการเกิดพยางค์

พยางค์ คือ เสียงที่เปล่งออกมา ซึ่งมีความหมายหรือไม่มีความหมายก็ได้ โดยการเปล่งเสียง ครั่ง 1 หมายถึง การเปล่งเสียงพยางค์ โดยมีองค์ประกอบที่เขียนในรูปแผนภูมิตามหลักสัทวิทยา ดัง รูปที่ 1



รูปที่ 1 แผนภูมิองค์ประกอบของพยางค์

โดย **σ** คือ พยางค์

O คือ พยัญชนะต้น ประกอบด้วยพยัญชนะ 44 รูป 21 เสียง โดยสามารถแตกกิ่งได้เป็นพยัญชนะควบ (Consonant Clusters)

R คือ ส่วนท้ายพยางค์ สามารถแยกได้เป็นสองส่วน ได้แก่ Nu คือ ส่วนของแกนพยางค์ คือ สระ(V) สามารถแตกกิ่งเป็นสระเสียงยาวหรือสระประสมได้ และ

Co เป็นส่วนท้ายของพยางค์ ได้แก่ ตัวสะกด (C)

โดย Co สามารถแตกกิ่งออกได้เป็นพยัญชนะควบท้าย

จากรูปแบบของพยางค์ จะได้โครงสร้างของพยางค์เป็น 4 แบบ ได้แก่ [CV] [CCV] [CVC] [CCVC] เมื่อนำพยางค์ภาษาไทยไปเทียบกับรูปแบบข้างต้น จะพบว่าการเขียนและการออกเสียงพยางค์มีการเรียงลำดับตัวอักษรที่ทั้งเหมือนและไม่เหมือนกัน เพราะความพิเศษของแกนพยางค์หรือสระของภาษาไทยสามารถเขียนวางอยู่ในตำแหน่งใดๆ ของพยางค์ก็ได้ โดยมีได้ถึง 9 แบบ ได้แก่ สระหน้า (Vf) ประกอบไปด้วย -เ- -แ- -โ- -ใ- และ -เ- สระหลัง (Vb) ประกอบไปด้วย -ะ -า -อ ฤ และ -ว สระบน (Vt) ประกอบไปด้วย -ิ -ี -ึ -ื และ -ึ สระล่าง (Vu) ประกอบไปด้วย -ุ และ -ู สระที่อยู่ได้ตามลำพัง คือ ฤ ฦ ตำแหน่งของรูปสระข้างต้นสามารถนำมารวมกันเพื่อออกเสียงได้อีกแบบหนึ่ง ได้แก่ สระหน้าและสระหลัง เช่น -เ-า และ -เ-ะ -เ-าะ -เ-อะ และ -เ-อ สระบนและหลัง เช่น -ัวะ -ัว -ำ สระหน้า บนและหลัง เช่น -เียะ -เีย -เือะ -เือ และสุดท้ายคือ สระล่างและบน เช่น -เี -เือ อีกทั้งสระในภาษาไทยบางตัวจะมีการเปลี่ยนรูปและลดรูปเมื่อมีตัวสะกด การเปลี่ยนรูปได้แก่ สระอะเมื่อมีพยัญชนะจะเปลี่ยนตัวเองให้เป็นไม้หันอากาศ เช่น ขะ เดิมตัวสะกด น เขียนว่า ขัน สระอะเมื่อมีตัวสะกดจะเปลี่ยนตัวเองเป็น

สระเอ ไม่ไต่คู่ เช่น เปะ เต็มตัวสะกด ด เขียนว่าเปิด สระแอะเมื่อมีตัวสะกดจะเปลี่ยนรูปเป็นสระแอะ ไม่ไต่คู่ เช่น แอะ เต็มตัวสะกด ง เขียนว่า แข็ง สระเออเมื่อมีตัวสะกดจะเปลี่ยนรูปเป็นสระเอ สระอิ เช่น เขอ เต็มตัวสะกด น เขียนว่า เขิน สระที่มี การลดรูปนั้น คือ สระโอะเมื่อมีตัวสะกดจะลดรูปเหลือแค่พยัญชนะต้นตามด้วยสะกด เช่น โกะ เต็มตัวสะกด ด เขียนว่า กด สระเออจะมีการลดรูปเมื่อมีตัวสะกดโดยจะลดรูปเหลือเพียงสระเอตามด้วยตัวสะกด เช่น เคอ เต็มตัวสะกด ย อ่านว่า เคย สระอัวเมื่อมีตัวสะกดจะลดรูปเหลือเพียง ว เช่น จัว เต็มตัวสะกด ง เขียนว่า วงง นอกเหนือจากสระแล้วอีกหนึ่งสิ่งที่การเขียน และอ่านพยางค์ภาษาไทยนั้นพิเศษคือการมีคือวรรณยุกต์ใช้กำกับกับการออกเสียง โดยวรรณยุกต์สามารถปรากฏได้อยู่บน พยัญชนะเมื่อสระนั้นไม่ใช่สระบน หากมีสระบน วรรณยุกต์ก็จะปรากฏอยู่บนสระบน ส่วนพยัญชนะภาษาไทยนั้นมี 44 รูป 21 เสียง ทุกตัวสามารถทำหน้าที่เป็นพยัญชนะต้นได้ แต่ไม่ใช่พยัญชนะทุกตัวที่สามารถทำหน้าที่เป็นตัวสะกดได้ คือ ย ฮ ฬ ฦ และฦ ไม่สามารถออกเสียงเป็นตัวสะกดได้ เมื่อนำตัวอย่างการเขียนพยางค์แบ่งตามตำแหน่งของรูปสระและเทียบกับแผนภูมิ องค์ประกอบของพยางค์โดยใช้ T แทนตำแหน่งของวรรณยุกต์ ได้ดังตารางที่ 1

ตารางที่ 1 ตัวอย่างการเขียนพยางค์แบ่งตามตำแหน่งของรูปสระและเทียบกับแผนภูมิองค์ประกอบของพยางค์โดยใช้ T แทนตำแหน่งของวรรณยุกต์

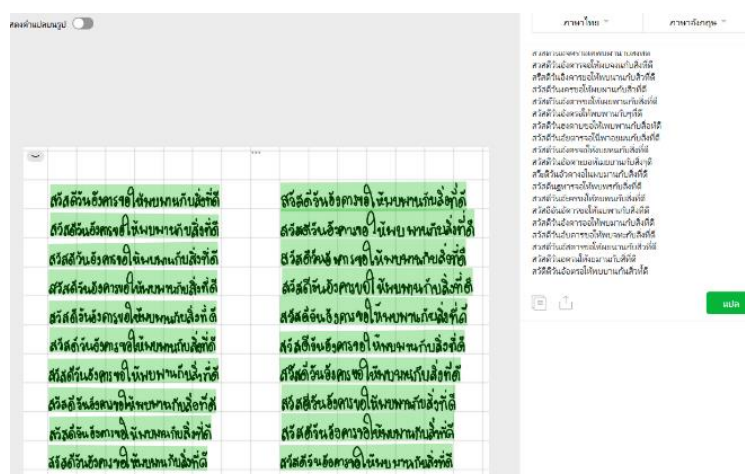
ตำแหน่งของ แกนพยางค์	ตัวอย่างการเขียนพยางค์	รูปแบบการเขียนของพยางค์	การออกเสียง	รูปแบบการออกเสียง ของพยางค์
หน้า	ไป	VfC	ป-ไ-ไป	CV
	ไป	VfCT	ป-ไ-ไป-ไม้เอก-ไป	CVT
	ไหล	VfCC	ห-น-ไ-ไหล	CCV
	ไหร	VfCCT	ห-ร-ไ-ไหร-ไม้เอก-ไหร	CCVT
	โรง	VfCC	ร-ไ-ง-โรง	CVC
	โล่ง	VfCTC	ล-ไ-ง-โล่ง-ไม้เอก-โล่ง	CVCT
	โครง	VfCCC	ค-ร-ไ-ง-โครง	CCVC
	โคร่ง	VfCCTC	ค-ร-ไ-ง-โครง-ไม้เอก-โคร่ง	CCVCT
หลัง	นา	CVb	น-อา-นา	CV
	นำ	CTVb	น-อา-นำ-ไม้เอก-นำ	CVT
	ปลา	CCVb	ป-ล-อา-ปลา	CCV
	คร่า	CCTVb	ค-ร-อา-คร่า-ไม้เอก-คร่า	CCVT
	บาง	CVbC	บ-อา-ง-บาง	CVC
	บ้าง	CVbTC	บ-อา-ง-บาง-ไม้โท-บ้าง	CVC
	คลอง	CCVbC	ค-ล-อ-ง-คลอง	CCVC
	คล่อง	CCVbTC	ค-ล-อ-ง-คลอง-ไม้โท-คล่อง	CCVCT
บน	คือ	CVt	ค-อิ-คือ	CV
	ชื่อ	CVtT	ช-อิ-ชื่อ-ไม้เอก-ขึ้น	CVT
	หรือ	CCVt	ห-ร-อิ-หรือ	CCV
	พริ้ว	CCVtT	พ-ร-อิ-พริ้ว-ไม้เอก-พริ้ว	CCVT
	กิน	CVtC	ก-อิ-น-กิน	CVC
	ถิ่น	CVtTC	ถ-อิ-น-ถิ่น-ไม้เอก-ถิ่น	CVCT
	กลืน	CCVtC	ก-ล-อิ-น-กลืน	CCVC
	คลืน	CCVtTC	ค-ล-อิ-น-คลืน-ไม้เอก-คลืน	CCVCT

ตารางที่ 1 ตัวอย่างการเขียนพยางค์แบ่งตามตำแหน่งของรูปสระและเทียบกับแผนภูมิองค์ประกอบของพยางค์โดยใช้ T แทนตำแหน่งของวรรณยุกต์ (ต่อ)

ตำแหน่งของ แกนพยางค์	ตัวอย่างการเขียนพยางค์	รูปแบบการเขียนของพยางค์	การออกเสียง	รูปแบบการออกเสียง ของพยางค์
ล่าง	ดู	CVu	ด-อู-ดู	CV
	คู่	CVuT	ด-อู-คู-ไม่เอก-คู่	CVT
	กลู	CCVu	กล-อู-กลู	CCV
	ครู่	CCVuT	คร-อู-ครู่-ไม่เอก-ครู่	CCVT
	ขุน	CVuC	ข-อู-น-ขุน	CVC
	ขุน	CVuTC	ข-อู-น-ขุน-ไม่เอก-ขุน	CVCT
	กรุ่น	CCVuTC	กร-อู-น-กรุ่น-ไม่เอก-กรุ่น	CCVCT
หน้าและหลัง	และ	VfCVb	ล-แอะ-และ	CV
	โป๊ะ	VfCTVb	ป-โอะ-โปะ-ไม้ตรี-โป๊ะ	CVT
	แกละ	VfCCVb	กล-แอะ-แกละ	CCV
	เล่น	VfCTC	ล-เอะ-น-เลน-ไม้เอก-เล่น	CVCT
	เสร็จ	VfCCVtC	สร-เอะ-จ-เสร็จ	CCVC
บนและหลัง	กำ	CVb	ก-อำ-กำ	CV
	ก้ำ	CTVb	ก-อำ-ก้ำ-ไม้เอก+ก้ำ	CVT
	พรำ	CCVb	พร-อำ-พรำ	CCV
	พรำ	CCTVb	พร-อำ-พรำ-ไม้เอก-พรำ	CCVT
	พวง	CVbC	พ-อัว-ง-พวง	CVC
	พ่วง	CTVbC	พ-อัว-ง-พวง-ไม้เอก-พ่วง	CVCT
	กลวง	CCVbC	กล-อัว-ง-กลวง	CCVC
	กล้วย	CCVbTC	กล-อัว-ย-กล้วย-ไม้โท-กล้วย	CCVCT
ล่างและบน	เขิน	VfCVtC	ข-เออ-น-เขิน	CVC
	เป็น	VfCVtTC	ป-เออ-น-เป็น-ไม้เอก-เป็น	CVCT
	เกริม	VfCCVtC	กร-เออ-ม-เกริม	CCVC
	เกริ่น	VfCCVtTC	กร-เออ-น-เกริ่น-ไม้เอก-เกริ่น	CCVCT
หน้า บนและ หลัง	เรือ	VfCVtVb	ร-เอือ-เรือ	CV
	เรือ	VfCVtVb	ร-เอือ-เรือ-ไม้เอก-เรือ	CVT
	พลี	VfCCVtVb	พล-เอือ-พลี	CCV
	เหยื่อ	VfCVtTVb	หย-เอือ-เหยื่อ-ไม้เอก-เหยื่อ	CCVT
	เรือ	VfCVtVbC	ร-เอือ-เรือ	CVC
	เรือ	VfCVtTVbC	ร-เอือ-ง-เรือ-ไม้เอก-เรือ	CVCT
	เคลือบ	VfCCVbC	คล-เอือ-บ-เคลือบ	CCVC
	เกลี้ยง	VfCCVtTVbC	กล-เอือ-ง-เกลี้ยง-ไม้โท-เกลี้ยง	CCVCT

2. การวิเคราะห์คำผิดที่เกิดจากการรู้จำอักขระ

การวิเคราะห์ตัวอย่างคำผิดจากผลลัพธ์ที่ได้จากการรู้จำอักขระ ตัวอย่างประโยคลายมือเขียน “สวัสดีวันอังคารขอให้พบพานกับสิ่งที่ดี” โดยใช้แอปพลิเคชันส่งข้อความแอปพลิเคชันหนึ่งแสดงดังรูปที่ 2 และผลลัพธ์ที่ได้จากการรู้จำอักขระแสดงดังรูปที่ 3



รูปที่ 2 การรู้จำอักษรโดยใช้แอปพลิเคชันส่งข้อความ

สวัสดิ์วันอังคารขอให้พบจนกับสิ่งที่ดี
 สวัสดิ์วันอังคารขอให้พบนานกับสิ่งที่ดี
 สวัสดิ์วันครขอให้พบพานกับสิ่งที่ดี
 สวัสดิ์วันอังคารขอให้เผยพานกับสิ่งที่ดี
 สวัสดิ์วันอังคารขอให้พบพานกับๆที่ดี
 สวัสดิ์วันองคาบขอให้เพพบพานกับสิ่งที่ดี
 สวัสดิ์วันอัยตารขอให้เพาอยมนกับสิ่งที่ดี
 สวัสดิ์วันอังคารขอให้งบยหนกับสิ่งที่ดี
 สวัสดิ์วันอ้อคาขอให้เมยยานกับสิ่งที่ดี
 สวัสดิ์วันอัวคางขอให้พบพานกับสิ่งที่ดี
 สวัสดิ์วันธูทารขอให้พบพรกับสิ่งที่ดี
 สวัสดิ์วันอะครขอให้คบพนกับสิ่งที่ดี
 สวัสดิ์วันอ้อครขอให้นบพาเก็บกับสิ่งที่ดี
 สวัสดิ์วันอังคารขอให้พบมานกับสิ่งที่ดี
 สวัสดิ์วันอับครขอให้พบจจะกับสิ่งที่ดี
 สวัสดิ์วันอัสครขอให้ผยนานกับสิ่งที่ดี
 สวัสดิ์วันอครนให้่งยมากับสิ่งที่ดี
 สวัสดิ์วันอ้อตรขอให้พบพานกับสิ่งที่ดี

รูปที่ 3 ผลลัพธ์การเรียนรู้จำอักขระ

จากผลลัพธ์การจำอักขระประโยคสวัสดีวันอังคารขอให้พบพานกับสิ่งที่ดีในรูปที่ 3 สามารถแยกผลลัพธ์ค่าที่คิดได้
ดังนี้

สวัสดี	ได้ผลลัพธ์ที่ผิดเป็น สวสดี สรียดี สวส์ดี สะวะดี สวสดี สวดีดี
วัน	ได้ผลลัพธ์ที่ผิดเป็น อัน นฐ
อังคาร	ได้ผลลัพธ์ที่ผิดเป็น อิงคาร งคร อังตาร อัยตาร ฮงคาบ อังคร อ้อคาะ ทาร อัคร อ๊ะคร อัวคาร อ้อต อัสคาร อครน อับคาร
ขอ	ได้ผลลัพธ์ที่ผิดเป็น จอ งอ ยอ ขง รอ
ให้	ได้ผลลัพธ์ที่ผิดเป็น ให้ ไ้ ห้ ใน
พบ	ได้ผลลัพธ์ที่ผิดเป็น ผบ เผย เพบ พา งบ งย นบ คบ ผย เมย
พาน	ได้ผลลัพธ์ที่ผิดเป็น จงน นาน พาน ยหน อยมน มาน ยาน จทะ พา

กับ	ได้ผลลัพธ์ที่ผิดเป็น กัน
สิ่ง	ได้ผลลัพธ์ที่ผิดเป็น สิว ลือ ลี
ที่	ได้ผลลัพธ์ที่ผิดเป็น หั หั ๆ
ดี	ไม่มีผลลัพธ์ที่ผิด

สรุปประเภทคำผิดที่ได้จากการรู้จำอักขระ

1. คำผิดที่อ่านไม่ได้ตามรูปแบบของพยางค์
2. คำผิดที่ต้องมีการปรับให้ถูกต้อง

จากรูปแบบของการเกิดพยางค์ สามารถวิเคราะห์คำผิดที่เกิดจากการรู้จำอักขระแล้วไม่สามารถอ่านได้ดังนี้

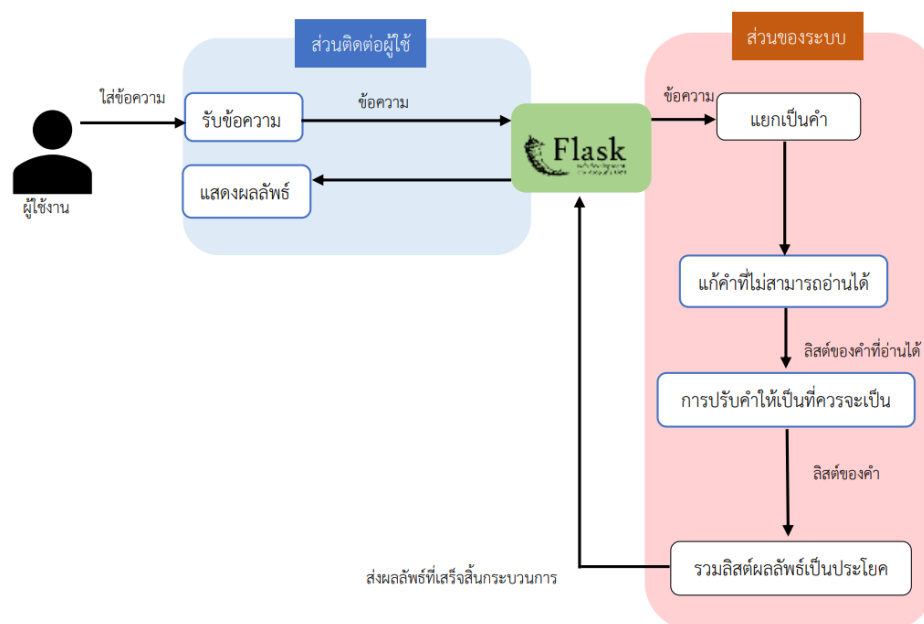
1. ผลลัพธ์การรู้จำอักขระ ปรากฏพยัญชนะไม่สามารถออกเสียงเป็นส่วนท้ายของพยางค์หรือตัวสะกดได้ เช่น อ้อ

2. มีสระบน ปรากฏขึ้นตามหลังพยัญชนะต้นของพยางค์ที่มีสระหน้าขึ้นต้น และตัวสระหน้านั้นไม่สามารถรวมสระบนนั้น เพื่ออ่านออกเสียงแบบสระหน้า บนและหลังได้ หรือมีสระปรากฏขึ้นมาติดกัน โดยไม่ตรงกับรูปแบบตำแหน่งของสระที่อ่านออกเสียงได้ เกิดขึ้นกับผลการรู้จำอักขระของพยางค์ที่เป็นสามตัวอักษร เช่น ใริ ะ

3. สระบางตัวอยู่ในสถานะเปลี่ยนรูปหรือลดรูปที่ต้องมีส่วนท้ายของพยางค์แต่ผลการรู้จำอักขระที่ออกมาไม่มีส่วนท้ายของพยางค์ เช่น สวัดดี

3. ภาพรวมของระบบ

ระบบปรับปรุงคำผิดจะแบ่งออกเป็น 2 ส่วน คือ ส่วนของระบบและส่วนติดต่อผู้ใช้ ดังรูปที่ 4 โดยจะอธิบายรายละเอียดของการทำงานแต่ละส่วน ในลำดับถัดไป



รูปที่ 4 การทำงานโดยรวมของระบบปรับปรุงคำผิด

ส่วนของระบบ มีกระบวนการทำงานดังนี้

1. การทำงานของการแยกข้อความเพื่อเข้ากระบวนการแก้คำให้อ่านได้ในรูปแบบของลิสต์

หลังจากรับข้อความที่ได้จากการรู้จำอักขระของผู้ใช้งานที่ส่งผ่าน API มาแล้ว ใช้ word_tokenize เพื่อตัดคำให้อยู่ในรูปแบบของลิสต์

2. การทำงานในส่วนของการแก้คำที่ไม่สามารถอ่านได้

จะนำลิสต์ของคำมาตรวจสอบทีละ 1 คำว่าคำนั้นมีอยู่ในพจนานุกรมหรือไม่ หากมีจะร่อนำไปทำในกระบวนการถัดไป หากไม่มีแสดงว่าอาจมีความผิดปกติของคำ แก้ไขโดยนำคำมาตัดเป็นพยางค์ และแยกพยางค์ 1 พยางค์ออกเป็นลิสต์ของตัวอักษรจากนั้นจะนำลิสต์ของตัวอักษรไปหาจำนวนอักขระและเทียบรูปแบบของพยางค์เพื่อแก้ไขส่วนที่ทำให้พยางค์นั้นอ่านไม่ได้ โดยรูปแบบพยางค์ได้มาจากการเทียบการออกเสียงกับการเขียนรูปแบบพยางค์ภาษาไทย ซึ่งได้มารูปแบบหลักจำนวน 16 รูปแบบ ให้ C แทนตำแหน่งของพยัญชนะต้น c แทนตำแหน่งของอักษรควบหรืออักษรนำ V แทนตำแหน่งของสระ Co แทนตำแหน่งของตัวสะกด มาใช้อธิบายรูปแบบหลักของพยางค์ เริ่มจากรูปแบบที่ขึ้นต้นด้วยพยัญชนะมี 4 รูปแบบ คือ CV CcV CVCo CcVCo และรูปแบบขึ้นต้นด้วยสระมี 12 รูปแบบ แบ่งได้ 4 ประเภท ได้แก่ รูปแบบสระหน้าตัวเดียว 4 รูปแบบ คือ VC VCc VCCo VCcCo รูปแบบที่เป็นสระหน้าและหลัง 2 รูปแบบ คือ VCV VCcV รูปแบบที่เป็นสระหน้าบน และหลัง 4 รูปแบบ คือ VCVV VCVCVCo VCcVV VCcVCVCo สุดท้ายคือ รูปแบบที่เป็นสระหน้าและบน 2 รูปแบบ ได้แก่ VCVCVCo VCcVCVCo เมื่อนำรูปแบบ 16 รูปแบบไปเทียบพยางค์ 1 พยางค์ที่พบบ่อย สามารถมีได้ 56 รูปแบบ โดยมีขนาดความยาวตั้งแต่ 2 - 7 ตัวอักษร และให้ T แทนตำแหน่งของวรรณยุกต์ แสดงรูปแบบทั้ง 56 รูปแบบแยกกันตามความยาวของตัวอักษรดังตารางที่ 2 ถึงตารางที่ 7 ดังนี้

ตารางที่ 2 รูปแบบของพยางค์ที่มี 2 ตัวอักษร 3 รูปแบบ

ลำดับ	รูปแบบการเขียน	ตัวอย่างพยางค์	ลำดับการเขียน	ลำดับการอ่าน	รูปแบบการอ่าน
1	CCo	กต	ก-ต	ก-โ-ด-กต	CVC
2	CV	กา	ก-า	ก-า-กา	CV
3	VC	ไป	ไ-ป	ป-ไ-ไป	CV

ตารางที่ 3 รูปแบบของพยางค์ที่มี 3 ตัวอักษร 10 รูปแบบ

ลำดับ	รูปแบบการเขียน	ตัวอย่างพยางค์	ลำดับการเขียน	ลำดับการอ่าน	รูปแบบการอ่าน
1	CVCo	กัต	ก-ัต	ก-ัต-กัต	CVC
2	CVT	ปู่	ป-ู่	ป-ู่-ปู่	CV
3	CTV	ป่า	ป-่า	ป-่า-ป่า	CV
4	CTCo	ป่น	ป-่น	ป-โ-น-ป่น	CVC
5	CcV	ปลา	ป-ล-า	ป-ล-า-ปลา	CCV
6	CcCo	ปลด	ป-ล-ด	ป-ล-โ-ด-ปลด	CCVC
7	VCCo	เหว	เ-ห-ว	ห-เว-เหว	CVC
8	VCc	เปล	เ-ป-ล	ป-ล-เ-เปล	CCV
9	VCV	เบา	เ-บ-า	บ-เ-า-เบา	CV
10	VCT	เป้	เ-ป-๊	ป-เ-เป-๊-เป้	CV

ตารางที่ 4 รูปแบบของพยางค์ที่มี 4 ตัวอักษร 17 รูปแบบ

ลำดับ	รูปแบบการเขียน	ตัวอย่างพยางค์	ลำดับการเขียน	ลำดับการอ่าน	รูปแบบการอ่าน
1	CTVCo	บ้าน	บ-้า-น	บ-า-น-บ้าน-บ้าน	CVC
2	CVTCO	ปิ่น	ป-ิ่น	ป-ิน-ปิ่น-ปิ่น	CVC
3	CCo ตัวการันต์	มนต์	ม-น-ตฺ์	ม-เ-น-มน	CVC
4	CV ตัวการันต์	บาร์	บ-า-รฺ์	บ-า-บา	CV
5	CcTV	คร่า	ค-ร-่า	ค-ร-า-ครา-คร่า	CCV
6	CcVT	หมู่	ห-ม-ู	ห-ม-ู-หมู่-หมู่	CCV
7	CcTCO	หล่น	ห-ล-่น	ห-ล-เ-น-หล่น-หล่น	CCVC
8	CcVCo	ครีบ	ค-ร-ี-บ	ค-ร-ี-บ-ครีบ	CCVC
9	CVC	ลือก	ล-ี-อ-ก	ล-เ-า-ลือก	CVC
10	VC ตัวการันต์	เมล์	เ-ม-ลฺ์	ม-เ-เม	CV
11	VCcT	ไหล่	ไ-ห-ล-่	ห-ล-ไ-ไหล-ไหล่	CCV
12	VCcV	เพลา	เ-พ-ล-า	พ-ล-เ-า-เพลา	CCV
13	VCVW	เรือ	เ-ร-เ-อ	ร-เ-อ-เรือ	CV
14	VCVCo	เทอม	เ-ท-อ-ม	ท-เ-อ-ม-เทอม	CVC
15	VCTV	เป่า	เป-่า	ป-เ-า-เปา-เป่า	CV
16	VCTCo	เม่น	เ-ม-่น	ม-เ-น-เม่น-เม่น	CVC
17	VCcCo	เพลง	เ-พ-ล-ง	พ-ล-เ-ง-เพลง	CCVC

ตารางที่ 5 รูปแบบของพยางค์ที่มี 5 ตัวอักษรจำนวน 17 รูปแบบ

ลำดับ	รูปแบบการเขียน	ตัวอย่างพยางค์	ลำดับการเขียน	ลำดับการอ่าน	รูปแบบการอ่าน
1	CVC ตัวการันต์	ราวน์	ร-า-ว-น์	ร-า-ว-ราว	CVC
2	CV ตัวการันต์C	การ์ด	ก-า-รฺ์-ด	ก-า-ด-กา-ด	CVC
3	CVT ตัวการันต์	รีส์	ร-ี-สฺ์	ร-ี-ร-ี-รี	CV
4	CTV ตัวการันต์	ลาร์	ล-่า-ร-์	ล-า-ลา-ล่า	CV
5	CcVTCO	กริ่ง	ก-ร-ิ่ง	ก-ร-ิ่ง-กริ่ง-กริ่ง	CCVC
6	CcTVCo	กว้าง	ก-ว-่า-ง	ก-ว-า-ง-กวา-ง-กว้าง	CCVC
7	CcVCo	บลิ๊ก	บ-ล-ี-อ-ก	บ-ล-เ-า-บ-บลิ๊ก	CCVC
8	CCo ตัวการันต์	ยนตร์	ย-น-ตฺ์-ร	ย-เ-น-ย-น	CVC
9	VCT ตัวการันต์	เลห์	เ-ล-หฺ์	ล-เ-เล-เลห์	CV
10	VCCo ตัวการันต์	เพนต์	เ-ท-น-ตฺ์	เ-ท-น-เพน	CVC
11	VCc ตัวการันต์	เพลย์	เ-พ-ล-ย-์	พ-ล-เ-เพล	CCV
12	VCcVW	เกลื้อ	เ-ก-ล-เ-อ	ก-ล-เ-อ-เกลื้อ	CCV
13	VCcVCo	แบล็ค	แ-บ-ล-เ-ค	บ-ล-เ-แ-ค-แบล็ค	CCVC
14	VCcTV	เหล้า	เ-ห-ล-่า	ห-ล-เ-า-เหลา-เหล้า	CCV
15	VCcTCO	เหม่น	เ-ห-ม-่น	ห-ม-เ-น-เหม่น-เหม่น	CCVC
16	VCVTV	เชื้อ	เ-ช-เ-อ	ช-เ-อ-เชื้อ-เชื้อ	CV
17	VCVCo	เรียน	เ-ร-เ-ย-น	ร-เ-ย-น-เรียน	CVC

ตารางที่ 6 รูปแบบของพยางค์ที่มี 6 ตัวอักษรจำนวน 7 รูปแบบ

ลำดับ	รูปแบบการเขียน	ตัวอย่างพยางค์	ลำดับการเขียน	ลำดับการอ่าน	รูปแบบการอ่าน
1	CcVCo ตัวการันต์	กรานต์	ก-ร-า-น-ต-	ก-ร-า-น-กราน	CCVC
2	CVCo ตัวการันต์ (สระอู/สระอุ-ทัณฑฆาต)	พันธุ์	พ-น-ร-ู-	พ-ะ-น-พัน	CVC
3	CcV การันต์ (สระอู/สระอุ-ทัณฑฆาต)	ปรีดี	ป-ร-ี-ดิ-	ป-ร-ี-ปรี	CCV
4	CVCo การันต์ (พยัญชนะ-พยัญชนะ-ทัณฑฆาต)	จันทร์	จ-น-ท-ร-	จ-ะ-น-จัน	CVC
5	VCcCo ตัวการันต์	แบรนต์	แ-บ-ร-น-ต-	บ-ร-แ-น-แบรน	CCVC
6	VCcVTV	เพรื่อ	เพ-ร-ื่อ-อ	พ-ร-ื่อ-เพรื่อ-เพรื่อ	CCV
7	VCcWCo	เครียต	เค-ร-ีย-ต	ค-ร-เีย-ต-เครียต	CCVC

ตารางที่ 7 รูปแบบของพยางค์ที่มี 7 ตัวอักษรจำนวน 2 รูปแบบ

ลำดับ	รูปแบบการเขียน	ตัวอย่างพยางค์	ลำดับการเขียน	ลำดับการอ่าน	รูปแบบการอ่าน
1	VCcVตัวการันต์	เคลียร์	เค-ล-ี่-ย-ร-	ค-ล-เีย-เคลีย	CCV
2	VCcVTVCo	เคลื่อน	เค-ล-ี่-อ-น	ค-ร-เือ-น-เคลื่อน-เคลื่อน	CCVCT

นำรูปแบบที่แสดงในตารางที่ 2 ถึงตารางที่ 7 ไปเขียนโปรแกรมและตรวจสอบและแก้ไข โดยสิ่งที่ตรวจสอบอย่างแรกคือ หากขึ้นต้นด้วยสระมีตัวอักษรไม่เกิน 3 ตัวอักษรแล้วมีสระอื่นที่รวมกันแล้วไม่สามารถอ่านออกได้ปรากฏขึ้น แก้ไขโดยการไม่เก็บตัวอักษรที่เป็นสระที่เกินมาตัวนั้น จัดเก็บเพียงสิ่งที่เก็บครบตามรูปแบบพยางค์ไว้ในลิสต์เพื่อไปทำในกระบวนการถัดไป การตรวจสอบที่สอง ตรวจสอบที่ตำแหน่งของตัวสะกดว่าตัวอักษรตัวนั้นสามารถทำหน้าที่เป็นตัวสะกดได้หรือไม่ หากไม่ก็จะถูกแก้ไขโดยนำตัวอักษรที่มีรูปคล้ายกันมาแทนที่ สิ่งที่ตรวจสอบและแก้ไขอีกอย่างคือหากสระนั้นอยู่ในรูปของการเปลี่ยนรูปหรือลดรูปแล้วไม่ปรากฏตัวสะกดในผลลัพธ์การรู้จำอักขระจะสุมตัวสะกดมาเพิ่มให้รูปแบบพยางค์ครบสมบูรณ์ โดยสรุปการแก้ไขทั้งหมดตามกฎได้ดังนี้

กฎการแก้ไขที่ได้จากหลักการเกิดพยางค์

- เมื่อเจอสระเกินจากรูปสระที่มีไม่เกิน 3 ตัวอักษรจะไม่เก็บสระตัวนั้นเพื่อรวมกลับเป็นพยางค์
- ที่ตำแหน่งตัวสะกด ตัวอักษรที่ไม่สามารถเป็นตัวสะกดได้จะดำเนินการดังนี้
 - ผ จะเปลี่ยนเป็นตัวอักษรที่มีรูปคล้ายกันคือ พ
 - ฝ จะเปลี่ยนเป็นตัวอักษรที่มีรูปคล้ายกันคือ ฟ
 - ณ จะเปลี่ยนเป็นตัวอักษรที่มีรูปคล้ายกันคือ น
 - ณ จะเปลี่ยนเป็นตัวอักษรที่มีรูปคล้ายกันคือ ณ
 - ห จะเปลี่ยนเป็นตัวอักษรที่มีรูปคล้ายกันคือ ท
 - ฮ จะเปลี่ยนเป็นตัวอักษรที่มีรูปคล้ายกันคือ ส
- เมื่อตัวอักษรที่ได้จากการแยกพยางค์ไปตรงกับรูปแบบที่เป็นสระที่ต้องมีตัวสะกดแต่ไม่มีตัวสะกดในพยางค์นั้น จะสุมตัวสะกดมาใส่

เมื่อตรวจสอบเสร็จแล้วนำอักขระกลับไปรวมให้เป็นพยางค์ จากนั้นนำพยางค์ที่ผ่านการแก้ไขไว้ในลิสต์ที่รอนำไปทำในส่วนการแก้คำที่อ่านได้ให้เป็นคำที่ควรจะเป็น โดยเขียนกระบวนการทั้งหมดเป็นชุดโค้ดได้ดังรูปที่ 5

```

1. Pseudocode Can not read by Thai syllable ref. edited
2. Input = list of word
3. Output = list of Syllable
4. editSyllable = [], resultEdit = [], character = []
5. for first word to last word do
6. if word not in Thai dictionary then
7.   subword_tokenize(word)
8. for first syllable to last syllable do
9. for first character to last character in syllable do
10. character collect character
11. end for
12. If When encountering a vowel that exceeds the syllable pattern have 3 letters then
13. Not collect vowel character cannot be in that form and combine back to editSyllable
14. if syllable patterns have vowel complete form and need spelling but didn't have spelling then
15. random spelling for fix character pattern and combine back to editSyllable
16. if syllable patterns have character in Spelling position can't be Spelling then
17. if character in Spelling position is ฉ or ณ or พ or ฝ or ฮ or ห then
18. Replace ฉ to น and combine back to editSyllable
19. Replace ณ to ณ and combine back to editSyllable
20. Replace พ to พ and combine back to editSyllable
21. Replace ฝ to ฝ and combine back to editSyllable
22. Replace ฮ to ฮ and combine back to editSyllable
23. Replace ห to ห and combine back to editSyllable
24. end if
25. if character all is in syllable patterns
26. combine back to editSyllable
27. end if
28. resultEdit collect editSyllable
29. clear editSyllable, character
30. end for
31. else resultEdit collect word
32. end for

```

รูปที่ 5 ชุดโค้ดของการแก้คำที่ไม่สามารถอ่านได้

3. การทำงานส่วนการปรับคำให้เป็นคำที่ควรจะเป็น

1. **Pseudocode Adjust the words that should be**
2. **Input** list result from Can not read by Thai syllable ref. edited process
3. **Output** String sentence
4. position = 0, allword= length of Syllable list, allofword = length of Syllable list, newword = [],
5. sentence = '', n = number between 1 to 6
6. **while** allofword > 0 **do**
7. **if** number of syllable >= 7 **then**
8. **if** diffilib return value from highest summary of word from 7 to 1 **then**
9. Update position of Syllable , newword add value , allofword decrease
10. **If** position >= allword-1 **then**
11. break
12. **end if**
13. **end if**
14. **If** number of syllable == n **then**
15. **If** diffilib return value from summary n words to 1 word **then**
16. Update position of Syllable , newword add value, allofword decrease
17. **If** position >= allword-1 **then**
18. break
19. **end if**
20. **end if**
21. **for** i = 0 to length(newword) **do**
22. sentence = newword[0] +...+ newword[last]
23. **end loop**
24. **return** sentence

รูปที่ 6 ชุดโค้ดของส่วนการปรับคำให้เป็นคำที่ควรจะเป็น

เมื่อได้ลิสต์จากกระบวนการแก้คำที่ไม่สามารถอ่านได้ จะนำพยางค์และคำในลิสต์ไปรวมกันและใช้ diffilib มาหาคำที่มีความเหมือนกัน โดยรวมสูงสุดที่ 7 พยางค์ไปจนถึง 1 พยางค์ อ้างอิงคำที่นำมาหาความเหมือนกันมาจากพจนานุกรมที่ได้มาจากคลังคำที่มีใน pythainlp ซึ่งตั้งเงื่อนไขให้ diffilib คำนวณค่าที่ใช้ปรับปรุงก็ต่อเมื่อ หากรวม 1 พยางค์แล้วมีผลลัพธ์ต้องมีความเหมือนกัน 66% รวม 2 - 3 พยางค์แล้วมีผลลัพธ์ต้องมีความเหมือนกัน 80% รวม 4 - 7 พยางค์แล้วมีผลลัพธ์ต้องมีความเหมือนกัน 90% เมื่อเสร็จสิ้นกระบวนการแล้วจะรวมผลลัพธ์ทั้งหมดในลิสต์ให้เป็นประโยค เขียนกระบวนการทั้งหมดเป็นชุดโค้ดได้ดังรูปที่ 6

ส่วนติดต่อผู้ใช้

หน้าเว็บแอปพลิเคชันประกอบไปด้วย ช่องกรอกข้อความ ให้ผู้ใช้งานกรอกข้อความลงไป เมื่อกรอกข้อความแล้วกดปุ่มแก้ไขข้อความ จะส่งข้อความผ่าน API ไปแก้ไข และจะรับผลลัพธ์ไปไว้ในช่องแสดงผลลัพธ์ ซึ่งมีปุ่มคัดลอกข้อความ ใช้ในการคัดลอกข้อความในช่องข้อความแสดงผลลัพธ์ ปุ่มรีเซ็ตใช้กดรีเซ็ตค่าในช่องกล่องข้อความทุกช่อง ดังรูปที่ 7

รูปที่ 7 แสดงหน้าแอปส่วนติดต่อผู้ใช้

ผลการวิจัยและวิจารณ์ผล

1. การประเมินคุณภาพของข้อความที่ได้รับการแก้ไข

การประเมินคุณภาพของข้อความที่ผ่านการแก้ไขแสดงผลโดยใช้อัลกอริทึม BLEU Score ประเมินผลและใส่ค่าลงในตาราง โดย BLEU Score เป็นอัลกอริทึมที่ใช้ประเมินคุณภาพของข้อความโดยนำข้อความที่แปลด้วยเครื่องมาเปรียบเทียบกับข้อความที่แปลอ้างอิงโดยมนุษย์ว่ามีความตรงกันมากแค่ไหน โดยมีช่วงคะแนนตั้งแต่ 0 ถึง 1 ถ้าได้ BLEU คะแนนเท่ากับ 1 หมายความว่าข้อความที่มีการเปรียบเทียบกันมีความเหมือนกันทั้งข้อความ การเปรียบเทียบนั้นใช้การจับคู่ของ N-gram โดยเราใช้ unigram คือ การเทียบคำต่อคำ bigram คือ การเทียบทีละสองคำ 3-gram คือ การเทียบทีละสามคำ 4-gram คือ การเทียบทีละสี่คำ ซึ่งบทความนี้ใช้การคำนวณแบบ unigram

การคำนวณ BLEU score แบบ unigram มีการคำนวณดังนี้

p_n คือ การคำนวณ N-Gram เป็นการวัดแบบ Precision โดยนับจากจำนวน n-gram ที่เลือกเทียบกับ (unigram, bigram, 3-gram, 4-gram) ที่เกิดขึ้นหารด้วยจำนวนคำทั้งหมด เรียกว่า modified n-gram precision ซึ่งมีสูตรการคำนวณที่แปลงมาจาก Precision ดังสมการที่ 1

$$p_1 = \frac{\sum_{C \in \{Candidate\}} \sum_{n-gram \in C} Count_{clip}(n-gram)}{\sum_{C \in \{Candidate\}} \sum_{n-gram \in C} Count(n-gram)} \quad (1)$$

การคำนวณค่า p เมื่อคำนวณ Bleu Score แบบ unigram ได้ดังสมการที่ 2

$$p_1 = \frac{\sum_{c \in \{Candidate\}} \sum_{unigram \in c} Count_{clip}(unigram)}{\sum_{c \in \{Candidate\}} \sum_{unigram \in c} Count(unigram)} \quad (2)$$

$Count_{clip}(unigram)$ คือ จำนวนคำที่มีเหมือนกันน้อยที่สุดเมื่อดูจากการเทียบคำต่อคำของข้อความ Reference กับข้อความ Candidate

$Count(unigram)$ คือ จำนวนคำทั้งหมดของข้อความ Candidate

Brevity Penalty (BP) คือ การคำนวณบทลงโทษเมื่อจำนวนคำของตัวข้อความ Candidate น้อยกว่าจำนวนคำของตัวข้อความ Reference ที่มีจำนวนคำใกล้เคียงกันที่สุด โดยมีการคำนวณตามดังสมการที่ 3

$$BP = \begin{cases} 1 & \text{เมื่อ } c > r \\ e^{(1-\frac{r}{c})} & \text{เมื่อ } c \leq r \end{cases} \quad (3)$$

โดย c คือ จำนวนคำของ Candidate

r คือ จำนวนคำของ Reference

สมการคำนวณ BLEU score ดังสมการที่ 4

$$BLEU = BP \cdot \exp(\sum_{n=1}^N w_n \log p_n) \quad (4)$$

โดย w_n คือ น้ำหนักของ n -gram

แทนค่าการคำนวณ Bleu Score แบบ unigram ได้ดังสมการที่ 5

$$BLEU = BP \cdot \exp(\sum_{n=1}^N w_1 \log p_1) \quad (5)$$

2. ผลคะแนนของข้อความตัวอย่างที่ได้รับการแก้ไขแล้วกับข้อความต้นโดยใช้ BLEU Score

จากข้อความต้นที่เขียนว่า สวัสดิ์วันอังคารขอให้พบพานกับสิ่งที่ได้ ผลลัพธ์เมื่อรู้จำอักขระแล้วได้ผลออกมาเป็น สวัสดิ์อันอ้วคคะขอให้พบพานกับสิ่งที่ได้ ผลคะแนน BLEU Score เป็นดังตารางที่ 8

ตารางที่ 8 ผลคะแนน BLEU ของข้อความที่ได้จากการรู้จำอักขระและข้อความต้น

ข้อความ	BLEU Score
สวัสดิ์วันอังคารขอให้พบพานกับสิ่งที่ได้	1.0
สวัสดิ์อันอ้วคคะขอให้พบพานกับสิ่งที่ได้	0.21

จากผลการรู้จำอักขระในตารางที่ 8 เริ่มแก้ไขโดยวิธีแรก คือ การแก้ไขคำให้สามารถอ่านได้ เขียนโปรแกรมโดยอ้างอิง มาจากวิธีการออกเสียงพยางค์ภาษาไทยมาแก้ไขข้อความ เมื่อใช้ BLEU Score คำนวณคะแนนจากขั้นตอนการแก้ไขคำให้สามารถอ่านได้ ได้ผลตามตารางที่ 9

ตารางที่ 9 ผลคะแนน BLEU ของข้อความที่ได้จากการรู้จำอักขระ ข้อความต้นและข้อความที่ผ่านวิธีการแก้ไขคำให้สามารถอ่านได้

ข้อความ / วิธี	BLEU Score
สวัสดิ์วันอังคารขอให้พบพานกับสิ่งที่ได้	1.0
สวัสดิ์อันอ้วคคะขอให้พบพานกับสิ่งที่ได้	0.21
สวัสดิ์อันอ้วคคะขอให้พบพานกับสิ่งที่ได้	0.21

จากตารางที่ 9 เมื่อแก้ไขข้อความโดยวิธีแรกพบว่าคะแนน BLEU ได้เท่าเดิม คือ 0.21 เนื่องจากการแก้ไขทำให้สามารถอ่านได้นั้น ใช้วิธีที่อ้างอิงมาจากการออกเสียงพยางค์ภาษาไทย สิ่งที่เกิดขึ้นได้นั้นกลายเป็นผลลัพธ์ที่คล้ายเดิมแค่ปรับปรุงพยางค์ให้สามารถอ่านออกเสียงได้แล้วเท่านั้น ตัวข้อความที่แก้ไขไม่ได้เปลี่ยนแปลงไปจากข้อความที่ได้จากการรู้จำอักขระมากนัก หลังจากแก้ไขด้วยขั้นตอนแรกแล้วจึงแก้ไขต่อด้วยขั้นตอนที่สอง คือ การแก้คำที่อ่านได้ให้เป็นคำที่ควรจะเป็น โดยใช้ไลบรารี difflib ในการเลือกคำที่คล้ายกันจากพจนานุกรมกับคำที่ปรับปรุงให้อ่านได้ เมื่อใช้ BLEU score คำนวณคะแนนได้ผลดังตารางที่ 10

ตารางที่ 10 ผลคะแนน BLEU ของข้อความที่ได้จากการรู้จำอักขระ ข้อความต้น ข้อความที่ผ่านวิธีการแก้ไขคำให้สามารถอ่านได้ และข้อความที่ผ่านวิธีการแก้ไขคำที่อ่านได้ให้เป็นคำที่ควรจะเป็น

ข้อความ / วิธี	BLEU Score
สวัสดีวันอังคารขอให้พบพานกับสิ่งที่ดี	1.0
สวัสดีวันอาทิตย์ขอให้พบพานกับสิ่งที่ดี	0.21
สวัสดีวันอาทิตย์ขอให้พบพานกับสิ่งที่ดี	0.21
สวัสดีวันอาทิตย์ขอให้พบพานกับสิ่งที่ดี	0.66

จากตารางที่ 10 เมื่อแก้ไขด้วยวิธีการที่ 2 พบว่าคะแนนความเหมือนกันกับข้อความต้นที่ได้เพิ่มขึ้นมาจาก 0.21 เป็น 0.66 คะแนน หมายความว่ามีความเหมือนกันกับประโยคต้น 66% ที่แก้ไขออกมาได้ไม่ผิดนัก เนื่องจากผลลัพธ์ที่ได้จากการรู้จำอักขระที่ได้มีบางคำที่ไม่เหมือนเค้าโครงเดิมของข้อความต้นเอาไว้ ทำให้การแก้ไขคำให้เป็นคำที่ควรจะเป็นนั้นจะได้ผลออกมาไม่ดีตามไปด้วย

สรุปผลการวิจัย

บทความนี้นำเสนอเว็บแอปพลิเคชันเพื่อลดคำผิดที่เกิดจากการรู้จำอักขระลายมือเขียนภาษาไทย โดยแก้ไขคำที่ไม่สามารถอ่านได้ให้สามารถอ่านได้ตามหลักวิธีการออกเสียงพยางค์ภาษาไทย และใช้ไลบรารีไพธอน difflib มาหาคำที่มีรูปใกล้เคียงเพื่อแก้ไขคำที่อ่านได้แล้วแต่ยังเป็นคำผิดอยู่ให้เป็นคำที่ควรจะเป็น สามารถเพิ่มความถูกต้องของข้อความได้ในระดับหนึ่ง เมื่อใช้ BLEU Score ได้รับผลคะแนนทดสอบความตรงกันของข้อความต้นที่นำมารู้จำอักขระกับผลลัพธ์ที่แก้ไขออกมาได้คะแนนประมาณ 0.66 โดยเมื่อดูจากผลทดสอบสามารถตีความได้ว่าหากผลลัพธ์การรู้จำอักขระคงเค้าโครงเดิมของข้อความต้นไว้การแก้ไขจะทำให้ค่อนข้างดี หากไม่มีการแก้ไขจะได้ผลลัพธ์ออกมาไม่ผิดนัก

เอกสารอ้างอิง

- Aiman, K., Qamar, U., Zafar, I. and Shaheen, A. (2018). Automated misspelling detection and correction in clinical free-text records. In: 2018 International Conference on Artificial Intelligence and Big Data (ICAIBD). IEEE, Chengdu. 277 - 280. doi: 10.1109/ICAIBD.2018.8396209.
- Chumwatana, T., Rattana-amnuaychai, W. and Chauychu, P. (2022). Patient Information Extraction Using Optical Character. Journal of The Thai Medical Informatics Association 8(1): 22 - 27.
- Islam, M.M., Kabir, M.N., Sadi, M.S., Morsalin, I., Haque, A. and Wang, J. (2019). A Novel Approach Towards Tamper Detection of Digital Holy Quran Generation. In: Conference: 5th International Conference on Electrical, Control and Computer Engineering (InECCE2019). Pahang, Malaysia. 297 - 308. doi: 10.1007/978-981-15-2317-5_25.

- Meesad, P., Kleechaya, P., Aun-a-nan, A. and Kijrunpaisarn, K. (2022). Artificial Intelligent Techniques for Thai Fake News Detection. The Journal of Applied Science 21(1): 1 - 19. doi: 10.14416/j.appsci.2022.01.012.
- Ngamcharoen, P., Sanglerdsinlapachai, N. and Vejjanugraha, P. (2022). Automatic Thai Text Summarization Using Keyword-Based Abstractive Method. In: 2022 17th International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP). IEEE, Chiang Mai. 1 - 5. doi: 10.1109/iSAI-NLP56921.2022.9960265.
- Pal, A., Mallick, S. and Pal, A. R. (2021). Detection and Automatic Correction of Bengali Misspelled Words using N-Gram Model. In: 2021 International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT). IEEE. 1 - 5. doi: 10.1109/ICAECT49130.2021.9392406.
- Papineni, K., Roukos, S., Ward, T. and Zhu, W-J. (2002). Bleu: a Method for Automatic Evaluation of Machine Translation. Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia. 311 - 318. doi: 10.3115/1073083.1073135.
- Phaphan, W. and Pimpisal, A. (2020). The predictions of a daily stock price direction from the Thai news content by using natural language processing. The Journal of Applied Science 19(1): 59 - 79. doi: 10.14416/j.appsci.2020.01.006.
- Puttipornchai, C., Chanyachatchawan, S. and Tuaycharoen, N. (2022). Multi-Label Classification for Articles in Thai Journal Database from Article's Abstract. In: 2022 19th International Joint Conference on Computer Science and Software Engineering (JCSSE), IEEE. 1 - 6. doi: 10.1109/JCSSE54890.2022.9836270.
- Tanaka, Y., Murawaki, Y., Kawahara, D. and Kurohashi, S. (2020). Building a Japanese Typo Dataset from Wikipedia's Revision History. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Student Research Workshop. Association for Computational Linguistics. 230 - 236. doi: 10.18653/v1/2020.acl-srw.31.

