การใช้โทคเกนพาสซิงอัลกอริทึมในการแก้ไขคำผิดใน OCR ภาษาไทย

THAI OCR ERROR CORRECTION USING TOKEN PASSING ALGORITHM

มนัส รอดพ้น
MANUS RODPHON

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า
บัณฑิตวิทยาลัย
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2546

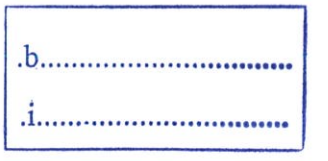
ISBN 974-324-303-8

การใช้โทคเกนพาสซิงอัลกอริทึมในการแก้ไขคำผิดใน OCR ภาษาไทย

THAI OCR ERROR CORRECTION USING TOKEN PASSING ALGORITHM

มนัส รอดพ้น

MANUS RODPHON



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2546

ISBN 974 - 324 - 303 - 8

THAI OCR ERROR CORRECTION USING TOKEN PASSING ALGORITHM

MANUS RODPHON

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING

SCHOOL OF GRADUATE STUDIES

KING MONGKUTS INSTITUTE OF TECHNOLOGY LADKRABANG

2003

ISBN 974 - 324 - 303 - 8

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองวิทยานิพนธ์

หัวข้อวิทยานิพนธ์ การใช้โทคเกนพาสซิงอัลกอริทึมในการแก้ไขคำผิดใน OCR ภาษาไทย
THAI OCR ERROR CORRECTION USING TOKEN PASSING ALGORITHM

ชื่อนักศึกษา นายมนัส รอดพ้น

รหัสประจำตัว 42061131

ปริญญา วิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชา วิศวกรรมไฟฟ้า

อาจารย์ผู้ควบคุมวิทยานิพนธ์ รศ.ดร.บุญธีร์ เครือตราชู

คณะกรรมการสอบวิทยานิพนธ์		ลายมือชื่อ
รศ.ดร.บุญวัฒน์	อัตชู	
ดร.วรวัฒน์	ลิ้มโภคา	
ดร.สุรินทร์	กิตติธรกุล	
อาจารย์กฤตวัน	ศิริบูรณ์	
รศ.ดร.บุญธีร์	เครือตราชู	

วัน/เดือน/ปี ที่สอบ 24 กุมภาพันธ์ 2546 เวลา 10.30-12.30 น.

สถานที่สอบ ณ อาคาร 12 ชั้น ชั้น 4 (ห้อง E12-403)

บัณฑิตวิทยาลัยรับรองแล้ว

(รศ.ดร.บุญวัฒน์ อัตชู)

คณบดีบัณฑิตวิทยาลัย

วันที่.....24.....**เดือน**.....กุมภาพันธ์.....**พ.ศ**.....2546.....

หัวข้อวิทยานิพนธ์	การใช้โทคเกนพาสซิงอัลกอริทึมในการแก้ไขคำผิดใน OCR ภาษาไทย
นักศึกษา	มนัส รอดพ้น
รหัสประจำตัว	42061131
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมไฟฟ้า
พ.ศ.	2546
อาจารย์ผู้ควบคุมวิทยานิพนธ์	รศ.ดร. บุญธีร์ เครือตราชู

บทคัดย่อ

การตรวจสอบคำสะกดสามารถนำมาเพิ่มความถูกต้องผลลัพธ์ของ OCR ,ซึ่งจะใช้เวลามากสำหรับภาษาไทย เนื่องจากไม่มีขอบเขตที่แบ่งระหว่างแยกคำอย่างชัดเจน, การตรวจสอบคำสะกดจึงต้องทำการตรวจสอบทุกตัวอักษรที่ติดกัน และทุกคำที่เป็นไปได้ ซึ่งยังมีความคลุมเครือในการแบ่งแยกขอบเขตคำต่อเนื่องกันไป ในงานวิจัยนี้จะนำเสนอ โทคเกนพาสซิง อัลกอริทึม, ซึ่งมักในปัญหาการรู้จำเสียงคำพูด, เข้ามาช่วยในการแก้ไขปัญหานี้ โดยจะใช้ผลลัพธ์จาก OCR ซึ่งประกอบด้วยตัวอักษรซึ่งมีค่าความน่าจะเป็นสูงที่สุด ห้าตัว โทคเกนจะถูกสร้างขึ้นมา จากตัวอักษรแต่ละตัวและส่งผ่านต่อไปรวมกับตัวอักษรชุดต่อไป ในแต่ละครั้งที่โทคเกนจะถูกส่งไป, จะทำการตรวจสอบตรวจสอบคำสะกดของแต่ละโทคเกนด้วยพจนานุกรม, โทคเกนที่มีคำซึ่งสะกดผิดจะถูกทิ้งไป ส่วนโทคเกนซึ่งเป็นคำที่สมบูรณ์แล้วจะถูกใช้ในการสร้างกราฟของคำ, ซึ่งจะประกอบด้วยคำทั้งหมด ซึ่งมาจากตัวอักษรของแต่ละโทคเกน ที่มีลำดับคะแนนสูงที่สุดห้าตัวอักษร โทคเกนพาสซิง อัลกอริทึมจะถูกนำมาใช้อีกครั้งในระดับของคำในการเลือกประโยคที่มีความความน่าจะเป็นสูงที่สุด

Thesis Title	Thai OCR Error Correction using Token passing algorithm
Student	Mr. Manus Rodphon
Student ID.	42061131
Degree	Master of Engineering
Programme	Electrical of Engineering
Year	2003
Thesis Advisor	Assoc.Prof.Dr. Boontee Kruatrachue

ABSTRACT

Spell checking can be used to improve OCR result, which is quite time consuming for Thai language. Since, there is no explicit word boundary, the spell checking has to go through all possible ambiguity characters and ambiguity word boundary. This paper proposed a Token Passing algorithm, often used in speech recognition, to this problem. The output of the OCR consists string of 5 most probable characters. The letter token are generated for each letters and passed to the next 5 characters. Each time the token is parsed, the dictionary is used to check for the correct spelling. The wrong spelling token are discarded. Tokens with complete word are used to construct words graph, which will be fully constructed when letter token reach the best five characters. Token passing Algorithm is used again in the word level to select the best possible sentence.

กิตติกรรมประกาศ

กว่าบทความฉบับนี้จะเริ่มต้นและเป็นรูปเป็นร่างขึ้นมาได้ก็เพราะด้วยรับความเมตตาจาก ผศดร. บุญธีร์ เครือตราชู และภรรยา รวมถึง อ.กฤตวัน และ อ.ท่านอื่น ที่มิได้กล่าวถึงไว้ในที่นี้ หลายท่านที่ได้ให้ความกรุณาสละเวลาเพื่อให้คำแนะนำและปรึกษาแนวทางการแก้ปัญหาต่าง ๆ ตลอดการทำวิจัยนี้ ผู้วิจัยจึงขอกราบขอบพระคุณท่าน อ ทั้งหลายไว้เป็นอย่างสูง

ขอขอบคุณ บิดา, มารดา, น้องสาว ที่เป็นกำลังสนับสนุนกำลังใจและกำลังทุนทรัพย์ในการดำเนินชีวิตและการเรียน

ขอขอบคุณ เพื่อน ๆ ที่เป็นกำลังใจ และคำแนะนำเกี่ยวกับ OCR

ขอขอบคุณ น้อง ๆ เพื่อน ๆ ชุมนุมคอมพิวเตอร์ที่คอยเป็นห่วงเป็นใย

ขอขอบคุณ หลาย ๆ คนที่สระว่ายน้ำที่ช่วยคลายเครียดทุกวันตอนเย็น

ขอขอบคุณ คุณศุภชัย บริษัทไทยไพพรรณที่ให้เวลากลับมาทำงานวิจัย

และขอขอบคุณอีกหลาย ๆ ท่านที่ไม่ได้กล่าวถึงในที่นี้ซึ่งช่วยให้ผู้เขียนได้สามารถเขียนบทความฉบับนี้ให้ลุล่วงลงได้

มนัส รอดพ้น

สารบัญ

หน้า

สารบัญตาราง

ตารางที่ หน้า

สารบัญรูป

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

การเพิ่มความถูกต้องของ OCR ภาษาไทยนั้นมีความสำคัญมากเนื่องจากผลลัพธ์ที่ได้จากการรู้จำตัวอักษรภาษาไทยนั้นยังมีความผิดพลาดอยู่มาก สำหรับความถูกต้องของผลลัพธ์ที่ได้การรู้จำของ OCR นั้นต้องการความถูกต้องมากกว่า 99% ขึ้นไป ซึ่งในการที่จะได้ผลตามที่ต้องการนี้สามารถเป็นไปได้ ซึ่งในการที่จะสร้าง OCR ให้ได้ความถูกต้อง 100% จะง่ายขึ้น ถ้าผลลัพธ์ของการรู้จำนั้นอยู่ในกลุ่มของตัวอักษรที่เป็นไปได้ดังจะแสดงให้เห็นในบทต่อไป

ความผิดพลาดที่เกิดขึ้นจากขบวนการ OCR มนุษย์สามารถที่จะแก้ไขด้วยตนเองได้โดยไม่ยากลำบาก แต่จะใช้เวลามากถ้าเป็นเอกสารขนาดใหญ่ ดังนั้นถ้าหาก OCR มีความถูกต้องมากขึ้นจะเป็นการช่วยประหยัดเวลาและทรัพยากรเป็นอย่างมาก

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

งานวิจัยนี้ได้พยายามที่จะทำให้ผลลัพธ์ที่ได้จาก OCR มีความถูกต้องเพิ่มขึ้น โดยแก้ไขคำที่ผิดพลาดจากพจนานุกรม และแบบจำลองภาษาทางสถิติ โดยการใช้ชุดตัวอักษรที่เป็นผลลัพธ์จาก OCR และค่าคะแนนของแต่ละตัวอักษรที่เป็นไปได้สูงสุด 5 ตัว (ไม่เกิน 5) ในแต่ละตำแหน่งของตัวอักษร แล้วประยุกต์เทคนิค Token passing ในการสร้างคำ สร้างประโยค จนถึงขั้นการตรวจสอบความถูกต้อง โดยตรวจสอบความถูกต้องของคำโดยใช้พจนานุกรม และตรวจสอบความถูกต้องในระดับประโยคโดยใช้แบบจำลองภาษาทางสถิติ

1.3 สมมติฐานของการศึกษาและแนวความคิดที่ใช้ในงานวิจัย

มนุษย์สามารถแก้ไขคำผิดได้เนื่องจากมนุษย์มีความรู้ในด้านภาษา ซึ่งมนุษย์สามารถเรียนรู้สิ่งเหล่านี้ได้ในเวลาไม่กี่ปี ซึ่งสิ่งนี้ก็ยังคงเป็นปริศนาอยู่ว่ามนุษย์เราสามารถเรียนรู้สิ่งเหล่านี้ได้อย่างไร แม้เทคโนโลยีด้านคอมพิวเตอร์ในปัจจุบันจะมีความก้าวหน้าไปมากก็ตาม คอมพิวเตอร์ก็ยังไม่สามารถที่จะทำการลอกแบบความสามารถในการเรียนรู้ของมนุษย์ได้ ดังนั้นในการจะทำให้คอมพิวเตอร์สามารถรู้จักภาษามนุษย์ได้จึงต้องสร้างแบบจำลองภาษาขึ้นมา งานศึกษาวิจัยที่

เกี่ยวข้องกับการเรียนรู้ภาษามนุษย์ด้วยคอมพิวเตอร์ มักจะเป็นทฤษฎีทางคณิตศาสตร์ ที่ใช้ในการเรียนรู้ทางภาษา และสำหรับงานวิจัยนี้ได้ใช้มุมมองทางสถิติของภาษา และประยุกต์เอาหลักการทางสถิติเพื่อนำมาใช้ในการสร้างแบบจำลองทางภาษา ซึ่งแบบจำลองทางภาษานี้เองจะเป็นกระบวนการสำคัญในการแก้ไขคำผิดของงานวิจัยนี้ และกระบวนการที่สำคัญอย่างหนึ่งในการทำงานวิจัยนี้นั่นคือ โทคเกนพาสซิง อัลกอริทึม ซึ่งได้ถูกนำมาประยุกต์ใช้ตั้งแต่สร้างคำจากตัวอักษร จนไปถึงขั้นตอนการสร้าง ประโยคจนเสร็จสมบูรณ์

1.4 ขอบเขตงานวิจัย

ในงานวิจัยได้สร้างแบบจำลองทางสถิติของภาษาเพื่อเปรียบเทียบ bi-gram tri-gram จนถึง 6-gram สำหรับภาษาไทยและได้รวมรวบฐานข้อมูล corpus ไว้เพื่อใช้ในงานวิจัยอื่น ๆ ต่อไป

1.5 ขั้นตอนการศึกษา

สำหรับในวิทยานิพนธ์ฉบับนี้ได้เรียบเรียงเรื่องไว้ดังนี้

บทที่ 2
งานวิจัยที่เกี่ยวข้อง

2.1 ภาษาเขียน

คุณสมบัติทั่วไปที่สำคัญของภาษาเขียน

1. ประกอบด้วยรูปภาพที่ประดิษฐ์ขึ้นเป็นเครื่องหมายบนพื้นผิวใด ๆ
2. มีจุดประสงค์เพื่อการสื่อสารถึงสิ่งใดสิ่งหนึ่ง
3. โดยบรรลุถึงจุดมุ่งหมายนี้ ด้วยเหตุ ระเบียบแบบแผนความสัมพันธ์ของเครื่องหมายซึ่งสัมพันธ์กับภาษา [2]

ความแตกต่างในระบบภาษาเขียนได้แก่, รูปแบบ, การแสดงหน่วยในภาษา, คำ, พยางค์ และเสียง ตัวอย่างในระบบการเขียนต้นแบบเครื่องหมาย หรือตัวอักขระของ ละติน, กรีก, รัสเซีย เป็นต้นแบบของอักษรที่ใช้ในหลายภาษาเช่น อังกฤษ,ดัช , ฝรั่งเศส , ฯลฯ ซึ่งลักษณะตัวอักษรจะรับมาจากตัวอักขระของ ละติน สำหรับภาษา Devanagari , ซึ่งอักขระใช้เป็นการแสดงรูปของเสียง , เช่นภาษา อินเดีย รวมทั้งฮินดู สำหรับภาษาจีนจะเป็นภาษารูปภาพ ส่วนในภาษาญี่ปุ่นนั้นจะมีตัวคันจิซึ่งเป็นรับเอามาจากภาษาจีน และตัว คานะ ซึ่งเป็นเป็นตัวซึ่งใช้เขียนแทนเสียงซึ่งใช้กับคำที่มาจากต่างชาติ

แต่ละภาษาก็มีตัวอักขระของตัวเอง ซึ่งเรียกว่าตัวอักษรหรือตัวอักขระ ซึ่งเป็นรูปร่างพื้นฐาน แต่ละภาษาก็จะมีรูปแบบและกฎในการนำตัวอักษรแต่ละตัวมาเรียงต่อกัน

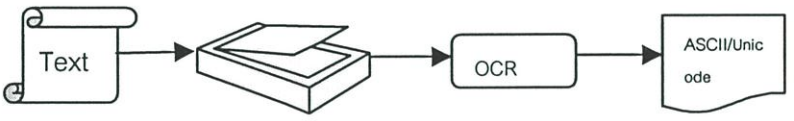
ประมาณศตวรรษที่ 15 ได้เกิดการประดิษฐ์คิดค้นสื่อสิ่งพิมพ์ขึ้นเป็นครั้งแรก ซึ่งได้มีการใช้อัลลอยเป็นแม่แบบตัวอักษร ใช้กลไกในการพิมพ์ และใช้หมึกพิมพ์

เอกสารบนแผ่นกระดาษ, ซึ่งอยู่ในรูปแบบของสื่อซึ่งเป็น อนาลอก, สามารถที่จะเปลี่ยนไปอยู่ในรูปแบบของดิจิตอล, โดยใช้กระบวนการสแกน ซึ่งในกระบวนการนี้ถ้าเป็นเอกสารขนาด 8.5x11 นิ้ว สแกนโดยใช้ความละเอียดที่ 300 จุดต่อนิ้ว (dpi) ในภาพแบบขาว-ดำ จะมีขนาด 8.4 เมกาไบต์

เอกสารที่ได้จากการสแกนนี้นอกจากจะมีขนาดใหญ่ นอกจากเป็นปัญหาในการเก็บซึ่งทำให้เปลืองพื้นที่แล้วยังทำให้การค้นหาและการแก้ไขข้อมูลทำได้ด้วยความลำบาก และขาดประสิทธิภาพจึงได้เกิดแนวความคิดในการเปลี่ยนตัวอักษรซึ่งจัดเก็บไว้ในลักษณะที่เป็นรูปภาพมาเป็นไฟล์ที่เป็นข้อความ

2.2 การรู้จำ (Recognition)

การรู้จำตัวอักษรเป็นการแปลงรูปแบบของภาษาจากลักษณะรูปภาพ/เครื่องหมายที่อยู่บนแผ่นกระดาษหรือพื้นผิวใด ๆ เป็น สัญลักษณ์ที่ใช้แสดงถึงตัวอักษรนั้น ๆ เช่นอยู่ในรูปแบบของ Unicode หรือ ASCII ดังแสดงในรูปที่ 2.1



รูปที่ 2.1 แสดงการแปลงข้อความบนกระดาษเป็น ASCII

โดยเฉพาะสำหรับภาษาซึ่งเขียนการแต่ละคำติดกันไปโดยไม่มีช่องว่างหรือเครื่องหมายเว้นระหว่างคำ เช่น ไทย จีน ญี่ปุ่น ทำให้การตรวจสอบคำสะกดสำหรับภาษาไทยจำเป็นต้องมีการตัดแบ่งคำก่อนทำการวิเคราะห์ความถูกต้องของคำ

สำหรับการแก้ไขคำผิด มีกรรมวิธีในการแก้ไขอยู่หลายวิธี ขึ้นอยู่กับชนิดของความผิดพลาดและลักษณะการเขียน[14] โดยเฉพาะประโยคในภาษาไทย นั้นจะไม่มีการแบ่งแยกระหว่างคำให้เห็นอย่างชัดเจน จนกว่าจะจบประโยค หรือมีการเว้นวรรค หรือ ขึ้นย่อหน้าใหม่ การตรวจสอบคำสะกดภาษาไทยนั้นจึงจะต้องทำกับตัวอักขระทุกตัวที่มีความคลุมเครือและกับทุกคำที่มีขอบเขตคลุมเครือเท่าที่เป็นไปได้

การทราบขอบเขตคำในประโยคจึงมีความจำเป็นอย่างยิ่งในการตัดแบ่งคำ[13] ด้วยลักษณะการเขียนในภาษาไทย เป็นการยากที่จะทำให้ทราบว่าเราควรจะตัดแบ่งคำอย่างไร หรือ มีคำใดที่มีคำผิดอยู่ในประโยค

จากผลงานวิจัยที่ผ่านมาการตรวจสอบตัวสะกดในภาษาไทยจะใช้หลักการทางอักขระวิธีและวิธีการตรวจสอบตัวสะกดโดยใช้หลักการเปรียบเทียบกับพจนานุกรม

ซึ่งการตรวจสอบทางอักขรวิธีนี้ จะสามารถตรวจสอบได้รวดเร็ว แต่จะมีความถูกต้องน้อย เพราะอาจจะพบคำที่ผิดแต่เป็นไปตามกฎ

ส่วนการตรวจสอบกับพจนานุกรมจะเป็นการนำคำในประโยคมาเปรียบเทียบกับคำในพจนานุกรม ซึ่งมีข้อดีคือ มีความแม่นยำ ถูกต้องสูง แต่ความเร็วในการประมวลผลจะช้ากว่าแบบแรก[15]

ซึ่งหากเอกสารต้นฉบับมีคุณภาพในการพิมพ์ต่ำ หรือต้นฉบับมีคุณภาพไม่ดี หรือคุณภาพที่ได้จากการสแกนต่ำ จะทำให้การรู้จำของตัวอักษรเกิดความผิดพลาดสูงขึ้น

2.3 ลักษณะคำผิดที่เกิดในงานเอกสารสำหรับเอกสารซึ่งเกิดความผิดพลาดจาก OCR[4]

1. ตัวอักษรเกิน เช่น การรณนรงค์
2. ตัวอักษรผิด เช่น ถารรณรงค์
3. ตัวอักษรขาด เช่น การรณรงค

ซึ่งในงานวิจัยนี้ ได้ทำการแก้ไขความผิดพลาดเฉพาะในกรณีที่ 2

2.4 มนุษย์สามารถอ่านตัวอักขระที่ไม่ชัดเจนได้อย่างไร

เราสามารถแยกแยะความแตกต่างระหว่างตัวอักษรได้อย่างง่ายดาย ถึงแม้ เอกสารนั้นจะมีคุณภาพค่อนข้างต่ำ ส่วนหนึ่งเนื่องจากเมื่อเจอคำที่อ่านแล้วเราไม่สามารถอ่านได้เราก็จะนึกถึงตัวอักษรที่ใกล้เคียงกับตัวนั้นๆ และลองสับเปลี่ยนในใจ ถ้าแก้ไขแล้วสามารถอ่านได้ใจความก็น่าจะเป็นคำที่เราคิดไว้ หรือถ้าเปลี่ยนไปแล้วไม่เข้ากับใจความรอบข้างก็จะลองเทียบเคียงกับคำอื่นที่ลักษณะใกล้เคียงกัน

-ใช้ความรู้เกี่ยวกับการสร้างคำ
-ใช้ตัวอักษรที่ลักษณะใกล้เคียงกัน
-ใช้ความรู้เกี่ยวกับความหมายของคำ
-ใช้ความคุ้นเคยกับคำและประโยค

สำหรับลักษณะของตัวอักษรที่คล้ายกัน[22] เช่น

- คด ดต ชช บป พฟ ผฝ ฏฎ ทฑ มฆ ฟพ ณฒ อฮ คศ อิอึ อีอื อีอื ถภ กภ ภก
  คด ดต ชช บป พฟ ผฝ ฏฎ ทฑ มฆ ฟพ ณฒ อฮ คศ อิอึ อีอื อีอื ถภ กภ ภก
  คด ดต ชช บป พฟ ผฝ ฏฎ ทฑ มฆ ฟพ ณฒ อฮ คศ อิอึ อีอื อีอื ถภ กภ ภก
  คด ดต ชช บป พฟ ผฝ ฏฎ ทฑ มฆ ฟพ ณฒ อฮ คศ อิอึ อีอื อีอื ถภ กภ ภก

จากตัวอย่างข้างบนนี้จะพบว่ามีแบบตัวอักษรหลายแบบซึ่งคล้ายกันมากโดยเฉพาะบางคู่ สำหรับบาง แบบของตัวอักษร มีความแตกต่างกันน้อยมาก
เช่น ภาพ --> กาพ , บรรทัด -->บรรทัด

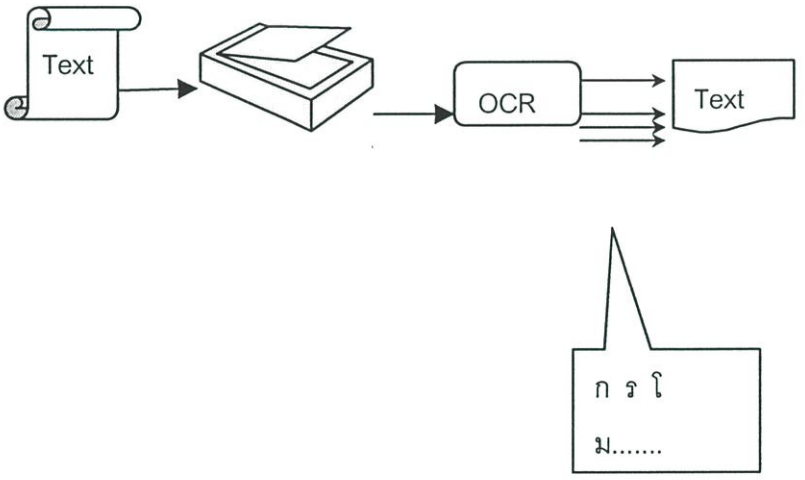
ซึ่งตัวอักษรที่ถูกต้องน่าจะอยู่ในผลลัพธ์ของ OCR ซึ่งมีค่าคะแนนรองลงมา เช่น

ก .70 ภ .68 ถ .50 ฤ .42 ฏ .32 ฏ .33 .....

า .80 ว .75 จ .40 ๅ .30 ๆ .30

พ .76 ผ .74 ฟ .71 ฝ.64 ฬ .50

ตามปกติ OCR ทั่วไปจะเลือกเอาตัวที่มีคะแนนสูงที่สุดในแต่ละตัวออกมา เช่นจากตัวอย่างข้างบนจะได้คำว่า กาพ ออกมา จึงเกิดแนวคิดในการแก้ไขคำที่ผิดพลาด โดยแทนที่จะใช้ตัวอักษรที่วิเคราะห์ได้จาก OCR เพียงตัวเดียวต่อหนึ่งตำแหน่งเราน่าจะรับมาทั้งชุดเพื่อใช้แก้ไขความผิดพลาดในขั้นตอนต่อไป



รูปที่ 2.2 แสดงการรับตัวอักษรเป็นชุดจาก OCR

การคำนวณเพื่อจัดการกับตัวอักษรที่ทำให้เกิดความกำกวม อยู่บนพื้นฐานของสมมุติฐานของเนื้อความของสารสนเทศน์นั้น ว่ามีเพียงใด

ถ้าตัวอักษรนั้นมีความเป็นอันหนึ่งอันเดียวกันและทำให้เกิดเป็นคำที่มีความหมายและสอดคล้องกัน สารสนเทศน์นั้นจะ สูง

ตรงกันข้าม ถ้าหากว่าตัวอักษรนั้นไม่สามารถอ่านได้หรือ อ่านออกมาแล้วไม่สามารถตีความได้หรือไม่มีความสอดคล้องกับคำอื่นใกล้เคียงกัน สารสนเทศน์นั้นจะ ต่ำ

ยกตัวอย่างเช่นหากมีข้อความ “กต” จะถือว่าสารสนเทศน์หรือข้อความนี้มีค่า ต่ำ เนื่องจาก กต ไม่มีความหมายหรือสารสนเทศน์ที่สื่อความหมายใด ๆ

จากตัวอย่างที่ยกมานี้ง่ายในการตรวจสอบเนื่องจากเป็นคำ แต่ถ้าเป็นประโยคยาวกว่าหนึ่งคำปัญหาหนึ่งคือการไม่ทราบขอบเขตที่แน่นอนของคำ เนื่องจากภาษาไทยนั้นเขียนคำติดต่อกันไปโดยไม่มีการเว้นวรรคหรือใช้สัญลักษณ์พิเศษเพื่อแยกระหว่างคำ ดังนั้นในการตรวจสอบคำสะกดในภาษาไทยจึงจำเป็นต้องผ่านการตัดแบ่งคำ

2.5 การตัดแบ่งคำ

สำหรับในภาษาไทยได้มีงานวิจัยในการตัดแบ่งคำโดยสองวิธีหลักคือ

1. ใช้อักษรวิธี
2. ใช้วิธีตรวจสอบคำจากพจนานุกรม

การใช้อักษรวิธี เป็นการตรวจสอบการสะกดคำโดยใช้ไวยากรณ์ในการเขียนภาษาไทย [5], [8] ,[9]

ข้อดีคือ ไม่จำเป็นต้องมีการฝึกฝนก่อน สามารถใช้งานได้ทันทีและสามารถทำงานได้รวดเร็ว

ข้อเสียคือ คำที่ตัดออกมาอาจเป็นคำที่สะกดผิด หรือตัดคำออกมาโดยไม่มีความหมาย ใด

เช่น คำว่า เผียง และ แผร เป็นคำที่เมื่อตรวจสอบด้วยการเขียนนั้นถูกต้องแต่ ไม่เป็นคำที่มีความหมายในภาษาไทย

ส่วนกรรมวิธีการตรวจสอบขอบเขตคำภาษาไทยด้วยการเทียบคำกับพจนานุกรมและประสิทธิภาพของวิธีต่าง ๆ ได้มีผู้ทำการวิจัยไว้ใน [11]

ตัดคำที่สั้นที่สุด

ตัดคำที่ยาวที่สุด

ใช้คำที่ถูกใช้บ่อยที่สุด

ใช้ back-tracking

เลือกใช้คำที่ทำให้เกิดจำนวนคำน้อยที่สุดในประโยค (Maximal-matching)

ใช้พจนานุกรมพิเศษช่วย สำหรับคำที่ทำให้เกิดความกำกวมซึ่งอาจพิจารณาตัดคำโดยใช้มนุษย์เป็นผู้เลือกการตัดคำ โดยพจนานุกรมสองชุดโดยชุดหนึ่งจะมีคำที่ทำให้เกิดความกำกวมในการตัดประโยค เช่น

รอยกร่าง = รอ ยก ร่าง

มีดอก = มี ดอก

โคลงเรือ = โค ลง เรือ = โคลง เรือ

มาสอบ = มา สอบ

สำหรับงานวิจัยนี้การตัดคำจะใช้ Token passing จะเป็นการเลือกคำที่เป็นไปได้ทั้งหมดซึ่งสามารถทำให้เกิดประโยคที่ต่อเนื่องไปจนจบประโยค ซึ่งจะได้อธิบายในบทต่อไป

มีงานวิจัยอื่น ๆ ที่เกี่ยวข้อง ได้แก่ การใช้พจนานุกรมและ pos tag [5] ,การใช้พจนานุกรมและการจัดเก็บด้วยโครงสร้างแบบ ทรัย เพื่อช่วยในการตัดแบ่งคำและหาขอบเขตของคำที่ไม่พบในพจนานุกรม[6]

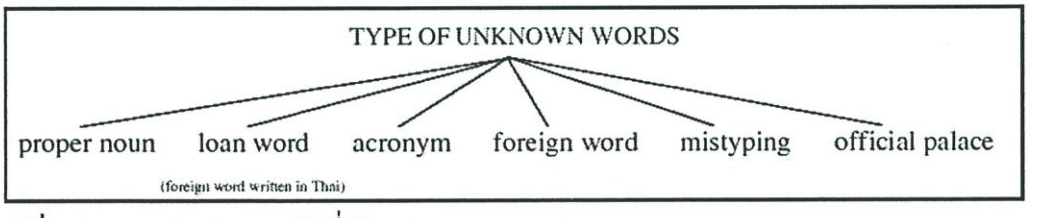
และนอกจากนี้ สำหรับงานวิจัยของต่างประเทศได้มีการใช้เทคนิคอื่น เข้ามาช่วยหรือใช้หลายเทคนิคผสมกันเช่น

ใช้ค่าสถิติของการเรียงกันของตัวอักษร และใช้หลักไวยากรณ์[14]

ใช้ Corpus ที่แบ่งแยกขอบเขตของคำแล้ว

ใช้ระบบผู้เชี่ยวชาญ ซึ่งในกรณีนี้จะสามารถช่วยแก้ไข ในกรณี insertion และ deletion ได้ และสามารถ แก้ไขการเกิด garbage ได้[4]

สำหรับการใช้พจนานุกรมในการตัดคำ คำที่ตรวจสอบไม่พบ(unknown word) ในพจนานุกรมอาจเป็นคำที่ผิด หรืออาจจะไม่ผิดก็ได้ซึ่งได้มีผู้ทำวิจัยคำที่ไม่พบในพจนานุกรมสำหรับเอกสารภาษาไทยไว้ดังนี้



รูปที่ 2.3 แสดงประเภทของคำที่เป็น unknown word

จากงานวิจัย [10] ได้ทำการสุ่มตัวอย่างเอกสาร 5000 คำของหนังสือพิมพ์, รายงานทางวิทยาศาสตร์ และเรื่องทั่วๆ ไป, โดยเฉลี่ยพบว่ามีคำที่เป็น unknown อยู่ประมาณ 15% ซึ่งสามารถแยกแยะได้ดัง รูปที่ 2.3

ในงานวิจัยนี้เรา สนใจการการแก้ไข mistyping ซึ่งความผิดพลาดอาจเกิดขึ้นจากตัวต้นฉบับหรือเกิดจาก OCR

2.5 ลักษณะคำที่ผิดสำหรับ OCR

คำผิดที่ไม่ปรากฏในพจนานุกรม สามารถตรวจสอบได้

"ถาร เพาะ ปลูก"

คำผิดที่ไม่สามารถตรวจสอบได้ด้วยพจนานุกรม

"ภาร เพาะ ปลูก"

คำที่ถูกต้องแต่ตรวจสอบไม่พบในพจนานุกรมเนื่องจากเป็นการลดรูปของคำ เช่น ชีววัสดุ

คำที่ไม่ผิดซึ่งเป็นคำที่ยืมมาจากภาษาต่างประเทศ หรือเป็นชื่อเฉพาะ กรดอะมิโน

2.6 การแก้ไขคำผิด

ได้มีงานวิจัยอยู่หลายแนวทางด้วยกันในการแก้ไขคำผิดได้แก่

-การใช้ไวยากรณ์ในการตรวจสอบความถูกต้องและแก้ไขคำผิด[16]

-การใช้ระบบผู้เชี่ยวชาญ[4]

-การใช้วิธี parsing [21]

-การใช้ winnow [335]

-การใช้ tri-gram และ winnow [14]

-การใช้กฎ [19]

-การใช้ ระบบเครือข่ายนิวรัล [20]

ส่วนการคำนวณทางคณิตศาสตร์สถิติที่ใช้งานทางด้านภาษาดูเพิ่มเติมจาก [7]

บทที่ 3
แบบจำลองภาษาทางสถิติ

แบบจำลองทางภาษา เป็นศิลปะในการ หาค่าความน่าจะเป็นการจัดเรียงลำดับของคำ มีความสำคัญและถูกใช้ประโยชน์อย่างขว้างขวางในด้านการรับรู้ การแยกแยะ, การค้นหา กลั่นกรอง ข้อมูลข่าวสาร, การแปลภาษาด้วยเครื่อง, ความสามารถในการจำลองความเข้าใจในภาษามนุษย์ด้วยเครื่อง ตลอดจนถึงการตรวจสอบคำสะกด และแก้ไขคำผิดด้วยคอมพิวเตอร์ ซึ่งเป็นส่วนของงานวิจัยนี้

3.1 Language modeling

จากผลในการสร้าง word graph จะพบว่ามีความเป็นไปได้หลายแบบในการสร้างประโยคด้วยกันและจากการนำค่าความน่าจะเป็นที่ได้จากผลลัพธ์ของ OCR มาใช้วิเคราะห์ความถูกต้องเพียงอย่างเดียวยังไม่เพียงพอ เช่น จากผลลัพธ์ที่ได้

ตารางที่ 3.1 แสดงประโยคที่ได้จาก words graph และคะแนนรวมของแต่ละประโยค

No.	ประโยค	คะแนนรวม
1	น้ำ เพื่อ ภาร เพาะ ปลูก	.88
2	น้ำ เพื่อ การ เพาะ ปลูก	.84
3	น้ำ เผื่อ ภาร เพาะ ปลูก	.86
4	น้ำ เพื่อ ภาร เผาะ ปลูก	.81
5	น้ำ เผื่อ การ เพาะ ปลูก	.80
6	น้ำ เพื่อ ภาร เพ วะ ปลูก	.75
7	น้ำ เผื่อ ภาร เผาะ ปลูก	.74
8	น้ำ เพื่อ การ เผาะ ปลูก	.70
9	น้ำ เพื่อ การ เพ วะ ปลูก	.70
10	น้ำ เผื่อ ภาร เพ วะ ปลูก	.66
11	น้ำ เพื่อ ภาร เผ วะ ปลูก	.63
12	น้ำ เผื่อ การ เผาะ ปลูก	.63
13	น้ำ เพื่อ การ เผ วะ ปลูก	.62
14	น้ำ เผื่อ ภาร เผ วะ ปลูก	.62
15	น้ำ เผื่อ การ เพ วะ ปลูก	.60

จะพบว่าประโยคที่ถูกต้องหมายเลข 2 มีค่าคะแนนรวม(คิดจากค่าคะแนนของแต่ละตัวอักษรที่ได้มาจาก OCR) ต่ำกว่าประโยคหมายเลข 1 ซึ่งการแก้ไขสามารถทำได้โดยใช้แบบจำลองภาษาเข้ามาช่วยในการหาประโยคที่ถูกต้อง

การสร้างแบบจำลองภาษา เป็นการพยามที่จะจับเอาลักษณะเฉพาะทางภาษาธรรมชาติโดยทำให้อยู่ในรูปแบบซึ่งสามารถนำ มาสร้างเป็นแบบจำลองได้

ภาษาธรรมชาติเป็นสิ่งที่มีความซับซ้อนโดยลักษณะของตัวมันอยู่ตามธรรมชาติ ซึ่งมันพัฒนาไปอย่างต่อเนื่องผ่านต่อไปจากรุ่นหนึ่งไปยังอีกรุ่นหนึ่งมาเป็นช่วงเวลานาน, ภาษามีการเปลี่ยนแปลง ไปตามยุคสมัย

3.2 สาเหตุในการใช้กรรมวิธีทางสถิติ

แม้ว่าในส่วนของงานด้าน NLP จะเป็นการประยุกต์ใช้ rule-based เป็นหลัก แต่ในด้าน LANGUAGE MODEL rule-based นั้น เป็นการยึดติดอยู่กับลักษณะเฉพาะทางกฎเกณฑ์มากเกินไปจนขาดความยืดหยุ่น อาจจะต้องใช้กฎ, ความรู้ (knowledge) เป็นจำนวนมากในการเพิ่มความถูกต้อง

กรรมวิธีหนึ่งซึ่งใช้ในการทำ LANGUAGE MODEL ซึ่งมีความแม่นยำและมีความยืดหยุ่นสูง คือการนำ กรรมวิธีทางสถิติเข้ามาใช้โดยในการสร้างแบบจำลองจะใช้ค่าสถิติโดยใช้การฝึกฝน (training) corpora ของ text ขนาดใหญ่

3.3 แบบจำลองทางสถิติและแบบจำลอง Knowledge-based

แบบจำลองทางสถิติมีข้อดีกว่าแบบจำลอง Knowledge-based ดังนี้

ค่าความน่าจะเป็นที่ได้จากแบบจำลอง มีประโยชน์มากกว่าคำตอบเพียง “ใช่”/“ไม่ใช่” ซึ่งค่าความน่าจะเป็นที่ได้นำไปใช้ประกอบกับเทคนิคอื่น ๆ ต่อไปได้มีประสิทธิภาพกว่า

การสร้างแบบจำลองทางสถิติสามารถพัฒนาและทำการฝึกฝนได้ด้วยโปรแกรมทางคอมพิวเตอร์ซึ่งสามารถทำได้รวดเร็วกว่าแบบจำลองทาง knowledge โดยเฉพาะเมื่อต้องการสร้างแบบจำลองสำหรับเรื่องในหัวข้อใหม่

ในทางปฏิบัติ, ฐานความรู้ส่วนมาก จะใช้เวลาในการคำนวณในขณะทำงานมากกว่า แบบจำลองทางสถิติ

ข้อเสียของแบบจำลองทางสถิติ

เนื่องจากแบบจำลองทางสถิติไม่มีการตรวจสอบความหมายของข้อความ ทำให้บางประโยคซึ่งอ่านดูไม่ถูกต้องตามสามัญสำนึก แต่อาจจะถูกลงความเห็นว่ายอมรับได้ จากโปรแกรม เนื่องจากความสัมพันธ์ของคำกลุ่มนั้นมีความความน่าจะเป็นสูง

แบบจำลองทางสถิติต้องการจำนวนของข้อมูลมาทำการฝึกขนาดใหญ่ และการส่งผ่านแบบจำลองระหว่างเรื่องต่างหัวข้อหรือต่างภาษากันไม่สามารถทำได้ เสมอไป
แบบจำลองทางสถิติมักจะไม่ได้ใช้ความรู้ทางภาษาที่มีอยู่ก่อนแล้วซึ่งบางส่วนก็น่าจะนำมาใช้เพิ่มความถูกต้องได้เป็นอย่างมาก
ในการนำเอาหลักการทาง สถิติเข้ามาช่วยในการแก้ปัญหา เพื่อหาค่าความน่าจะเป็นในการเกิดกลุ่มคำ a ซึ่งแวดล้อมด้วยคำ b หรือ p(a,b) ซึ่งใน corpus เราอาจไม่พบคำ a,b ในข้อมูลที่เรานำมาฝึกดังนั้นจึงต้องมีกรรมวิธีในการแก้ไขปัญหานี้ต่อไป

3.4 Information theory

การส่งข่าวสารข้อมูลซึ่งเรียกกันโดยทั่วไปว่าการติดต่อสื่อสาร ในปี 1948 Claude Shannon classic papers ได้เป็นผู้ให้กำเนิดทฤษฎีข้อมูลข่าวสารขึ้นซึ่งได้เป็นการเริ่มต้นของ การเข้ารหัสข้อมูลและการติดต่อสื่อสารในระบบ ดิจิตอล

ปัญหาหลักในการติดต่อสื่อสารคือการสร้างข้อมูลขึ้น ณ จุดหนึ่งให้มีความเหมือนหรือใกล้เคียงกันกับข้อมูล ที่ต้องการสื่อสาร ณ อีกจุดหนึ่ง [24]

ทฤษฎีสารสนเทศข้อมูลเป็นการใช้คณิตศาสตร์ ในปัญหาที่เกี่ยวข้องกับการเข้ารหัสข้อมูล, ส่งสัญญาณ, การถอดรหัสข้อมูล อย่างเป็นระบบ และมีระเบียบแบบแผน และเนื่องจากข้อความบนแผ่นกระดาษก็เป็นรูปแบบหนึ่งของการสื่อสาร, ทฤษฎีสารสนเทศน์ จึงมีส่วนเข้ามาเกี่ยวข้องในงานวิจัยนี้

3.5 เอนโทรปี (Entropy)

เราสามารถให้นิยามเกี่ยวกับปริมาณหรือจำนวนของ สารสนเทศโดย ใช้ทฤษฎีสารสนเทศน์ ตามทฤษฎี สารสนเทศน์ X_i ขึ้นอยู่กับความน่าจะเป็น ถ้าความน่าจะเป็น $P(x_i)$ มีค่าน้อย, เราจะได้ระดับดีกรีของสารสนเทศน์ที่สูง, เนื่องจากผลลัพธ์ที่ได้ออกมามีโอกาส เกิดขึ้นน้อย ในทางตรงกันข้ามหาก ค่าความน่าจะเป็นนั้นมีค่าสูง, ข้อมูลสารสนเทศน์ที่รับมานั้นจะมีขนาดเล็ก, เนื่องจากผลที่ได้ออกมานั้นสามารถประเมินได้ว่าดีมาก ดังนั้น เราสามารถกำหนดค่าสารสนเทศน์ได้ดังนี้

$$I(x_i) = \log \frac{1}{P(x_i)} \quad (3.1)$$

สาเหตุที่ใช้ค่าลอการิทึม เนื่องมาจาก มีข้อมูลของสองเหตุการณ์ที่เป็นอิสระต่อกัน (เป็นการ joint probability) สามารถนำมาบวกกันได้ และเมื่อใช้ ค่า ลอการิทึมฐานสอง หน่วยของสารสนเทศน์ก็จะเรียกว่าบิท (bit)

$$H(X) = E[I(X)] = \sum_{s} P(x_i)\log_2 \frac{1}{P(x_i)} = E[-\log_2 P(x)] \qquad (3.2)$$

entropy H(X) คือจำนวนเฉลี่ยของสารสนเทศน์ ซึ่งต้องการในการกำหนดสัญลักษณ์ชนิดใด ๆ ที่เกิดขึ้น แบบจำลองที่ดีจะให้ค่า entropy ที่ต่ำเมื่อนำไปทดสอบ, หรือกล่าวได้อีกอย่างหนึ่งว่า ข้อความที่นำมาทดสอบนั้นมีความเป็นไปได้สูง เมื่อมีค่า entropy ต่ำ

และค่า entropy นี้ยังเป็นค่าเฉลี่ยของความไม่แน่นอนของสัญลักษณ์ (symbol) สมมุติว่า แซมเปิลสเปส S คือจำนวนตัวอักษรขนาด $\|S\| = N$ ค่า entropy H(X) จะเข้าสู่ค่าสูงสุดเมื่อ p.f. มีค่าการกระจายที่สม่ำเสมอ (uniform distributaion) เช่น

$$P(x_i) = P(x_j) = \frac{1}{N} \quad \text{สำหรับ ทุก i และ j} \qquad (3.3)$$

สมการ 3.3 นี้สามารถแปลความหมายของ ความไม่แน่นอนซึ่งมีค่าสูงที่สุดเมื่อค่าความน่าจะเป็นของเหตุการณ์ทั้งหมดซึ่งสามารถเกิดขึ้นได้เท่า ๆ กัน ไม่มีเหตุการณ์ใดมีค่าความน่าจะเป็นสูงกว่าเหตุการณ์อื่น ๆ และยังสามารถพิสูจน์ได้ว่า ค่า entropy H(X) ไม่เป็นลบและจะมีค่าเป็นศูนย์ ถ้าเพียงค่าความน่าจะเป็น มีค่าเป็น หนึ่ง

$$H(X) \quad 0 \qquad (3.4)$$

branching factor ของภาษาเป็นการวัดระดับระดับความยากของระบบภาษา โดยใช้ความสัมพันธ์กับขนาดของจำนวนคำที่ใช้เป็นคำศัพท์ ที่จำเป็นต้องใช้ในการแยกแยะประโยคที่ได้มาจากคำนิยามของ entropy ที่กล่าวมาเราสามารถกำหนดค่า branching factor ได้โดยคำนวณจาก

$$PP(X) = 2^{H(X)} \qquad (3.5)$$

เราเรียก PP(X) ว่า perplexity ของ X , และค่า perplexity นี้ยังเป็นตัวที่เรามักใช้วัดคุณภาพของแบบจำลองทางภาษา โดยทั่วไปอีกด้วย[25]

เมื่อนำไปทดสอบกับภาษาที่ใช้จริง สำหรับแบบจำลอง ที่ให้ค่า perplexity ออกมาต่ำจะสามารถคาดเดาได้ว่า แบบจำลองนั้นน่าจะดี และค่า perplexity เองก็ยังเป็นส่วนกลับค่าเฉลี่ยทางเลขคณิตของความน่าจะเป็นของแต่ละคำในประโยคอีกด้วย

3.6 แบบจำลองภาษาทางสถิติ N-gram

เราสามารถคำนวณค่าความน่าจะเป็นของ P(W) ของคำ W ซึ่งแสดงให้เห็นว่า W มีความถี่เท่าไรในการเกิดคำ W เช่นเราอาจหาค่าความน่าจะเป็นของคำว่าไป P(ไป) = 0.01 , แต่เนื่องจากเรามีประโยคและคำเป็นจำนวนมากการคำนวณค่าความน่าจะเป็นจากคำ เพียงคำเดียวจึงไม่สมควรนัก ,เราจึงคำนวณจาก P (ไป เที่ยว ก้น) = 0 , ซึ่งไม่น่าจะมีประโยคที่เขียนเช่นนี้
P(W) สามารถเขียนใหม่ได้เป็น

$$\begin{aligned} P(W) &= P(w_1, w_2, ..., w_n) \\ &= P(w_1)P(w_2 \mid w_1)P(w_3 \mid w_1, w_2) \cdots P(w_n \mid w_1, w_2, ..., w_{n-1}) \\ &= \prod_{i=1}^{n} P(w_i \mid w_1, w_2, ..., w_{i-1}) \end{aligned} \tag{3.6}$$

ซึ่ง $P(w_i \mid w_1, w_2, ..., w_{i-1})$ คือความน่าจะเป็นของ w_i ที่สืบเนื่องมาจากลำดับของคำ $w_1, w_2, ..., w_{i-1}$ ซึ่งเกินขึ้นก่อนหน้า w_i ซึ่งจากสมการข้างบนนี้ w_i จะขึ้นอยู่กับคำทั้งหมดที่อยู่ก่อนหน้า w_i สำหรับคำศัพท์ขนาด v จะมี v^{i-1} history , ดังนั้น $P(w_i \mid w_1, w_2, ..., w_{i-1})$ จึงมีขนาด v^i ค่าที่นำมาใช้ในการคำนวณ ซึ่งในความเป็นจริงแล้วความน่าจะเป็นของ $P(w_i \mid w_1, w_2, ..., w_{i-1})$ ไม่สามารถคำนวณได้จากทุกคำก่อนหน้า w_i , และ $w_1, w_2, ..., w_{i-1}$ ที่เป็นประโยคต่อเนื่องยาว ๆ นั้นมีโอกาสเกิดขึ้นน้อยมาก หรือแทบจะไม่พบเลย ในทางปฏิบัติเราจึงแบ่งแยกกลุ่มคำออกเป็นช่วงละเท่า ๆ กันโดยกลุ่มคำสามารถแบ่งได้โดยใช้คำก่อนหน้า $w_1, w_2, ..., w_{i-1}$ เป็นจำนวนช่วงเท่า ๆ กัน ซึ่งจะได้ว่า $P(w_i \mid w_1, w_2, ..., w_{i-1})$ หาจากกลุ่มคำโดยแบ่งเป็นส่วนย่อย ๆ จำนวน i คำ เช่นถ้าเราใช้การแบ่งเป็นกลุ่มละสามคำ (trigram) เราจะหาความน่าจะเป็นของคำจาก $P(w_i \mid w_{i-2}, w_{i-1})$
นั่นคือเราคำนวณค่าความน่าจะเป็น N-gram ของคำ โดยใช้คำก่อนหน้านั้น N-1 คำ N-gram[25]

ในทางปฏิบัติเราสามารถหาค่าความน่าจะเป็นได้โดยการนับจำนวนเหตุการณ์ ที่เราสนใจใน corpus, ให้ $C(w_{i-2}w_{i-1}w_i)$ เป็นการนับจำนวน $w_{i-2}w_{i-1}w_i$ ที่เกิดขึ้นใน corpus ที่นำมาฝึก, เช่นเดียวกันกับ $C(w_{i-2}w_{i-1})$ ดังนั้นจึงหาได้ว่า

$$P(w_i \mid w_{i-2}) \quad \frac{C(w_{i-2}w_{i-1}w_i)}{C(w_{i-2}w_{i-1})} \tag{3.7}$$

แต่การการหาค่าความน่าจะเป็นด้วยวิธีการนี้ จะพบว่าจะเกิดปัญหาถ้าเกิด ไม่พบลำดับของคำบางคำในการฝึกจะทำให้หาค่าความน่าจะเป็นออกมาเท่ากับ 0 เช่น

ถ้าเราหาความน่าจะเป็นของ P (ปีใหม่|เกิด วัน) แต่ภายใน corpus ที่นำมาฝึกไม่มีประโยคนี้, ดังนั้น C(เกิด วัน ปีใหม่) จึงเท่ากับ 0, ในขณะที่ C(เกิด วัน) อาจจะนับได้ 20 เราจึงหาค่าความน่าจะเป็นของ P (ปีใหม่|เกิด วัน) = 0 ซึ่งเราสามารถแก้ไขปัญหานี้ ด้วยวิธีเทคนิคการทำ Smooting ดังที่จะกล่าวต่อไป

3.7 Smoothing

ในการสร้าง N-gram LANGUAGE MODEL ปัญหาที่สำคัญอย่างหนึ่งก็คือความจำกัดของข้อมูลที่นำมาใช้ ฝึกฝน นั่นคือไม่มีข้อมูลของคำที่ไม่ค่อยถูกใช้ หรือถูกใช้น้อยมาก

เราสามารถประมาณค่า maximum likelihood สำหรับค่าความน่าจะเป็นของเหตุการณ์ ε ซึ่งเกิดขึ้นเป็นจำนวน $C(\varepsilon)$ จากทั้งหมด R เป็น

$$P(\varepsilon) = \frac{C(\varepsilon)}{R} \tag{3.8}$$

และด้วยข้อมูลที่นำมาฝึกซึ่งมีอยู่จำกัด, ทำให้ค่าความน่าจะเป็นที่ได้จากสมการข้างบนนี้ สูงเกินความเป็นจริงสำหรับ events ที่ observed และ ต่ำเกินไปสำหรับเหตุการณ์ที่ unobserved สำหรับ N-gram LANGUAGE MODEL เราไม่สามารถหลีกเลี่ยงปัญหาที่จะพบศัพท์ที่ไม่พบในการฝึกได้

ยกตัวอย่างเช่นถ้ามีคำศัพท์ 32000 คำ สำหรับ trigram จะมี trigram ที่เป็นไปได้ทั้งหมดเท่ากับ 3.2768e+13 trigram

ถ้า corpus ที่นำมาฝึก มีขนาด 100 ล้านคำ ซึ่งเป็นเพียง ขนาด 0.00030517578125% ของ trigram ที่เกิดขึ้นได้เท่านั้นเอง

และสำหรับกรณีที่ไม่พบคำศัพท์ที่เกิดขึ้น จากสมการข้างบน จะได้ความน่าจะเป็นเท่ากับศูนย์ ซึ่งไม่น่าจะเกิดขึ้น

ดังนั้นจึงมีการเทคนิคที่เรียกว่าการทำ smoothing เพื่อไม่ให้เกิดการ bias ค่า maximum likelihood และเพื่อให้แน่ใจว่าไม่มีค่าความน่าจะเป็นเท่ากับศูนย์

smoothing ได้ถูกนำมาใช้ในการแก้ไข กรณีซึ่งเกิดค่าความน่าจะเป็นเท่ากับศูนย์ สำหรับในกรณีซึ่ง พบคำ ซึ่งไม่ปรากฏใน language model คำว่า smoothing มาจากกระบวนการของเทคนิค smoothing ซึ่งทำให้เกิดการกระจายค่าของ ความน่าจะเป็น มีความราบเรียบมากขึ้น โดย

ปรับค่าความน่าจะเป็นที่ต่ำเช่น ศูนย์ให้มีค่าสูงขึ้น ปรับค่าความน่าจะเป็นที่สูงให้ต่ำลง นอกจากการนำเทคนิค smoothing มาใช้จะช่วยป้องกันในการเกิดค่าความน่าจะเป็น เท่ากับศูนย์แล้ว เทคนิคนี้ยังช่วยให้ โมเดลมีความแม่นยำเพิ่มขึ้น [3]

จากตัวอย่าง เพื่อง่ายในการทำความเข้าใจ จะยกสมการ smoothing อย่างง่ายสำหรับกรณี bigram [29]

$$p(w_i \mid w_{i-1}) = \frac{1 + c(w_{i-1}w_i)}{\sum_{w_i} 1 + c(w_{i-1}w_i)} = \frac{1 + c(w_{i-1}w_i)}{|V| + \sum_{w_i} c(w_{i-1}w_i)} \tag{3.9}$$

V = จำนวนคำศัพท์ทั้งหมด

โดยที่คำใด ๆ ที่ไม่ปรากฏในพจนานุกรมที่ใช้สำหรับ แบบจำลองจะถูกมองเป็น unknown word

แต่ค่าที่ได้จากสมการข้างบนนี้ก็ยังให้ความแม่นยำของแบบจำลองได้ไม่ดีขึ้นมากนักจึงได้มีการปรับปรุงสมการต่อไปเป็น

$$P_{smooth}(w_i \mid w_{i-n+1}...w_{i-1}) = \begin{cases} \alpha(w_i \mid w_{i-n+1}...w_{i-1}) & if\ C(w_{i-n+1}...w_i) > 0 \\ \gamma(w_{i-n+1}...w_{i-1}) P_{smooth}(w_i \mid w_{i-n+2}...w_{i-1}) & if\ C(w_{i-n+1}...w_i) = 0 \end{cases} \tag{3.10}$$

จากสมการ ถ้า n-gram สามารถนับจาก corpus ที่ฝึกได้โดยไม่เป็นศูนย์ เราจะใช้สมการ $\alpha(w_i \mid w_{i-n+2}...w_{i-1})$ นอกนั้นเราจะทำการ backoff ไปยังลำดับของ n-gram ที่ต่ำกว่า โดยใช้สมการ $P_{smoōth}(w_i \mid w_{i-n+2}...w_{i-1})$ โดยที่เงื่อนไขของค่า scaling factor $\gamma(w_{i-n+1}...w_{i-1})$ จะเป็นตัวที่ทำให้ผลรวมของการกระจายทั้งหมดมีค่าเป็นหนึ่ง เราเรียกอัลกอริทึมในลักษณะ แบบนี้ว่าแบบจำลอง backoff

อัลกอริทึมอื่นสำหรับการทำ smoothing ได้แก่ linear interpolation สำหรับอันดับ n-gram ที่อันดับสูงกว่าหรือต่ำกว่า

$$P_{smooth}(w_i \mid w_{i-n+1}...w_{i-1}) = \lambda P_{ML}(w_i \mid w_{i-n+1}...w_{i-1}) + (1-\lambda) P_{smooth}(w_i \mid w_{i-1+2}...w_{i-1}) \tag{3.11}$$

โดยที่ λ คือค่า weight ของการ interpolation ซึ่งขึ้นอยู่กับ $w_{i-n+1}...w_{i-1}$ ดังนั้นเราจึงเรียกแบบจำลองแบบนี้ว่าแบบจำลอง interpolated

สิ่งที่แตกต่างกันระหว่างแบบจำลอง backoff และ interpolated คือสำหรับค่าความน่าจะเป็นของ n-grams ซึ่งสามารถพบได้ใน corpus (nonzero counts) แบบจำลอง interpolated จะใช้ข้อมูลจากการกระจายของ ลำดับ n-grams ที่ต่ำกว่ามาร่วมในการคำนวณด้วยส่วน backoff นั้นจะไม่ใช้

3.8 สัมประสิทธิ์การลดทอน (discounting)

การลดทอนค่า เป็นส่วนที่เพิ่มเติมในหลักการของการแก้ไข การ bias ของเหตุการณ์ที่สังเกต ของการประมาณค่าโดยใช้ maximum likelihood probability estimates เหตุการณ์ที่นับได้จะถูกปรับลดค่าโดยคูณด้วยค่าสัมประสิทธิ์การลดทอน $d_{C(\varepsilon)}$ ซึ่ง 0 <= $d_{C(\varepsilon)}$ <=1

สำหรับทุก $C(\varepsilon)$ 1 ดังนั้นค่าลดทอนแบบง่ายในการนับคือ

$$C^*(\varepsilon) = d_{C(\varepsilon)} C(\varepsilon) \tag{3.12}$$

ให้ r คือจำนวนครั้งที่นับได้ และ r* คือจำนวนที่ปรับแก้แล้วเขียนสมการใหม่ได้เป็น

$$r^* = r d_r \tag{3.13}$$

สำหรับการกำหนดค่าสัมประสิทธิ์การลดทอน หรือค่า d_r มีอยู่หลายวิธีด้วยกันได้แก่

Good-Turing discounting

ถ้าเรากำหนดค่า n_r เป็นจำนวนหรือเหตุการณ์ ซึ่งเกิดขึ้นเป็นจำนวน r ครั้ง, รูปแบบสำหรับการลดทอนค่าแบบ Good-turing สามารถเขียนได้ดังนี้

$$d_r = \frac{\frac{(r+1)n_{r+1}}{rn_r} - \frac{(k+1)n_{k+1}}{n_1}}{1 - \frac{(k+1)n_{k+1}}{n_1}} \tag{3.14}$$

สำหรับ r < k (ซึ่งโดยทั่วไปค่า k 7) และ d_r = 1 สำหรับ r ที่สูงกว่า k [26],

Linear discounting

ในการหาค่าสัมประสิทธิ์การลดทอนแบบ linear [27]

$$d_r = 1 - \frac{n_1}{R} \tag{3.15}$$

โดยที่ R คือจำนวนคำทั้งหมดของข้อมูลที่นำมาฝึก

Absolute discounting

สัมประสิทธิ์การลดทอน หาได้จากการลบจำนวนที่นับได้ออกด้วย b แล้วหารด้วย จำนวนที่นับได้นั้นอีกครั้งหนึ่ง

$$d_r = \frac{r - b}{r} \tag{3.16}$$

ดังแสดงไว้ใน [27] ซึ่งการสมมุติค่า $b = \frac{n_1}{n_1 + 2n_2}$ ให้ค่าออกมาดีที่สุด

Witten-Bell discounting

การหาค่าสัมประสิทธิ์การลดทอนของ Witten-Bell ในที่นี้หมายถึง แบบ C ใน [17], และได้มีผู้นำมาใช้เป็นครั้งแรก ใน [14] สำหรับสัดส่วนในการลดทอนค่าไม่ขึ้นอยู่กับจำนวนที่นับได้ แต่จะขึ้นอยู่กับ t โดย t เป็นจำนวนเหตุการณ์ที่แตกต่างกันเช่นสำหรับ bi-gram "A B", t เป็นจำนวน ของ bi-gram . "A *" ทั้งหมดในแบบจำลอง

$$d_r(t) = \frac{R}{R + t} \tag{3.17}$$

เมื่อนำเอา เทคนิคการลดทอนค่ารวมเข้ากับเทคนิคการทำ smoothing backoff

จะได้สมการเป็น

$$P_{kate}(w_i \mid w_{i-1}) = \begin{cases} C(w_{i-1}w_i)/C(w_{i-1}) & if\ r > k \\ d_r C(w_{i-1}w_i)/C(w_{i-1}) & if\ k \quad r \quad 0 \\ \alpha(w_{i-1})P(w_i) & if\ r = 0 \end{cases} \tag{3.18}$$

โดยที่

$$d_r = \frac{\frac{r^*}{r} - \frac{(k+1)n_{k+1}}{n_1}}{1 - \frac{(k+1)n_{k+1}}{n_1}} \text{ และ } \alpha(w_{i-1}) = \frac{1 - \sum_{w_i:r>0} P_{Katz}(w_i \mid w_{i-1})}{1 - \sum_{w_i:r>0} P(w_i)} \quad (3.19)$$

โดยสมการข้างบนนี้เรียกว่าแบบจำลองของ Katz

3.9 เทคนิคการลดขนาดของแบบจำลองภาษา

3.9.1 Count-Cutoffs

เป็นเทคนิคที่ง่ายและธรรมดาที่สุด โดยสมมุติว่าค่าความน่าจะเป็นของคำ z ที่ตามด้วยคำ x และ y หาได้จาก

$$P(z \mid xy) = \begin{cases} \frac{C(xyz) - D(C(xyz))}{C(xy)} & if\ C(xyz) > 0 \\ \alpha(xy)P(z \mid y) & otherwise \end{cases} \quad (3.20)$$

โดยที่ C(xyz) หมายถึงการนับจำนวน xyz ที่เกิดขึ้นทั้งหมดในข้อมูลที่นำมาฝึก ฟังชันก์ α เป็นค่าคงที่สำหรับ normalization ส่วนฟังก์ชัน D(C(xyz)) คือ discount ฟังชันก์

เทคนิคการ cutoff, สมมุติว่าถ้า cutoff ที่ 3 , ถ้า C(xyz) <= 3 จะไม่นำมาใส่ใน แบบจำลองทางภาษา ถึงแม้ C(z) จะมากกว่า 3 ก็ตาม สำหรับเทคนิคนี้จะมีผลมากในแบบจำลองที่มีขนาดเล็ก, จะพบว่าค่า perplexity เพิ่มขึ้นอย่างเห็นได้ชัดเมื่อเพิ่มค่า cutoff

3.9.2 N-gram Pruning

เมื่อลำดับแบบจำลอง n-gram สูงขึ้น, ขนาดของแบบจำลองก็จะจะมีขนาดใหญ่โตขึ้นด้วย การ pruning เพื่อลดขนาดของ n-gram จึงมีความจำเป็นยิ่ง โดยแบบจำลองที่ prune แล้วจะต้องให้ความแตกต่างของผลลัพธ์ระหว่าง แบบจำลองที่ prune แล้วและของเดิม แตกต่างกันน้อยที่สุด

สำหรับตัวอย่างการ prune n-gram ดูได้จาก [28]

3.9.3 Class n-gram

นอกจากนี้ยังได้มีการใช้เทคนิคการแบ่งแยกประเภทของคำเพื่อเพิ่มเติมความถูกต้องของแบบจำลอง [23] โดยจัดคำแยกเป็นกลุ่มเช่น

วัน เดือน ปี, ...

ส้ม แตง ฝรั่ง ขนุน ทุเรียน, ...

น้ำเงิน ฟ้า ส้ม แดง เขียว เหลือง, ...

เมื่อเจอคำว่า สี กลุ่ม คำที่มักต่อท้ายด้วย "น้ำ เงิน แดง ..."

3.10 Winnow

ตัว winnow เองนั้นเป็นอัลกอริทึมในการเรียนรู้แบบหนึ่ง [29] ซึ่งนำมาใช้ในการแก้คำผิดแบบ context-sensitive

โดยในการแก้ไขคำผิดด้วย Winnow ก่อนอื่น จะทำการรวบรวม คำที่มีรูปร่าง คล้ายกันหรือคำที่มักจะผิดหรือเพี้ยน (สำหรับในกรณีที่ความผิดพลาดเกิดจากการรู้จำ), คำที่มักพิมพ์ผิด (เกิดจากขั้นตอนการพิมพ์) เป็นกลุ่มคำที่จะนำมาเรียนรู้

เช่น "ฝน" อาจจะเป็นคำอื่น ๆ ได้คือ "ฝัน ฟัน พัน พ้น พ่น "

รวบรวมไว้เป็นกลุ่มเข้าด้วยกัน เพื่อจะทำการแยกแยะรายละเอียดในขั้นตอนต่อไปว่า คำเหล่านี้มีความสัมพันธ์กับคำรอบข้างอย่างไร เช่น

คำว่า ฝัน ภายในคำสิบคำรอบ ๆ จะพบคำว่า "นอน กลางวัน ดี ร้าย เลขเด็ด หวย งู เห็น"

คำว่า พ้น ภายในคำสิบคำรอบ ๆ จะพบคำว่า "ภัย รอด อันตราย เกือบ ตาย ... "

...

ซึ่งคำที่อยู่รอบ ๆ เหล่านี้ จะเรียกว่า features สำหรับ feature นี้ นอกจากจะเป็นคำที่อยู่รอบ ๆ แล้วยังจะอาจรวมถึง ชนิดของคำด้วย เช่น คำนาม คำกริยา คำวิเศษ และอาจรวมถึงหมวดหมู่ของคำด้วยก็ได้ เช่น กิน ภายในสิบคำ รอบ ๆ จะเจอ คำที่อยู่ในหมวดหมู่ของของที่กินได้

3.9.1 Weightted-Majority: combining experts

เป็นอัลกอริทึมซึ่งมีหน้าที่หลักในการตั้งข้อสมมุตในการกำหนดค่าคะแนนของ features ที่กำหนดให้มาซึ่งอาจจะมี feature เพียงสอง feature หรือ อาจมีเป็นจำนวนมากก็ได้ การที่จะมีมากเพียงใดขึ้นอยู่กับว่า จำนวน feature นั้นเพียงพอในการสร้างการทำนายที่แม่นยำหรือไม่ โดยมีหลักสำคัญคือจะมีการตั้งค่าคะแนนขึ้นมา สำหรับทุก ๆ โหนดของ feature แล้วทำการปรับค่าในแต่ละโหนด ผ่านการ ฝึกฝน

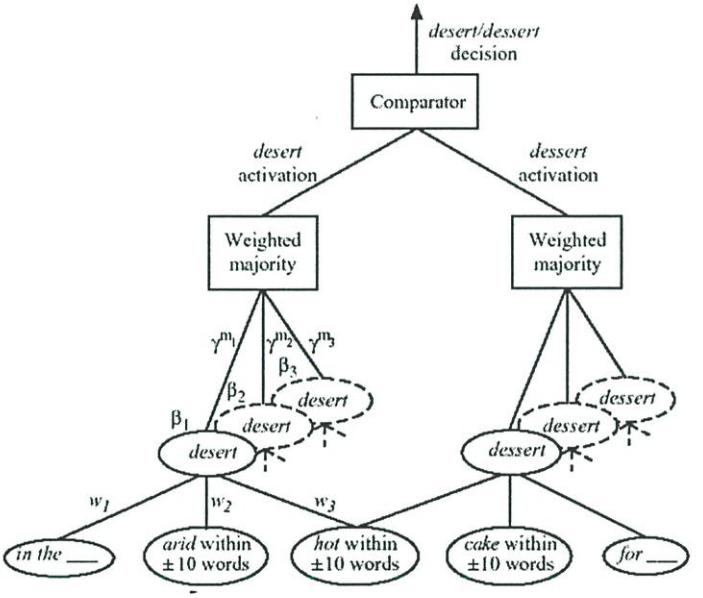
3.9.2 Winnow : combining specialist

Winnow [1][18] (multiplicative weight updating algorithm) เป็นอัลกอริทึมในการที่จะรวบรวมและทำการเลือกค่าคะแนนน้ำหนักต่าง ๆ ที่ได้จาก expert และในการฝึกฝนจะทำการปรับค่าคะแนนโดยเงื่อนไขว่า จะปรับค่าคะแนนเฉพาะ ในกรณีที่ผลการฝึกให้ผลลัพธ์ออกมาผิดพลาดเท่านั้นจึงจะ มีการ ปรับค่าคะแนน โดยข้อมูลที่จะนำมาฝึกฝนนั้นจะเลือกโดยเฉพาะเจาะจงเพื่อ

ฝึก คำ คำหนึ่งโดยเฉพาะ เช่น เมื่อฝึกคำว่า ฝัน ก็จะนำประโยคที่มีคำว่าฝัน เข้ามา ฝึกเมื่อผลการฝึกคำว่าฝันในประโยคให้ผลลัพธ์ออกมาผิดพลาด ก็จะทำการปรับค่าคะแนน สำหรับ feature ที่ปรากฏในประโยคที่ผิดพลาดเท่านั้น

จากจำนวนคำภายในกลุ่มคำที่เกิดขึ้นจะสังเกตได้ว่าคำที่เป็นไปได้นั้นมีเป็นจำนวนมาก ในการทำงานจริงหากใช้ทุกคำที่เป็นไปได้มาตรวจสอบด้วย winnow ทุกคำที่เป็นไปได้จะทำให้เกิดการทำงานที่ไม่จำเป็น ดังนั้นจึงนำเอา 3-gram เข้ามาทำการตรวจสอบแล้วเลือกเฉพาะคำ ที่น่าจะเป็นไปได้ แล้วทำการตรวจสอบด้วย winnow ต่อไป

สำหรับในงานวิจัยของ [18] ได้นำเอาแบบจำลอง n-gram ผนวกเข้าไปเป็นส่วนหนึ่งของ ตัว winnow ด้วยแล้วเรียกว่า complete winspell algorithm ซึ่งตัวแบบจำลอง n-gram จะเป็นเสมือน layer หนึ่งภายใน winspell algorithm



รูปที่ 3.1 แสดงตัวอย่างเครือข่ายของ Winspell network

แม้ว่า winnow และ n-gram จะมีความคล้ายกันอยู่มากสำหรับการดึงเอาความรู้จากฐานข้อมูลโดยลักษณะที่เป็น linear แต่ความแตกต่างที่สำคัญของ แบบจำลอง n-gram และ winspell คือ ตัว winnow นั้นใช้อัลกอริทึมในการเรียนรู้เพื่อการปรับค่าคะแนน เพื่อช่วยให้ผลลัพธ์ที่ได้มีความถูกต้องเพิ่มมากขึ้น และข้อดีของ winnow อีกอย่างหนึ่งคือเมื่อ ถึงจะเปลี่ยน domain ที่แตกต่างไปจากการฝึก winnow ก็ยังให้ผลที่ดีกว่า n-gram เนื่องจาก ตัว winnow นั้นไม่ได้กำหนดตำแหน่งของคำรอบ ๆ ให้อยู่ในตำแหน่งที่ตายตัว เหมือนกับ n-gram และตัว winnow ต้องการการปรับค่าด้วยมนุษย์ ซึ่งการจะสร้าง winnow ที่สมบูรณ์นั้นยากกว่าแบบจำลอง n-gram

บทที่ 4
กรรมวิธีการตัดแบ่งคำด้วยโทคเกนพาสซิ่ง

4.1 การตรวจสอบคำสะกด

การตรวจสอบคำสะกดจะนำอักษรของคำเพิ่มเข้ามาในโทคเกนทีละตัวแล้วนำไปเปรียบเทียบกับคำในพจนานุกรม เมื่อพบว่าเป็นคำหรือมีโอกาสที่จะเกิดเป็นคำ โทคเกนนั้นก็จะยังเก็บเอาไว้ และถ้าโทคเกนใดตรวจไม่พบคำหรือไม่มีโอกาสที่จะเกิดเป็นคำแล้วโทคเกนนั้นก็จะถูกทิ้งไป ดังตัวอย่างในตารางที่ 4.1

ตารางที่ 4.1 เปรียบเทียบจำนวนคำที่ทำการตรวจสอบระหว่างการมีขอบเขตของคำและไม่มีขอบเขตของคำ

<table>
<tr><th>มีเว้นช่องว่างระหว่างคำ</th><th>ไม่มีเว้นช่องว่างระหว่างคำ</th></tr>
<tr><td>เขียน ติด กัน</td><td>เขียนติดกัน</td></tr>
<tr><td rowspan="3">มีสามคำที่ต้องทำการตรวจสอบนั่นคือ
เขียน, ติด, กัน</td><td>คำแรกที่จะตรวจสอบ
เข, เขี, เขีย, เขียน, เขียนต, เขียนติ, เขียนติด, เขียนติดก, เขียนติดกั, เขียนติดกัน</td></tr>
<tr><td>คำที่สองที่เป็นไปได้
(เริ่มจาก เขียน) ติ, ติด</td></tr>
<tr><td>คำที่สามที่เป็นไปได้
(เริ่มจาก ติ) ดก
(เริ่มจาก ติด) กัน</td></tr>
<tr><td></td><td>ประโยคที่เป็นไปได้
เขียน ติด กัน</td></tr>
</table>

สำหรับในงานวิจัยนี้, ผลลัพธ์จาก OCR จะเป็นกลุ่มของตัวอักษรออกมา 5 ลำดับตัวอักษรซึ่งมีค่าความน่าจะเป็นของแต่ละตัวสูงกว่า 70% ขึ้นไป ด้วยเหตุนี้กรรมวิธีในการตรวจสอบคำสะกดจึงใช้เวลามากขึ้นเนื่องจากแต่ละตัวอักษรจากตาราง 4.1 ถูกแทนที่ด้วยชุดของตัวอักษรดังแสดงให้เห็นในตารางที่ 4.2

ตารางที่ 4.2 อักขระ 5 ตัวที่เป็นไปได้และค่าความน่าจะเป็นจากการรู้จำของแต่ละตัวอักษร

0	1	2	3	4	5	6	7
เ	ข	อิ	ย	น	ต	อิ	ด
. .87	.88	.86	.88	.85	.89	.86	.87
โ	ช	อึ	บ	บ	ด	อี	ต
.78	.77	.79	.78	.79	.78	.78	.79
ไ	ซ	อื	น	ม	ค	อื	ค
.77	.75	.77	.76	.79	.76	.75	.78
ใ	บ	อี	ม	ข	ศ	อี	ศ
.75	.74	.75	.73	.74	.73	.73	.75
ร	ป	อ์	ษ	ษ	ศ	อ์	ศ
.74	.72	.72	.70	.70	.73	.72	.73

จากตารางที่ 4.2 แสดงผลลัพธ์ที่ได้จาก OCR อ่านประโยค “เขียนติด” ในที่นี้สระ อิ ในตำแหน่งที่ 2 นั้นมีผลลัพธ์จากการรู้จำได้คะแนนสูงเป็นอันดับแรก และถ้าคิดเฉพาะอักษรซึ่งมีคะแนนสูงสุดแล้วจะได้เป็นประโยคว่า “เขียนติด” ซึ่งเป็นกรณีของอักษรผิดไม่ถูกต้อง จากประโยคที่ถูกต้องแล้วน่าจะเป็นสระ อี และหากใช้ตัวแก้ไขด้วย Microsoft word 2000 แล้วคำที่แนะนำคือ “เขิน เชิง เข็บ เข” ซึ่งไม่สามารถแก้ไขให้ถูกต้องได้เลย

ในการนำโทคเกนพาสซิ่งอัลกอริทึมมาใช้ โดยอัลกอริทึมจะทำการค้นหาประโยคทั้งหมดที่เป็นไปได้จากตัวอักษรทั้ง 5 ตัว ซึ่งอัลกอริทึมนี้จะทำการค้นหาอักขระที่ถูกต้องจากทั้ง 5 ตัวอักขระและหาขอบเขตคำที่ถูกต้อง โดยที่โทคเกนนั้นจะถูกสร้างและส่งผ่านไปยังอักขระแต่ละตัวดังแสดงในตารางที่ 4.1 โทคเกนที่เก็บไว้นั้นจะประกอบด้วยคำ(หรือส่วนเริ่มต้นของคำ)ที่พบในพจนานุกรม

สำหรับโทคเกนที่ไม่มีคำปรากฏในพจนานุกรม จะถูกทิ้งไป และเมื่อโทคเกนนั้นได้สิ้นสุดการตรวจสอบคำสะกดและค้นหาคำในพจนานุกรมแล้วอาจสร้างโทคเกนขึ้นใหม่ได้สองโทคเกนในหนึ่งอักขระที่เพิ่มเข้ามา โดยอันหนึ่ง เป็นของโทคเกนซึ่งสิ้นสุดลงด้วยขอบเขตของคำ(ซึ่งในที่นี้จะแสดง

โดยปิดท้ายด้วยเครื่องหมายขีดล่าง _) และอีกอันหนึ่งเป็นโทคเกนที่คำยังไม่สิ้นสุด เช่น จากรูปที่ 4.1 เข,เข_ ซึ่งคำว่า เข_ นั้นสิ้นสุดเกิดเป็นคำที่สมบูรณ์แล้ว แต่ เข อาจจะยังสามารถรวม กับตัวอักษรตัวอื่นต่อไปเป็นคำอื่นซึ่งมีอยู่ในพจนานุกรมได้อีกเช่น เขียน เข่น เข้ม เข็ม ฯลฯ

จากตาราง 4.1 สามารถเขียนได้ใหม่เป็นตาราง 4.3 จากตารางจะพบว่าการส่งโทคเกน ไปทั้งหมดจนสิ้นสุดประโยคนั้น ทำให้ต้องส่ง โทคเกน ขนาดใหญ่ขึ้นเมื่อจำนวนคำที่ต่อกันนั้นมีมากขึ้น จึงได้ทำการลดขนาด โทคเกนที่ทำการส่งลงโดยเมื่อตรวจสอบ โทคเกนที่สมบูรณ์เป็นคำแล้วจะไม่ทำการส่งต่อ โทคเก้นที่เป็นคำสมบูรณ์แล้วต่อไปอีก (โทคเกนที่เติมท้ายด้วยขีดล่าง) แต่จะนำไปเก็บไว้เพื่อนำมาทำการ passing ในระดับคำต่อไปอีกครั้งหนึ่งเพื่อสร้างเป็นประโยคต่อไป ดังในตาราง 4.4 จะเห็นได้ว่า โทคเกน ที่ทำการส่งต่อนั้นขนาดลดลงมาก และในขั้นตอนสุดท้ายเมื่อทำการส่งต่อ โทคเกน ในระดับคำเพื่อสร้างเป็นประโยคก็ยังเหลือจำนวนประโยคอยู่เท่าเดิม

4.2 การคำนวณคะแนนของคำ (โทคเกน)

ในขณะที่ทำการสร้างโทคเกน คำที่เป็นคำที่สมบูรณ์ในการจัดเก็บคำจะนำค่าคะแนนของตัวอักษรที่ได้นำมาคำนวนเป็นค่าคะแนนของคำด้วย โดย

$$P(W \mid S) = n \prod_{i}^{n} P(c_i) \qquad (4.1)$$

โดยที่

c = อักขระ

n = จำนวนตัวอักษรของคำ

S = สายอักขระ $c_1c_2...c_n$ โดยที่ c_n คืออักขระตัวสุดท้ายของคำ

เช่นคำว่าเขียน สามารถคำนวณ ค่าคะแนนได้เท่ากับ

.87x.88x.86x.88x.85 = .4089502

4.3 การปรับค่าคะแนนของคำ

จากตัวอย่างพบว่าค่าคะแนนจะน้อยลงเมื่อคำยาวขึ้น ดังนั้นจึงทำการปรับค่าคะแนนโดยนำค่าที่มีคะแนนสูงสุด ของกลุ่มตัวอักษรมาหาร คะแนนของตัวอักษรภายในกลุ่มทำให้ได้ตารางดังตารางที่ 4.5 เพื่อให้คำที่มีค่าเฉลี่ยเลขคณิตมีคะแนนเข้าใกล้ 1

นอกจากค่าคะแนนของแต่ละคำแล้วโทคเกนยังประกอบด้วยตำแหน่งเริ่มต้นและตำแหน่งสิ้นสุดของคำนั้น ๆ ลงไปด้วยเพื่อใช้ในการส่งต่อของโทคเกนในระดับประโยคต่อไป

ตารางที่ 4.3 ตัวอย่างของโทคเกนพาสซิ่ง (แสดงเฉพาะโทคเกนที่เป็นคำสมบูรณ์) ในกรณีซึ่งไม่มีความผิดพลาดในกรณีอักษรขาดหรือเกิน

0	1	2	3	4	5	6	7
เ	ข	อิ	ย	น	ต	อิ	ด
เ	เข, เข_, โข_, ไข_,	เขิ, เขี	เขีย, เชีย, เชีย_,เชีย, เปีย_, เบีย	เขียน_, เชียน _, เบียน_	เขียน_ต, เชียน_ต, เบียน_ต, เชียบ_ต ,เชียบ_ต ,เชียม_ต,	เขียน_ติ,เชียน_ติ, เบียน_ติ,เชียบ_ติ, เชียบ_ติ,เชียม_ติ, เชียน_ติ_,เชียน_ติ _,เบียน_ติ_,เชียบ_ ติ_,เชียบ_ติ_,เชียม _ติ_,เชียน_คิ,เชียน _คิ,เบียน_คิ,เชียบ_ คิ,เชียบ_คิ,เชียม_คิ	เขียน_ติด_,เชียน_ ติด_,เบียน_ติด_, เชียบ_ติด_,เชียบ_ ติด_,เชียม_ติด_, เชียน_ดีด_,เชียน_ ดีด_,เบียน_ดีด_, เชียบ_ดีด_,เชียบ_ ดีด_,เชียม_ดีด_, เชียน_คิด_,เชียน_ คิด_,เบียน_คิด_, เชียบ_คิด_,เชียบ_ คิด_,เชียม_คิด_
โ	ข	อื	บ	บ	ด	อี	ต
โ	เข, ไข_		เขิบ_, เปิบ_	เชียบ_, เชียบ_	เขียน_ด, เชียน_ด, เบียน_ด, เชียบ_ด, ,เชียบ_ด, ,เชียม_ด	เขียน_ดี,เชียน_ดี, เบียน_ดี,เชียบ_ดี, เชียบ_ดี,เชียม_ดี, เชียน_ดี_,เชียน_ดี _,เบียน_ดี_,เชียบ_ ดี_,เชียบ_ดี_,เชียม _ดี_,เชียน_ดี,เชียน _ดี,เบียน_ดี,เชียบ_ ดี,เชียบ_ดี,เชียม_ดี	เขียน_ดีด_,เชียน_ ดีด_,เบียน_ดีด_, เชียบ_ดีด_,เชียบ_ ดีด_,เชียม_ดีด_
ไ	ข	อื	น	ม	ค	อี	ค
ไ	เข, เข_, โข, ไข		เขิน_	เชียม_	นค_, เชียน_ค, เชียน_ค, เบียน_ค, เชียบ_ค, ,เชียบ_ค, ,เชียม_ค	เขียน_ดี_,เชียน_ดี _,เบียน_ดี_,เชียบ_ ดี_,เชียบ_ดี_,เชียม _ดี_,เชียน_ดี,เชียน _ดี,เบียน_ดี,เชียบ_ ดี,เชียบ_ดี,เชียม_ดี	
ใ	บ	อี	ม	ข	ศ	อี	ศ
ใ	โบ_, ใบ_, รบ_ ,เบ	เขี, เชี, เปี, เชี, เบี				เขียน_ดี, เชียน_ดี, เบียน_ดี, เชียบ_ดี, ,เชียบ_ดี, ,เชียม_ดี	
ร	ป	อ์	ษ	ษ	ศ	อ์	ศ
ร	โป_, ไป_, เป						

ตารางที่ 4.4 ตัวอย่างของโทคเกนพาสซึ่งโดยส่งเพียงโทคเกนเป็นคำที่ไม่มีขอบเขตของคำ (โทคเกนที่เป็นคำสมบูรณ์จะนำไปเก็บไว้ ซึ่งถือเป็นการพรุนนิ่งแบบหนึ่ง)

0	1	2	3	4	5	6	7
เ	ข	อิ	ย	น	ต	อิ	ด
เ	เข, เข_, โข_, ไข_,	เขิ, เขี	เขีย, เชีย, เชีย_,เชีย, เปีย_, เบีย	เขียน_, เชียน _, เบียน_	ต,	ติ,ติ_	ติด,คิด,ดีด
โ	ช	อึ	บ	บ	ด	อี	ต
โ	เช, ไช_		เขิบ_, เปิบ_	เชียบ_, เตียบ_	ด,	ตี,ตี_	ตืด
ไ	ช	อื	น	ม	ค	อื	ค
ไ	เช, เช_, โช, ไช		เขิน_	เขียม_	ค,	ตื	
ใ	บ	อี	ม	ข	ศ	อื	ศ
ใ	โบ_, ใบ_, รบ_ , เบ	เขี, เชี, เปี, เชี, เบี			ศ	ตื,	
ร	ป	อ์	ษ	ษ	ศ	อ์	ศ
ร	โป_, ไป_, เป				ศ		

ตารางที่ 4.5 ตัวอย่างคะแนนของโทคเกนในระดับตัวอักษรที่ปรับค่าใหม่

0	1	2	3	4	5	6	7
เ 1	ข 1	อิ 1	ย 1	น 1	ต 1	อิ 1	ด 1
โ 0.90	ช 0.88	อื 0.92	บ 0.89	บ 0.93	ด 0.88	อี 0.91	ต 0.91
ไ 0.89	ซ 0.85	อื 0.90	น 0.86	ม 0.93	ค 0.85	อื 0.87	ค 0.90
ใ 0.86	บ 0.84	อี 0.87	ม 0.83	ข 0.87	ศ 0.82	อื 0.85	ศ 0.86
ร 0.85	ป 0.82	อ์ 0.84	ษ 0.80	ษ 0.82	ศ 0.82	อ์ 0.84	ศ 0.84

พิจารณา ตัวอย่างที่ซับซ้อนมาขึ้น จากตัวอย่างประโยค

"น้ำเพื่อการเพาะปลูก"

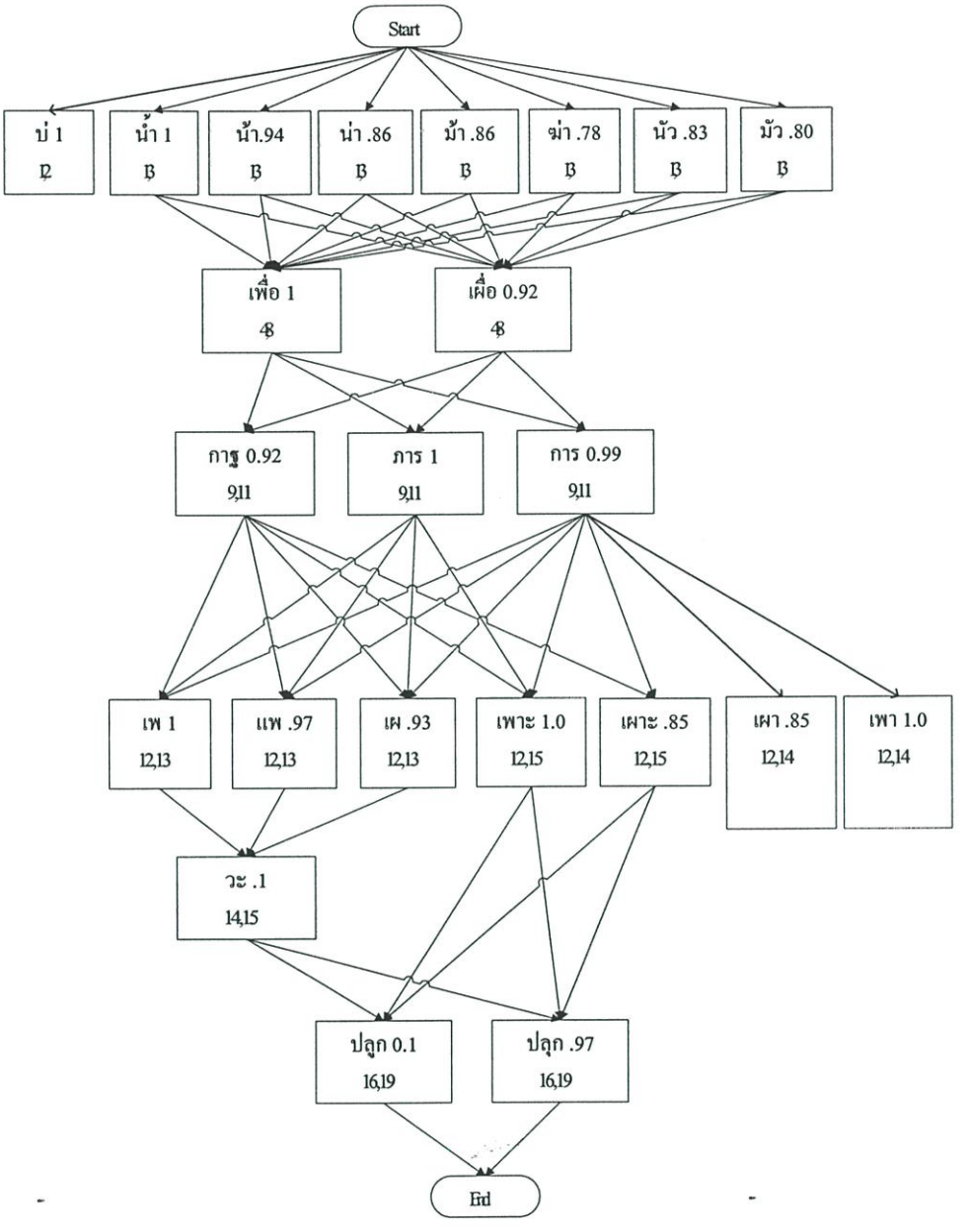
จากประโยคตัวอย่างนำมาสร้างและส่งต่อโทคเกนในระดับตัวอักษร จนได้โทคเกนของคำที่เกิดขึ้นทั้งหมด เมื่อนำมาเชื่อมต่อกันจะพบว่ามีบางคำซึ่งทำให้เกิดความไม่ต่อเนื่องดังรูปที่ 4.1

คำว่า "บ่ เผา เพา" เป็นคำที่ไม่ต่อเนื่องในรูปที่ 4.1 สามารถตัดทิ้งไปได้ เมื่อทำการตัดคำไปแล้วเราจะได้ดังรูปที่ 4.2

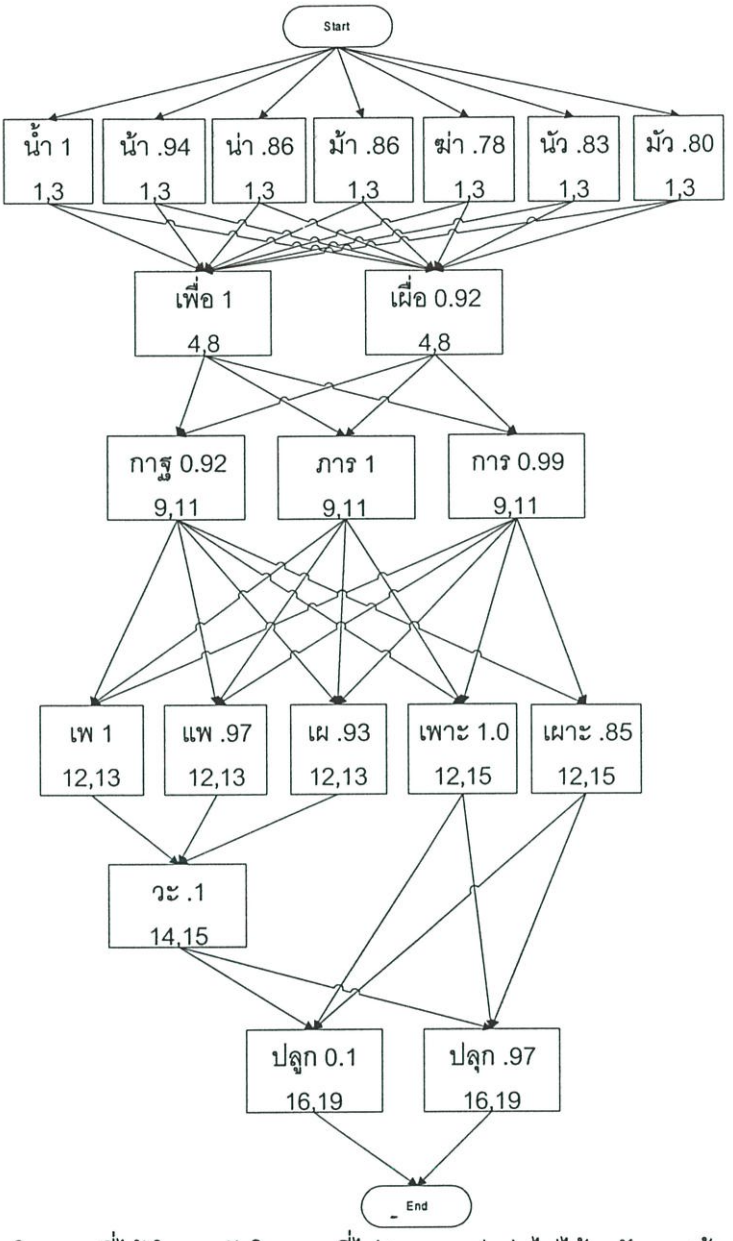
การส่งต่อโทคเกนระดับคำ

จากรูปที่ 4.2 เมื่อเราเริ่มต้นใหม่ในการส่งต่อโทคเกนคำ(โหนด) ทั้งหมดต่อไปก็จะทำให้เราได้ประโยคทั้งหมดที่สามารถเป็นไปได้ แล้วจึงนำประโยคที่ได้นี้ ไปหาว่าประโยคใดเป็นประโยคที่ถูกต้องโดยใช้ แบบจำลองทางภาษาต่อไป โดยแต่ละประโยคจะนำคะแนนของคำในประโยค มาหาค่าเฉลี่ยเลขคณิต ของแต่ละประโยคเพื่อใช้ในการลดจำนวนประโยคในขั้นตอนต่อไป

ซึ่งจากรูปที่ 4.2 นี้เราสามารถสร้างประโยคได้ถึง 420 ประโยค ซึ่งจากการทดลอง พบว่าบางประโยคที่เขียนต่อเนื่องกันไปโดยไม่มีเว้นวรรค ซึ่งประกอบด้วยตัวอักษรเพียง 40-50 ตัวอักษรนั้น สามารถทำให้เกิดประโยคได้เป็นล้านประโยค ดังนั้นเราจึงต้องหาหนทางในการลดประโยคนั้นลง เนื่องจากประโยคจำนวนมหาศาลนี้ไม่สะดวกในการเก็บลงหน่วยความจำเพื่อตรวจสอบหาความถูกต้องในขั้นต่อไป

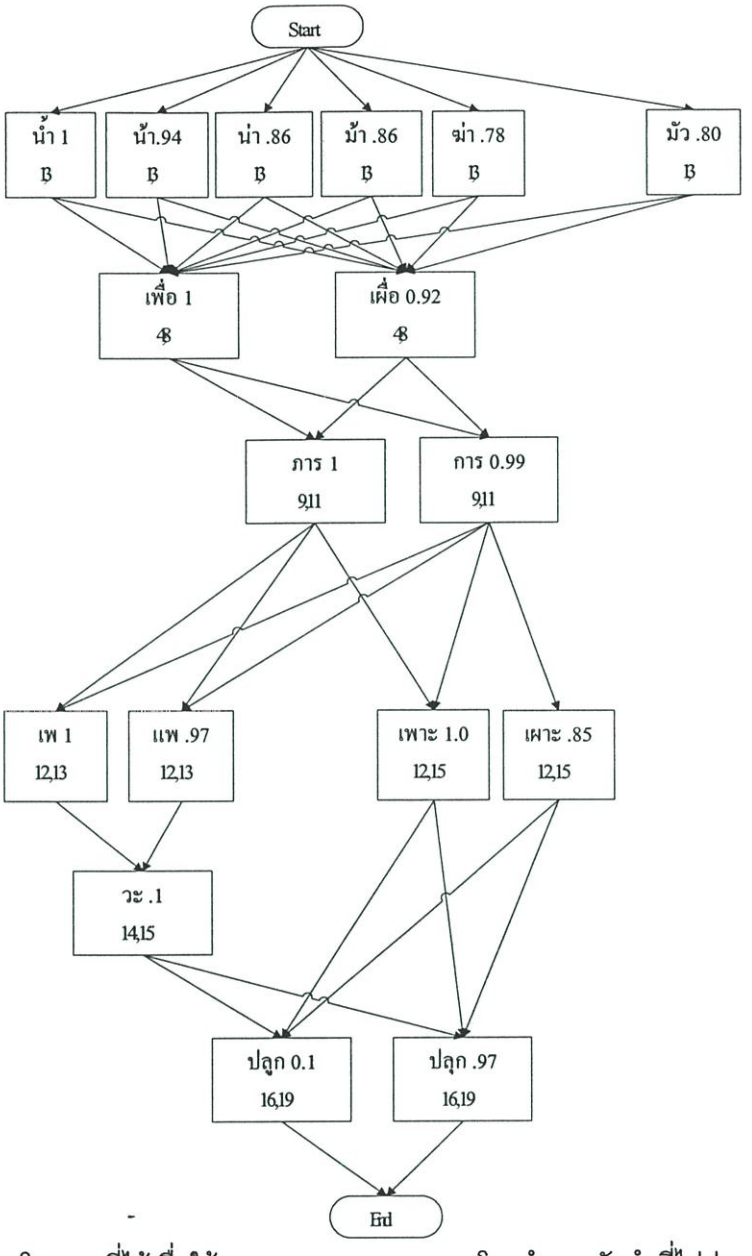


รูปที่ 4.1 แสดงโทคเกนในระดับคำที่ได้หลังการสร้างและส่งต่อโทคเกนในระดับตัวอักษร



รูปที่ 4.2 แสดงโทคเกนที่ได้ทำการตัดโทคเกนที่ไม่สามารถส่งต่อไปได้ หลังการสร้างและส่งต่อโทคเกนในระดับตัวอักษรไปยังโทคเกนระดับคำ

จากรูปที่ 4.2 นี้ เราจะพบว่าคำบางคำนั้นไม่ค่อยถูกใช้แล้ว ดังนั้นถ้าเราทำการตัดคำที่ไม่มีการใช้ออกไปจะเป็นการ ลดจำนวนคำหรือ โทคเกน ลงได้ ซึ่งการลด โทคเกน ที่เกิดขึ้นนี้ สามารถทำได้โดยตัด คำที่ไม่มีการใช้งานแล้วออกจาก พจนานุกรม หรือใช้พจนานุกรมที่เหมาะสมกับงานที่ใช้

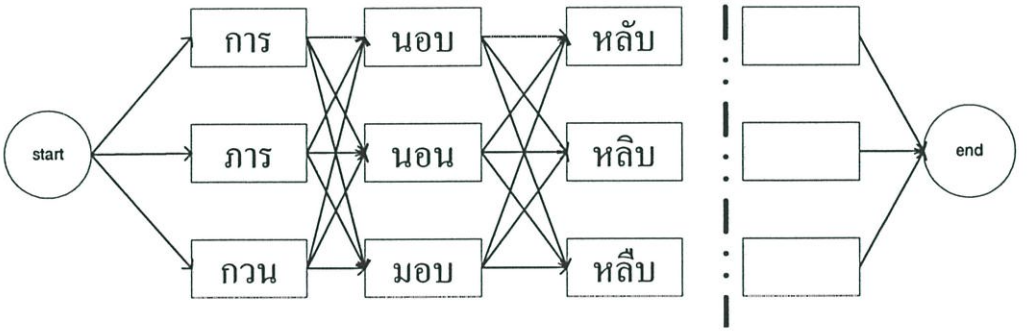


รูปที่ 4.3 แสดงโทคเกนที่ได้เมื่อใช้ พจนานุกรมลดขนาดลงโดยทำการตัดคำที่ไม่ค่อยถูกใช้งานออกไปแล้ว

จากรูปที่ 4.3 จะพบว่าจำนวนประโยคที่สร้างขึ้น 192 ประโยค เทียบกับรูปที่ 4.2 ซึ่งมีจำนวนประโยคทั้งสิ้น 420 ประโยค

4.5 การลดจำนวนโทคเกนที่เกิดขึ้น

4.5.1 ผลของจำนวนโทคเกนและจำนวนประโยค



รูปที่ 4.4 แสดงตัวอย่างโทคเกน

จากตัวอย่างรูปข้างบนนี้จากคำเพียงสามคำหรืออักษรจำนวนสิบตัว สามารถเกิดประโยคขึ้นได้ถึง 3*3*3=27 ประโยค

ดังนั้น ถ้าในหนึ่งบรรทัด มี 68 ตัวอักษร และให้ 4 ตัวอักษรเป็นหนึ่งคำ หรือหนึ่ง โทคเกน แล้วหนึ่งบรรทัดที่เขียนติดต่อกันโดยไม่มีวรรค อาจมีคำได้ 17 คำ สมมติให้แต่ละคำสร้าง โทคเกนคำที่เป็น candidate ออกมา โทคเกน ละ 3 จะมีประโยคที่เกิดขึ้นได้ ถึง

3^17 = 129,140,163 ประโยค

ถ้าหนึ่งบรรทัดมีแบ่งตรงระหว่างครึ่งบรรทัด แต่เกิด โทคเกน ในแต่ละคำเป็น 5

3^9 = 19,683 ประโยค

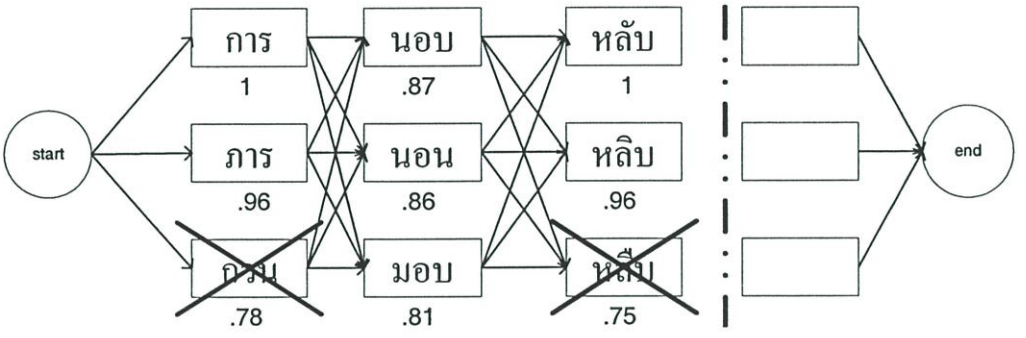
ถ้าหนึ่งบรรทัดมีแบ่งวรรคที่ประมาณกึ่งกลางบรรทัด แต่เกิด โทคเกน ในแต่ละคำเป็น 5

5^9 = 1,953,125 ประโยค

พบว่าโอกาสที่เกิดจำนวนประโยคขึ้นเป็นจำนวนมาก ดังนั้นจึงต้องหาทางในการลดจำนวนประโยคที่เกิดขึ้น

4.5.2 การลดจำนวนโทคเกน

1) พิจารณาจากคะแนนของโทคเกนที่เกิดขึ้นเมื่อทำการ passing ถ้าประโยคที่สร้างมีค่าคะแนนรวมอยู่ในระดับคำที่ต่ำกว่าค่าที่ตั้งไว้ก็จะไม่ทำการส่ง โทคเกนต่อไป



รูปที่ 4.5 แสดง คะแนนของแต่ละโทคเกนในระดับคำ

ตัวอย่างจากรูปที่ 4.5

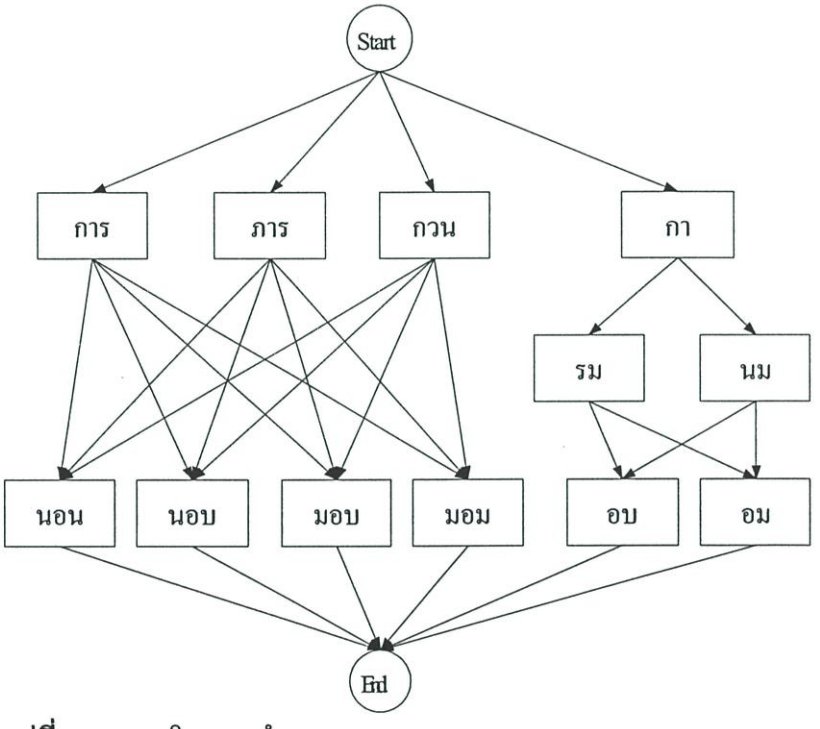
เมื่อกำหนดว่าคำที่มีคะแนนต่ำกว่า .79 จะไม่นำมาใช้ ทำให้สามารถลดจำนวนประโยคลงไปจาก 27 เหลือ 12 ประโยค

2) พิจารณาจาก โทคเกน ว่าแต่ละโทคเกนในระดับคำนั้นไม่น่าจะเป็นคำที่ประกอบขึ้นมาจากตัวอักษรซึ่งไม่ใช่ตัวที่มีค่าคะแนนสูงสุดทุกตัวอักษร

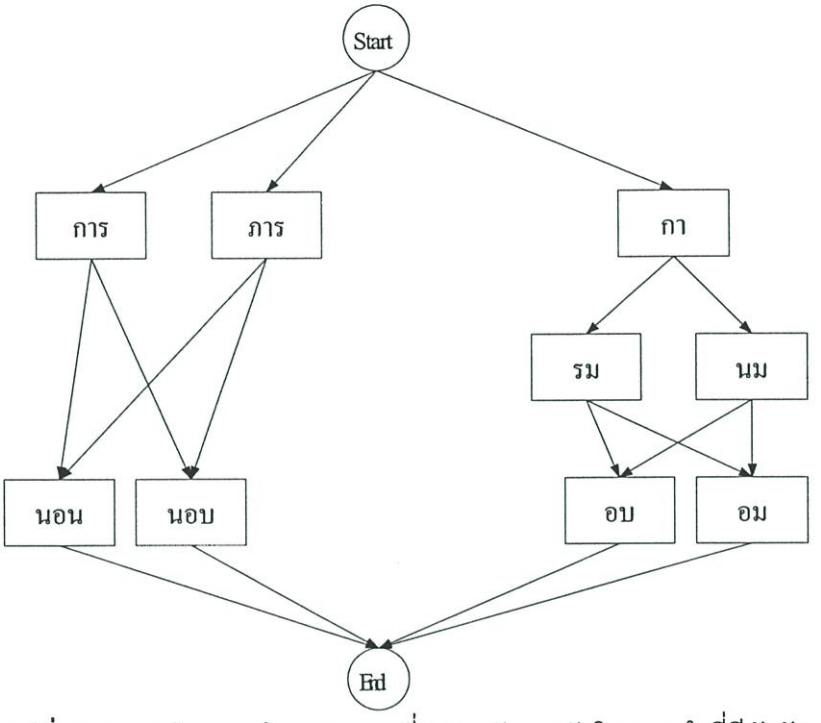
ตารางที่ 4.6 แสดงตัวอย่างและลำดับคะแนน โทคเกนตัวอักษร

ตัวที่ / ลำดับที่	1	2	3	4	5	6
1	ก	า	ธ	น	อ	น
2	ภ	ว	ร	ม	ฮ	บ
3			น		ฉ	ม

จากตาราง 4.6 เราจะได้โทคเกนดังรูปที่ 4.7 และเมื่อเราเลือกกำหนดให้ แต่ละคำนั้นสามารถประกอบขึ้นจากตัวอักษร ที่ไม่ใช่ตัวที่มีคะแนนมากสุดได้ไม่เกิน 2 ตัวอักษร ในหนึ่งคำ แล้วเราจะได้ โทคเกนที่สามารถสร้างเป็นประโยคได้ดังรูปที่ 4.8



รูปที่ 4.6 แสดงโทคเกนคำจากตาราง 4.6



รูปที่ 4.7 แสดงโทคเกนคำจากตารางที่ 4.6 หลังจากตัดโทคเกนคำที่มีตัวอักษรที่ ประกอบขึ้นจากตัวอักษรที่มีคะแนนสูงสุดเกิน หนึ่งตัวออกไปแล้ว

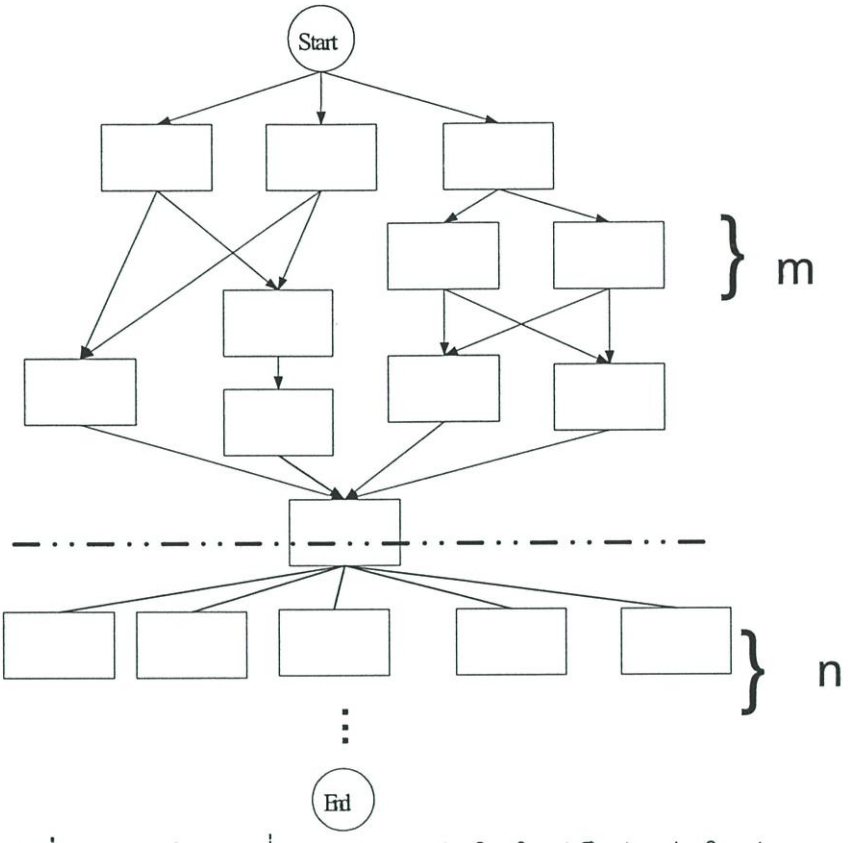
รูปที่ 4.6 มีประโยค 14 ประโยค

รูปที่ 4.7 มีประโยค 8 ประโยค

ซึ่งสำหรับจำนวนประโยคที่มากกว่านี้จะเห็นผลได้อย่างชัดเจน

ปัญหาต่อไปคือเมื่อเราเจอประโยคที่เขียนติดต่อกันทำให้เกิดประโยคขึ้นเป็นจำนวนมหาศาลเราต้องหาวิธีการลดจำนวนประโยคลง

3) ทำการแยกประโยคเป็นส่วน ๆ นำการแบ่งแยกประโยคมาใช้แล้วทำการตัดประโยคเป็นส่วน ย่อย ๆ สำหรับวิธีการนี้อาจจะมองได้ว่าเป็นการสร้าง โหนดในระดับที่ใหญ่ขึ้นเป็นโหนดในระดับประโยค จากรูปที่ 4.8 ประโยคทั้งหมดที่ได้ จะเท่ากับ mxnx... ไปจนจบประโยค วิธีนี้จำนวนประโยคจะเท่าเดิมแต่จะช่วยลดจำนวนประโยคที่ต้องสร้างทั้งหมดลง แต่เราสามารถที่จะลดประโยคลงได้โดยเลือกตัดประโยคที่มีคะแนนต่ำ ทิ้งไป



รูปที่ 4.8 แสดงโทคเกนที่ถูกตัดแบ่งจากประโยคใหญ่เป็นส่วนประโยคย่อย ๆ

บทที่ 5
วิธีดำเนินงานวิจัย

ขั้นตอนหลักในงานวิจัยนี้ประกอบด้วย

การรวบรวมฐานข้อมูล (corpus)

การสร้าง Language model

การจำลอง ผลลัพธ์ที่ได้จาก OCR

ทำการสร้างคำและประโยคโดยใช้เทคนิค โทคเกนพาสซิง (Post processing)

5.1 การรวบรวมฐานข้อมูล

ฐานข้อมูลที่ใช้ในงานวิจัยนี้ ได้มาจาก การรวบรวมจาก web site ต่าง ๆ เก็บไว้ในลักษณะของ text file โดยแยกแต่ละเรื่องด้วย เครื่องหมาย <p> จากไฟล์ข้อความ ที่ได้ นำไปทำการตัดคำโดยใช้เครื่องหมายว่าง (space bar) เป็นตัวแบ่งคำ ซึ่งในงานวิจัยนี้ได้ใช้ ฐานข้อมูลขนาด 10 Mb

5.2 การสร้าง Language model

การสร้าง Language model จะใช้เครื่องมือ จาก CMU-CAMBRIDGE TOOLKIT โดยนำมาจาก http://svr-ww.eng.cam.ac.uk/~prc14/toolkit.html เพื่อทำการเปรียบเทียบ Language model สำหรับเฉพาะเรื่องและ Language model แบบรวม (Mix) และแต่ละ Language model ที่สร้างจะแบ่งฐานข้อมูลที่ใช้สร้างเป็น ชุด ๆ เหมือนเพื่อใช้เป็นตัวเปรียบเทียบ ผลของขนาดของ corpus ที่นำมาใช้เป็น Language model กันดังนี้

LANGUAGE MODEL โดยใช้ข้อมูลจาก corpus ที่เนื้อหาเป็นเรื่องประเภทเดียวกัน

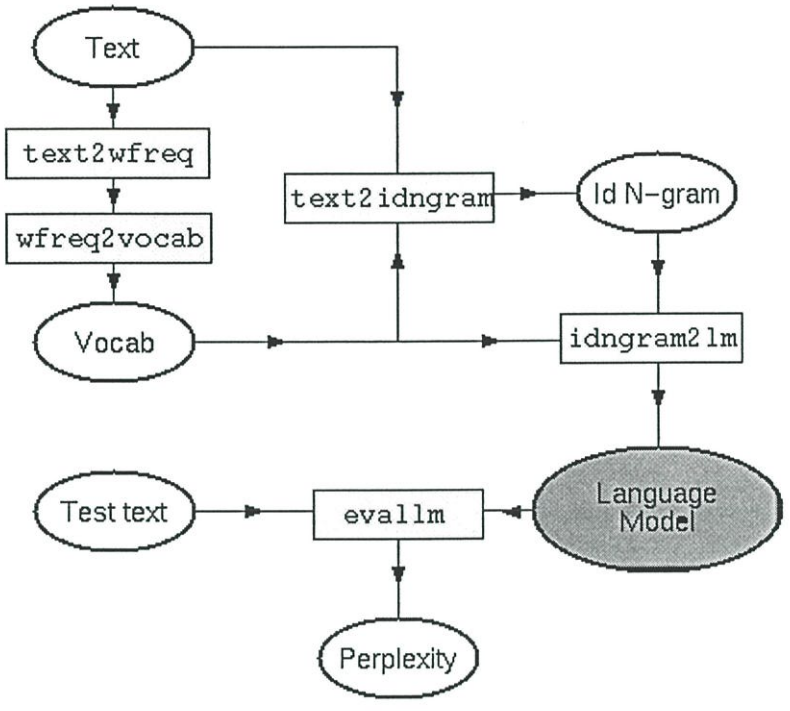
LANGUAGE MODEL โดยใช้ข้อมูลจาก corpus โดยรวมหลาย ๆ เรื่องเข้าด้วยกัน

เตรียมข้อมูลเพื่อนำมาใช้ทดสอบ corpus โดยใช้ ข้อมูลประมาณ 80% เป็นตัวฝึกฝน และใช้ 20% เป็นข้อมูลสำหรับทดสอบ

โปรแกรมที่ใช้สร้าง Language model นี้ ทำงานบน unix หรือ Linux ดังนั้นจึงต้องทำการติดตั้ง ระบบ Linux ไว้ในเครื่องด้วย หรือจะทำการ คอมไพลแล้วใช้งานโปรแกรมบนเครื่องของสถาบันก็ได้ แต่เวลาทำงานต้องการเนื้อที่ในการเขียนข้อมูลซึ่งเป็นไฟล์ชั่วคราวประมาณ 500 เมกาไบต์ ส่วนตัว โค้ดและเอกสารของโปรแกรมประมาณ 1 เมกาไบต์ ต้องการหน่วยความจำขณะทำการสร้าง แนะนำให้ใช้ 100 M. ขึ้นไป

ทำการคอมไพล์โปรแกรม และเตรียม corpus โดย corpus ที่ใช้จะต้องเว้นช่องว่างระหว่างคำ

สำหรับรายละเอียดในการสร้าง และตัวแปรต่าง ๆ ดูได้จากเอกสารซึ่งมาพร้อมกับโปรแกรม



รูปที่ 5.1 แผนผังการสร้าง แบบจำลองทางภาษา

ทำการดัดแปลงแก้ไขโค้ดโปรแกรมในส่วน evallm เพื่อให้สามารถทำงานใน dos ได้

5.3 การจำลองผลลัพธ์ที่ได้จาก OCR

สร้างตารางเทียบตัวอักษรที่คล้ายคลึงกัน ดังตาราง ในภาคผนวก เก็บเป็น ไฟล์ชื่อ map.txt

แบ่งส่วนของประโยคที่อยู่ติดกันภายใน บรรทัดแยกเป็นไฟล์เป็น input ของ ocr เพื่อใช้เป็นข้อมูลในการสร้างโทคเกนตัวอักษร และค่าคะแนนของแต่ละตัวอักษร ซึ่งค่าคะแนนทำการสุ่มโดยโปรแกรม genctk.exe

genctk input.txt

จะได้ inputresfil.txt (input.txt เป็นไฟล์ input ซึ่งสามารถใช้ชื่ออื่นได้ โดยไฟล์ที่เป็นผลลัพธ์จะต่อท้ายด้วยคำว่า resfil.txt) ออกมาเป็น โทคเกนในระดับคำ พร้อมกับตำแหน่งและค่าคะแนนของคำ

สร้างประโยคโดยใช้ token passing ในระดับคำโดย ใช้ list.exe

list inputresfil.txt

จะได้ inputresfil.txtout.txt ซึ่งเป็นไฟล์ที่ประกอบด้วยประโยค โดยปิดท้ายแต่ละประโยคด้วย $ และ แยกระหว่างประโยคแต่ละกลุ่มด้วย #

$
และ ยัง ใช่ เป็น $
และ ยัง ใช้ เป็น $
และ บัง ใช้ เป็น $
และ บัง ใช่ เป็น $
และ ม่ง ใช้ เป็น $
และ ม่ง ใช่ เป็น $
แสะ ยัง ใช้ เป็น $
แสะ ยัง ใช่ เป็น $
แสะ บัง ใช้ เป็น $
แสะ บัง ใช่ เป็น $
#
แหล่ง เลี้ยง $
แหล่ง เลี่ยง $
แห ล่ง เลี้ยง $
แห ล่ง เลี่ยง $
.....

จากนั้นจึงนำ ไฟล์ inputresfil.txtout.txt ไปหาค่า perplexity โดยใช้ evallm.exe ผลลัพธ์ที่ได้ออกมาจะบอกหมายเลขบรรทัดที่มีคะแนนดีที่สุดในแต่ละประโยคและค่า perplexity ของประโยคนั้น

บทที่ 6
ผลการทดลอง

ตารางที่ 6.1 แสดงตัวอย่างของความสัมพันธ์ระหว่าง จำนวน ตัวอักษร,โทคเกน,ประโยค และเวลาในการสร้างคำ

จำนวนตัวอักษร	จำนวนโทคเกน(คำ)	จำนวนประโยค	เวลาในการสร้างคำ(วินาที)
9	13-17	60-128	0.05-0.10
11-13	3-30	2-492	0.03-0.17
14-15	14-65	9-107,696	0.05-0.27
16-19	16-54	144-16,352	0.12-0.20
21-27	20-57	384-58,240	0.15-0.24
46	70	12x50x61	0.791
49	116	36x68x14x16x32	1.752
63	126	375x23x33x4900x8	1.301
127	272	72x796x161x109x34x97x172x516x374	1.27

หมายเหตุ จากตารางจำนวนประโยคบางส่วนเขียนเป็น nnn x nnn หมายถึงประโยคถูกแบ่งออกเป็นส่วนย่อย ๆ จำนวน nnn ประโยค และมีประโยคประมาณ 7% การพรุนนิง ไม่สามารถลดจำนวนประโยคที่เกิดขึ้นได้มากนัก และพบว่า เกิดประโยคน้อยแต่ใช้เวลาเกือบนาที โดยมีที่สังเกตว่า มีจำนวนระดับชั้น (level) สูงและชั้นเกิดการเหลื่อมล้ำกันซึ่งไม่สามารถตัดได้หรือตัดคำได้น้อย

ตารางที่ 6.2 แสดงเวลาที่ใช้ในการสร้าง sentence และเลือก sentence

จำนวน			เวลาในการสร้างประโยค (sec.)	เวลาในการเลือกประโยค (sec)
ตัวอักษร	โทคเกน	word per level		
49	116	36x68x14x16x32	0.12	0.1
46	70	12x50x61	0.08	0.1
63	126	375x23x33x4900x8	0.34	1.2
89	109	19x21x23x23x23x11x2	0.13	0.2
39	119	1027x493x1580x1	0.15	0.5
96	227	3028x323x54x2420x12x50x11	0.26	1.2

ตารางที่ 6.3 ตัวอย่างแสดงเวลาที่ใช้ในการทำงานของ แบบจำลองทางภาษา (5-gram) เพื่อหาประโยคที่มีค่า perplexity ดีที่สุด

คำในประโยค (node)	จำนวนประโยค (stream)	เวลา (วินาที)	คำที่ต้องคำนวณ	คำ/วินาที	บรรทัด/วินาที
7	18104	4.6	126728	27550	3936
7	11511	2.3	80577	35033	5005
7	1331	0.3	9317	31057	4437
6	5953	1.1	35718	32471	5412
5	10024	1.6	50120	31325	6265
5	658	0.1	3290	32900	6580
5	2412	0.5	12060	24120	4824
4	2896	0.4	11584	28960	7240

จากตัวอย่างทั้งหมด 318 ประโยค เมื่อใช้โทคเกนพาสซิงและทำการเลือกประโยคด้วย แบบจำลองภาษาทางสถิติ สามารถแก้คำให้เป็นคำที่ถูกต้องได้ 84.2% ของประโยคที่ผิด

บทที่ 7
สรุปผลการวิจัยและข้อเสนอแนะ

จากผลการทดลองสรุปได้ว่าสามารถแก้ไขคำที่ผิดได้ 84% โดยที่เทคนิคโทคเกนพาสซิงนี้ ช่วยทำให้เกิดความคล่องตัวเป็นอย่างมากตั้งแต่การสร้างคำจากชุดตัวอักษรที่คล้ายคลึงกัน สามารถตัดคำได้ภายในตัวเองโดยขอบเขตของแต่ละโทเกนก็คือขอบเขตของแต่ละคำ และการส่งพาสซิงของโทคเกนคำในขั้นต่อมาเพื่อสร้างเป็นประโยค การลดจำนวนประโยคที่ต้องสร้างทั้งหมด

เมื่อใช้แบบจำลองทางภาษาภายใน domain เดียวกันกับประโยคที่นำมาตรวจสอบนั้นจะทำให้ผลลัพธ์ที่ดี และเมื่อตัวแบบจำลองทางภาษาและประโยคที่ใช้มาจากต่าง domain กันทำให้ผลความถูกต้องลดลงอย่างเห็นได้ชัด

สำหรับ ในการวิจัยนี้พบว่า แบบจำลองภาษาทางสถิติที่เหมาะสมสำหรับภาษาไทยคือแบบจำลอง 5-grams

ข้อเสนอแนะ

ในงานวิจัยนี้ เป็นการทดลองซึ่งใช้กับภาษาไทยเป็นหลัก การจะนำไปประยุกต์ใช้กับภาษาอื่นนั้น มีข้อสังเกตคือ หากนำไปใช้กับภาษาที่มีตัวอักษรเป็นจำนวนน้อยเช่นภาษาอังกฤษ การนำเอาตัวอักษรที่มีคะแนนรองลงมาหลายตัว จะทำให้เกิด โทคเกน หรือ คำขึ้นเป็นจำนวนสูงมาก ดังนั้นการเลือกจำนวนตัวอักษรที่รับเข้ามา ควรพิจารณาในจุดนี้ด้วย

น่าจะลองนำเอา Genetic Algorithm มาใช้ร่วมกับการส่งโทคเกน เพื่อช่วยในการหาเส้นทางที่ถูกต้องของประโยค

นอกจากนี้ในส่วนของแบบจำลองทางภาษา เทคนิคการทำ smooting ยังมีเทคนิค อื่น ๆ ที่น่าจะนำมาใช้เพื่อเพิ่มความสามารถในการทำงาน ของ Language model เช่น

การทำ caching เช่นเมื่อเจอคำบางกลุ่มที่ถูกใช้บ่อย ในขณะทำการหาค่าของแบบจำลองทางภาษาก็จะทำการปรับค่าของกลุ่มคำที่พบบ่อยให้สูงขึ้น

การทำ smooting ของ Kneser-Ney

สำหรับในส่วนของพจนานุกรมนั้น อาจจะมีพจนานุกรมที่แยกแยะจัดไว้เป็นพิเศษเพื่อเพิ่มความถูกต้องเช่น พจนานุกรมชื่อ สถานที่ พจนานุกรมคำเฉพาะทางเพื่อเพิ่มความถูกต้อง

ในงานวิจัยนี้ส่วนเป็นเพียงส่วนหนึ่งในการแก้ไขความความผิดพลาดที่เกิดขึ้นเนื่องจากความผิดพลาดในกรณีที่ผลการรู้จำผิดเท่านั้น ส่วนในกรณีที่มีตัวอักษรขาดหรือเกิน ยังไม่สามารถทำได้จึงต้องทำการพัฒนาต่อไป

การนำผลงานวิจัยไปประยุกต์ใช้กับงานอื่น

ผลงานที่ได้จากการทำวิจัยนี้ สามารถนำไปประยุกต์ได้กับงานด้าน การประมวลผลภาษาธรรมชาติ (Natural Language Processing) สาขาต่าง ๆ ทั้งในด้าน การรู้จำเสียง ภาษาศาสตร์ และการประมวลผลภาษาทางคอมพิวเตอร์ เช่น การประยุกต์ใช้กับระบบการแปลภาษาด้วยคอมพิวเตอร์

เอกสารอ้างอิง

[1] Littlestone, N. "Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm Machine Learning." [CDROM]. IEEE. 1988.

[2] Coulmas, F.C. "The Writing Systems of the World. Blackwell." [Online]. Available : http://citeseer.nj.nec.com/. 1989.

[3] Stina Nylander and Jussi Karlgren "Statistics and Phonotactical Rules in Finding OCR Errors." [Online]. Available : http://citeseer.nj.nec.com/.

[4] Kazem Taghva et. al. "An expert system for automatically correction OCR output." [Online]. Available : http://citeseer.nj.nec.com/taghva94expert.html. 2001.

[5] Thanaruk Theeramunkong and Sasiporn Usanavasin "Non-Dictionary-Based Thai Word Segmentation Using Decision Trees." [Online]. Available : http://citeseer.nj.nec.com/hlt2001-57.html. 2001.

[6] Witoon Kanlayanawat and Somchai Prasitjutrakul "Automatic Indexing for Thai Text with Unknown Words using Trie Structure." [Online]. Available : http://citeseer.nj.nec.com/. 2000.

[7] Brigitte Krenn and Christer Samuelsson "The Linguist's Guide to Statistics Don't Panic." [Online]. Available : http://www.coli.uni-sb.de/~krenn,~christer. 1997.

[8] S. Charnyapornpong. "A Thai Syallable Separation Algorithm." [Online]. Available : http://citeseer.nj.nec.com/. 1983.

[9] ยืนภู่วรวรรณ และคณะ "การตัดพยางค์คำไทยด้วยโครงสร้างข้อมูลแบบต้นไม้." รายงานการประชุมวิชาการ ทางวิศวกรรมไฟฟ้า 8 สถาบันอุดมศึกษา, ครั้งที่ 8, ธันวาคม 2528.

[10] Kawtrakul Asanee et. al. "Automatic Thai Unknown Word Recognition." [Online]. Available : http://citeseer.nj.nec.com/. 2001.

[11] Pisit Promchan and Yunyong Teng-amnuay "Performance comparison of thai word of thai word separation algorithms" Telecom Asia Corp. Public Co. Ltd. [Online]. Available : http://citeseer.nj.nec.com/. 2001.

[12] S.J. Young, N.H.Russell and J.H.S Thornton, "Token Passing: a Simple Conceptual Model for Connected Speech Recognition Systems.", Cambridge University Engineering Department, 31, [Online]. Available : http://citeseer.nj.nec.com/. 1989.

[13] สรศักดิ์ ไทยแท้. "การตัดคำไทยโดยใช้ดิกชันนารีที่มีโครงสร้างข้อมูลแบบแฮชชิ่ง." วิทยานิพนธ์วิทยาศาสตร์มหาบัณฑิต สาขาเทคโนโลยีสาระสนเทศ บัณฑิตวิทยาลัย, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2541.

[14] Surapant Meknavin et. al. "Progress of Combining Trigram and Winnow in Thai OCR Error Correction." [CDROM] IEEE conference 1998.

[15] สรศักดิ์ ไทยแท้ และบุญธีร์ เครือตราชู. "การตัดพยางค์คำไทย โดยโครงสร้างข้อมูลแบบสแตติกและดิกชันนารีแบบแฮชชิ่ง." วิศวสารลาดกระบัง, ปีที่ 15 ฉบับที่ 1 มกราคม 2541.

[16] Uwe Quasthoff. "Tools for Automatic Lexicon Maintenance: Acquisition, Error Correction, and the Generation of Missing Values." [Online]. Available : http://citeseer.nj.nec.com/. 2001.

[17] Kazem Taghva et. al. "An expert system for automatically correcting OCR output." [Online]. Available : http://citeseer.nj.nec.com/. 2001.

[18] Andrew R. Golding and Dan Roth "A winnow-base Approach to Context-Sensitive Spelling Correction." Kluwer Acadamic Publishers, Boston. ICML 1996.

[19] Lidia Mangu and Eric Brill "Automatic Rule Acquisition for Spelling Correction." [Online]. Available : http://citeseer.nj.nec.com/. 2001.

[20] John M. Trenkle and Robert C. Vogt "Disambiguation and spelling correction for a neural network-based character recognition system." [Online]. Available : http://citeseer.nj.nec.com/. 2001.

[21] Juan C Perez-Cortes et. al. "Stochastic Error-Correcting Parsing for OCR Post-processing." [Online]. Available : http://citeseer.nj.nec.com/. 2001.

[22] Doug Cooper "Fuzzy Letters and Thai Optical Character Recognition." SNLP 1995. [Online]. Available : http://citeseer.nj.nec.com/. 2001.

[23] Peter F. Brown Vincent et. al. "Class-based n-gram models of natural language." December. 17, 1990. [Online]. Available : http://citeseer.nj.nec.com/. 2001.

[24] Lafferty John. "The Noisy Channel Model. Class Notes to Statistical Methods in Language Technologies." [Online]. Available : http://www.cs.cmu.edu/~lafferty/LS/syllabus.html. 2000.

[25] Lalit Bahl, Fred Jelinek and Robert Mercer. "A Maximum Likelihood Approach to Continuous Speech Recognition." IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-5, n. 2, March 1983., pp. 179–190.

[26] S.M. Katz. "Estimation of Probabilities from Sparse Data for the Language Model Component of Speech Recognizer." IEEE Transactions on Acoustics,Speech and Signal Processing, vol. 35,no. 3, 1987. pp. 400-401.

[27] H.Ney, U. Essen, and R. Kneser. "On Structuring Probabilistic Dependencies in Stochastic Language Modelling." Computer Speech and Language, vol. 8, no. 1, 1994. pp. 1-38.

[28] Stolcke, A., "Entropy-based Pruning of Backoff Language Models." DARPA Broadcast News Transcription and Understanding Workshop, 1998, Lansdowne, VA. [Online]. Available : http://citeseer.nj.nec.com/.

[29] Littlestone, N. "Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm." [Online]. Available : http://citeseer.nj.nec.com/.1988.

[30] Paisarn Charoenpornsawat et. al. "Feature-based Thai Unknown Word Boundary Identifications Using Winnow." [Online]. Available : http://citeseer.nj.nec.com/. 1998.

[31] Rattasit Sukhahuta and Dan Smith "Information Extraction Strategies for Thai Documents." [Online]. Available : http://citeseer.nj.nec.com/. 2001.

ภาคผนวก

แสดงการ map ตัวอักษร

ก ภ ถ ฏ ฏ

ข บ

ฃ ข บ

ค ด ต ศ

ค ด ค ศ

ฆ ม

ง

จ ฐ

ฉ อ ฮ

ช ซ

ช ช ข

ฌ ณ ญ

ญ ฌ ณ

ฎ ฏ ก ภ ถ

ฏ ฏ ก ภ ถ

ฐ จ

ฑ ท

ฒ ต

ณ ญ ฌ

ด ต ค ศ

ต ด ค ฒ ศ

ถ ภ ก ฏ ฏ

ท ห

ธ ร

น บ ม ข ษ

บ ข ษ ม น

ป ฝ ฟ พ

ผ บ พ

ฝ ฟ พ์ ผ พ

พ ผ ฟ ผ

ฟ ฝ พ ผ

ภ ถ ก

ม น บ ย ษ

ย บ น ม ษ

ร โ เ

ฤ ฦ

ล ส

ฦ ฤ

ว ำ า

ศ ค ด ต

ษ บ ข

ส ล

ห ท ฑ

พ์ ฟ ฝ

อ ฉ ฮ

ฮ อ ฉ

ฯ ๆ

ะ

า ว ำ

ำ า ว

ฺ

ฺ

ฺ

ฺ

ฺ

ฺ

ฺ

ประวัติผู้เขียน

นายมนัส รอดพ้น เกิดที่จังหวัดระยอง สำเร็จการศึกษา วิศวกรรมศาสตรบัณฑิต (เกษตร) จากสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปีการศึกษา 2541