บทที่ 2

ทฤษฎีและระบบงานที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 หลักการสะกดคำภาษาไทย

การสะกดคำ [1] คือ การอ่านโดยนำพยัญชนะต้น สระ วรรณยุกต์ และตัวสะกดมาประสมเป็นคำอ่าน ถือเป็นเครื่องมือการอ่านคำใหม่ โดยการสะกดคำนั้นมีหลากหลายวิธี ได้แก่

2.1.1.1 สะกดตามรูปคำ

เช่น กา สะกดว่า กอ – อา – กา

2.1.1.2 สะกดโดยสะกดแม่ ก กา ก่อน แล้วจึงสะกดมาตราตัวสะกด

เช่น คาง สะกดว่า คอ – อา – คา – คา – งอ – คาง

2.1.1.3 คำที่มีสระอยู่หน้าพยัญชนะ ให้สะกดพยัญชนะก่อนสระเสมอ

เช่น เก สะกดว่า กอ – เอ – เก, ไป สะกดว่า ปอ – ไอ – ไป

2.1.1.4 คำที่เป็นสระลดรูปหรือสระเปลี่ยนรูป อาจสะกดได้ดังนี้

เช่น กัน สะกดว่า กอ – อะ – นอ – กัน หรือ กอ – ไม้หันอากาศ – นอ – กัน

2.1.1.5 คำอักษรควบ สะกดเรียงตัวอักษร มุ่งเพื่อการเขียนให้ถูกต้อง อาจสะกดได้ดังนี้

เช่น กลอง สะกดว่า กอ – ลอ – ออ – งอ – กลอง

2.1.1.6 คำอักษรนำ สะกดเรียงตัวอักษร มุ่งเพื่อการเขียนให้ถูกต้อง อาจสะกดได้ดังนี้

เช่น อยาก สะกดว่า ออ – ยอ – อา – กอ – อยาก

2.1.1.7 คำที่ตัวสะกดไม่ตรงตามมาตราตัวสะกด คำที่มีตัวการันต์ให้ใช้หลักสังเกตรูปคำ รู้ความหมายของคำและจำคำให้ได้โดยอ่านและเขียนอยู่เสมอ เช่น เหตุ จันทร์

2.1.2 การตรวจสอบการสะกดผิด (Error Detection)

การตรวจสอบการสะกดผิด [2] เป็นการสกัดเอาคำจากข้อความที่ป้อนเข้ามาไปเทียบกับคำศัพท์ที่มีคำที่สะกดถูกต้อง เช่น คำศัพท์จากพจนานุกรม ว่าตรงกันหรือไม่ ถ้าตรวจสอบเจอคำที่สะกดไม่ถูกต้องจะทำการเปลี่ยนสีของคำที่สะกดผิด โดยวิธีนี้สามารถตรวจสอบได้เฉพาะคำที่เป็นการสะกดผิดแบบไม่เป็นคำเท่านั้น

Computer Science Department
Faculty of Informatics Mahasarakham University

2.1.3 การแก้ไขการสะกดผิด (Error Correction)

การแก้ไขการสะกดผิด [2] สามารถแบ่งย่อยออกเป็น ขั้นตอนการค้นหาคำที่น่าจะเป็นไปได้และขั้นตอนการคัดเลือกคำที่ถูกต้องเหมาะสมมากที่สุด ในขั้นตอนของการค้นหาคำที่น่าจะเป็นไปได้ โดยส่วนใหญ่แล้วคำที่น่าจะเป็นมักจะได้มาจากการปรับแก้ไข คำที่สะกดผิดนั้นให้ถูกต้องโดยมีจำนวนของการแก้ไขน้อยครั้งที่สุด และนำไปใช้แก้ไขคำที่สะกดผิด แต่ถ้าเจอคำที่น่าจะเป็นไปได้มากกว่าหนึ่งคำจะต้องเข้าขั้นตอนสุดท้าย คือ การคัดเลือกคำที่ถูกต้องเหมาะสมมากที่สุดจากรายการคำที่น่าจะเป็น ซึ่งวิธีการคัดเลือกคำที่ เหมาะสมที่สุดนั้นจะขึ้นอยู่กับผู้วิจัยของแต่ละคน คือ ถ้าได้คำที่น่าจะเป็นคำที่สะกดถูกก็จะแนะนำคำเหล่านั้นให้ผู้ใช้ได้เลือกคำที่เหมาะสมด้วยตัวเอง

2.2 การประมวลผลภาษาธรรมชาติ (Natural Language Processing: NLP)

การประมวลผลภาษาธรรมชาติ หรือ Natural Language Processing (NLP) [3] คือ หนึ่งในสาขาของวิทยาศาสตร์คอมพิวเตอร์ ที่เกี่ยวกับปัญญาประดิษฐ์ Artificial Intelligence และภาษาศาสตร์เชิงคำนวณคอมพิวเตอร์ Computational Linguistics เป็นศาสตร์ที่ศึกษาเกี่ยวกับการทำให้คอมพิวเตอร์สามารถสื่อสารโต้ตอบด้วยภาษาของมนุษย์ และทำให้คอมพิวเตอร์เข้าใจภาษามนุษย์มากขึ้น เทคนิคที่ใช้ในการประมวลผลภาษาธรรมชาติที่ผู้วิจัยได้นำมาใช้ในงานวิจัยมีดังนี้

2.2.1.1 การตัดคำ (Word Segmentation)

การตัดคำ [4] คือ การแยกข้อความในเอกสารออกเป็นคำ โดยใช้วิธีการต่าง ๆ ได้แก่

2.2.1.2 การใช้กฎการตัดคำ โดยใช้การตรวจสอบกฎเกณฑ์ทางอักขรวิธีที่กำหนดลักษณะการประสมตัวอักษร การเว้นวรรค และการขึ้นย่อหน้า เพื่อใช้เป็นกฎเกณฑ์ในการกำหนดขอบเขตของคำ วิธีการแบบนี้จะมีจำกัดในการทำงาน คือความถูกต้องของการตัดคำในระดับพยางค์สูงมากกว่าความถูกต้องของการตัดคำ

2.2.1.3 การใช้พจนานุกรม การตัดคำโดยการใช้พจนานุกรมเป็นการตัดคำที่ใช้สายอักขระมาเทียบกับคำที่มีอยู่ในพจนานุกรม ซึ่งวิธีนี้จะต้องจัดเก็บคำไว้ในพจนานุกรม และวิธีนี้ทำให้ได้ความถูกต้องในการตัดคำสูงมากกว่าการใช้กฎแต่จะใช้เวลานานมากกว่า

2.2.1.4 การใช้คลังข้อความ เป็นการตัดคำโดยใช้คลังข้อความเป็นการตัดคำโดยนำวิธีการทางสถิติเข้ามาใช้ในการประมวลผลภาษาที่ใช้คลังข้อความในการตัดคำ ซึ่งการตัดคำโดยใช้คลังข้อมูลแบ่งออกเป็น 2 วิธี คือการตัดคำโดยอาศัยความน่าจะเป็น (Probabilistic Word Segmentation) และวิธีการตัดคำโดยอาศัยคุณลักษณะของคำ (Feature-based Word Segmentation)

Computer Science Department
Faculty of Informatics Mahasarakham University

ตารางที่ 2.1 ตัวอย่างของกแบบจำลองไตรแกรม

ประโยคต้นฉบับ	ผลการตัดคำโดยใช้แบบจำลองเอ็นแกรม
“การพัฒนาระบบถาม-ตอบ”	การ/ ารพ/ รพั/ พัฒ/ ัฒน/ ฒนา/ นาร/ าระ/ ระบ/ บบถ/ บถา/ ถาม/ ามต/ มตอ/ ตอบ

2.2.2 เทคนิคที่ช่วยในการตัดคำ

เทคนิคที่ช่วยในการตัดคำ [5]

2.2.2.1 วิธีการเทียบคำที่ยาวที่สุด (Longest Word Pattern Matching) วิธีนี้จะทำการตรวจสอบสายอักขระที่นำเข้ามาจากซ้ายไปขวา และนำไปเปรียบเทียบกับคำที่มีอยู่ในพจนานุกรม หากตรวจสอบพบว่าพบพยางค์มากกว่า 1 พยางค์ในพจนานุกรม จะทำการเลือกพยางค์ที่ยาวที่สุดแล้วทำต่อไปเรื่อย ๆ จนจบสายอักขระ

ตัวอย่างคำว่า “กองกลาง” การตัดคำโดยวิธีนี้จะนำสายอักขระไปเปรียบเทียบกับคำที่มีอยู่ในพจนานุกรมจะพบคำว่า ก , กอและคำว่ากอง สวนคำว่ากองก ไม่พบอยู่ในพจนานุกรม

ดังนั้นจึงได้คำว่ากอง ซึ่งเป็น คำที่ยาวที่สุดที่หาพบ สวนที่เหลือคือ กลาง เมื่อนำไปค้นในพจนานุกรมจะได้ว่า ก , กล , กลาง ดังนั้นจึงเลือกคำว่า กลาง คำที่ได้จากการตัดคำโดยวิธีการนี้จึงเป็น กอง กลาง

2.2.2.2 วิธีการเทียบคำที่สั้นที่สุด (Shortest Word Pattern Matching) วิธีการนี้คล้ายกับวิธีการเทียบคำที่ยาวที่สุดเพียงแต่จะเลือกคำที่สั้นที่สุดที่พบก่อน แต่วิธีนี้พบว่าได้จำนวนคำมากที่สุดแต่ความถูกต้องของคำหลังทำการตัดคำน้อยกว่า

ตัวอย่างคำว่า “โคลงเรือ” การตัดคำโดยวิธีนี้จะเลือกเอาคำแรกที่ค้นหาเจอจากพจนานุกรม ดังนั้นจะได้ว่า โค ลง เรือ (โดยไม่เลือกคำว่า “โคลง” ที่จะพบต่อไปภายหลังหากทำการค้นหาต่อ) วิธีนี้ใช้เวลาน้อยกว่า และความถูกต้องที่ได้การตัดคำจะมากกว่า

2.2.2.3 วิธีการตัดคำที่ใช้ความถี่ของคำ (Word Usage Frequency) วิธีการนี้เป็นแนวทางหนึ่งในการแก้ปัญหาคำกำกวมของประโยคภาษาไทยโดยการ วิเคราะห์ความถี่ของการใช้คำในชีวิตประจำวันโดยจัดเรียงคำในพจนานุกรมตามความถี่ที่พบ

ตัวอย่างคำว่า “ก็อด” ในกรณีนี้หากใช้ความถี่ของคำจะได้ว่า ก็ อด

2.2.2.4 วิธีการย้อนรอยกลับ (Backtracking) เมื่อทำการเปรียบเทียบคำที่นำมาตัดคำกับคำที่มีอยู่ในพจนานุกรม อาจพบกรณีที่คำที่พบมี มากกว่า 1 คำแล้วทำการเลือกคำที่ยาวที่สุดทำให้สาย

Computer Science Department
Faculty of Informatics, Mahasarakham University

อักขระที่ตามมาจากคำนั้นไม่สามารถตัดคำได้ เนื่องจากไม่พบตามพจนานุกรม กรณีนี้จะทำการย้อนไปอีกคำที่ไม่ถูกเลือกแล้วทำการตัดคำต่อไป

ตัวอย่างเช่นคำว่า “เมื่อยามนี้” การเปรียบเทียบกับพจนานุกรมจะได้ว่า เมื่อ , เมื่อย ดังนั้นจึงเลือกคำที่ยาวที่สุดจะได้คำว่า เมื่อย สวนที่เหลือคือ –าม นี้ซึ่งไม่พบอยู่ในพจนานุกรม ดังนั้นจะทำการย้อนกลับไปเพื่อเลือกอีกคำหนึ่งคือเมื่อจะได้เป็น เมื่อยาม นี้ โดยคำว่ายามเกิดจากการเลือกคำที่ยาวที่สุดระหว่างยาและยาม

ตารางที่ 2.2 ตัวอย่างการตัดคำ

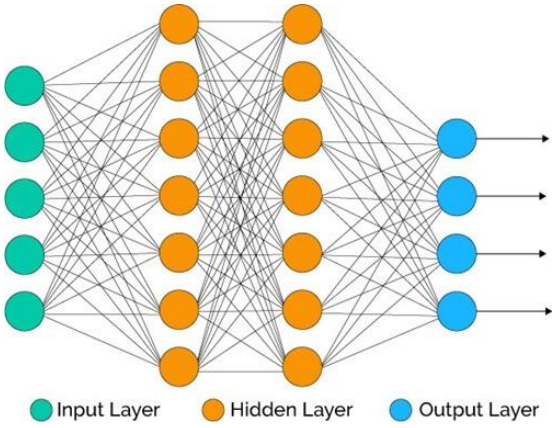
ประโยคต้นฉบับ	ผลการตัดคำ
ที่ถ่ายทอดมาจากบรรพบุรุษเดียวกัน ตามทฤษฎีวิวัฒนาการของสัตว์	ที่/ถ่าย/ทอด/มา/จาก/บรรพบุรุษ/เดียว/กัน ตาม/ทฤษฎี/วิวัฒนาการ/ของ/สัตว์
ความโสดไม่ใช่เนื้อคู่	ความ/โสด/ไม่/ใช่/เนื้อคู่

2.2.3 Neural Networks (NN)

Neural Networks [6] หรือ โครงข่ายประสาทเทียมนั้นมาจากเลียนแบบการทำงานของระบบประสาทของมนุษย์ จะจำลองการทำงานเหมือนกลุ่มเซลล์ประสาทที่เชื่อมโยงกัน สามารถรับรู้หลาย ๆ ได้ในเวลาเดียวกัน ด้วยการประมวลผลแบบขนาน (Parallel Network) ทำให้ระบบสามารถตัดสินใจได้ใกล้เคียงกับมนุษย์

การทำงานของ Neural Networks โดยจะมี ชั้น Input Layer ซึ่งรับข้อมูลจากภายนอก มาที่ชั้น Hidden Layer จะอยู่ระหว่าง Input layer และ Output Layer ทำหน้าที่ประเมินชุดข้อมูลหรือการแปลงสิ่งที่ Input Layer เข้ามาให้สามารถนำออกไปใช้ใน Output Layer ได้ ส่วนของชั้น Output Layer จะส่งออกข้อมูลที่แปลงมาแล้วหรือนำเสนอข้อมูลที่ได้ออกมา เครือข่ายประสาทส่วนใหญ่มีการเชื่อมต่อกันหมด หมายถึงว่า Hidden Layer ทั้งหมดจะเชื่อมต่อกับเซลล์ประสาททุกตัวในชั้น Input Layer และชั้น Output Layer

Computer Science Department
Faculty of Informatics, Mahasarakham University

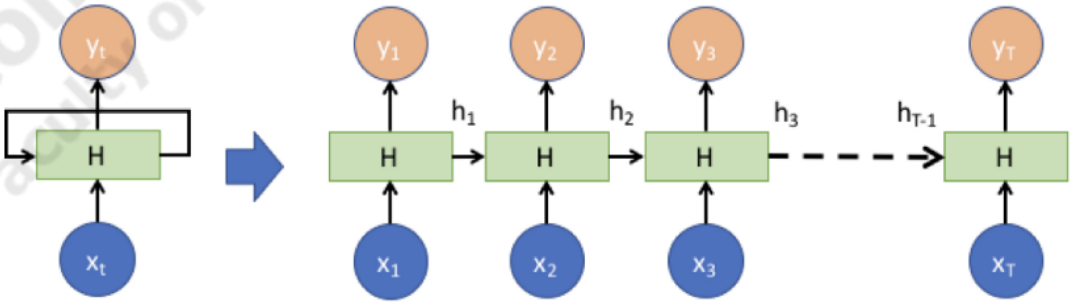


รูปภาพประกอบที่ 2.1 โครงสร้าง Neural Networks

2.2.4 Recurrent Neural Network (RNN)

Recurrent Neural Network (RNN) [7] เป็นลักษณะทั่วไปของโครงข่ายประสาทเทียมแบบ Feed Forward ที่มีหน่วยความจำภายใน RNN จะเกิดขึ้นซ้ำอีกเนื่องจากทำหน้าที่เดียวกันสำหรับทุกอินพุตของข้อมูลในขณะที่เอาต์พุตของอินพุตปัจจุบันขึ้นอยู่กับการคำนวณหนึ่งครั้งที่ผ่านมา หลังจากสร้างเอาต์พุตแล้วจะถูกคัดลอกและส่งกลับไปยังเครือข่ายที่เกิดซ้ำ สำหรับการตัดสินใจจะพิจารณาอินพุตปัจจุบันและเอาต์พุตที่ได้เรียนรู้จากอินพุตก่อนหน้า

ซึ่งแตกต่างจากเครือข่ายประสาทเทียม Feed Forward RNN สามารถใช้สถานะภายใน (หน่วยความจำ) เพื่อประมวลผลลำดับของอินพุต สิ่งนี้ทำให้สามารถใช้ได้กับงานต่าง ๆ เช่น การรู้จำลายมือที่ไม่มีการแบ่งส่วนการเชื่อมต่อหรือการรู้จำเสียง ในเครือข่ายประสาทอื่น ๆ อินพุตทั้งหมดไม่ขึ้นอยู่กับกันและกัน แต่ใน RNN อินพุตทั้งหมดมีความสัมพันธ์ซึ่งกันและกัน



รูปภาพประกอบที่ 2.2 การทำงานของRecurrent Neural Network

Computer Science Department
Faculty of Informatics Mahasarakham University

โดยที่

H = Hidden Layer

y_t= Output จาก RNN ที่เวลา t

x_t= Input Data ที่เวลา t

h_t= Hidden Date ที่เวลา t

จากรูปภาพประกอบที่ 2.2 แสดงรูปทางซ้ายมี Loop ที่วนกลับเข้ามาที่ Hidden Layer ของ Neural Network ดังนั้นที่บอกไปแล้วว่าสิ่งสำคัญสิ่งหนึ่งของ RNN คือ Hidden State ก่อนหน้า และ Input Data ตอนนั้น ประโยชน์หลักๆ ของ Loop ก็เพื่อที่จะนำ Hidden State ก่อนหน้ากลับเข้ามาใช้ (หรืออาจจะมองว่า RNN คือ Neural Network ที่มี Memory เพิ่มขึ้นมา เพื่อเก็บค่า Hidden State ที่ได้คำนวณเอาไว้ก่อนหน้า) ส่วนในรูปทางขวาเป็นรูปที่คลี่รูปทางซ้ายออกมาแล้วเพื่อแสดงการทำงานเป็นทีละขั้นตอน สมการ Recurrent Neural Network เขียนได้ดังนี้

$$h_t=f_h(U_h h_{t-1}+W_h x_t+b_h) \quad (1)$$
$$y_t=f_y(W_y h_t+b_y)$$

โดยที่

f_h คือ Activation Function ของ Hidden layer (เช่น tan h หรือ ReLU หรือ Sigmoid function)

f_y คือ Activation Function ของ Output layer (เช่น Softmax Function)

W_h คือ Weight Matrix ของ Hidden layer

U_h คือ Hidden-State-to-Hidden-State Matrix (หรือ Transition Matrix)จะเห็นว่าการที่จะคำนวณ Hidden state ที่เวลา t ออกมาได้นั้น (h_t) จะต้องใช้ 2 ตัวแปรสำคัญ หรือ Hidden State ก่อนหน้า (h_{t-1}) และ Input Data ณ ตอนนั้น (x_t)

การที่จะ Update Weights จะอาศัย Back Propagation Algorithm ซึ่งจะคำนวณ Gradient ของ Loss Function (E) เพื่อ Update Weights แต่สำหรับ RNN ก็จะมีความยุ่งยากอยู่ เพราะว่าการที่จะได้ Output y_t มานั้น มันไม่ได้มาจากช่วงเวลา t=t อย่างเดียว แต่ลากยาวมาตั้งแต่ที่ t=t-1, t-2, ... จนถึง t=1 มาเรื่อย ๆ (ผ่านการใช้ Hidden State และ Input Data ของก่อนหน้า ของก่อนหน้า ของก่อนหน้า มาเรื่อย ๆ) ดังนั้นการทำ Back Propagation มันก็เลยต้องร่วมคำนวณหมดตั้งแต่ t=1 จนถึง t=t หรือก็คือ

$$\frac{\partial E}{\partial w}=\sum_{k=1}^{t}\frac{\partial E_t}{\partial y_t}\frac{\partial y_t}{\partial h_t}\frac{\partial h_t}{\partial h_k}\frac{\partial h_k}{\partial w} \quad (2)$$

Computer Science Department
Faculty of Informatics, Mahasarakham University

โดยที่

$$\frac{\partial h_t}{\partial h_k} = \prod_{i=k+1}^{t} \frac{\partial h_i}{\partial h_{i-1}} \tag{3}$$

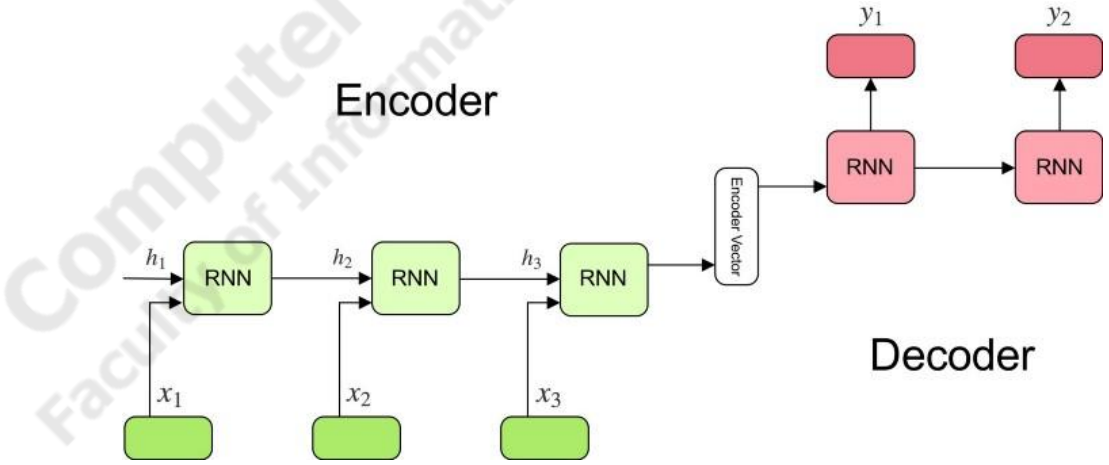
ซึ่งพอเอามารวมกัน จะได้ว่า

$$\frac{\partial E}{\partial w} = \sum_{k=1}^{t} \frac{\partial E_t}{\partial y_t} \frac{\partial y_t}{\partial h_t} \left(\prod_{i=k+1}^{t} \frac{\partial h_i}{\partial h_{i-1}} \right) \frac{\partial h_k}{\partial w} \tag{4}$$

จะเห็นว่าการคำนวณ Gradient ของ loss E นั้น จะต้องอาศัยการคูณกันของ Derivation หลายๆตัวซึ่งนั่นก็จะได้ข้อเสียมาแล้วว่า ถ้า Gradient มีค่าน้อย (น้อยกว่า 1) การคูณต่อเนื่องยาวๆแบบนี้ก็จะทำให้ตัว Gradient มันก็จะลดไปเรื่อย ๆตามความยาว Sequence หรือสรุปได้ว่า RNN ยังมีปัญหากับข้อมูลที่ Sequence มีขนาดยาวเกินไป

2.2.5 Sequence-to-Sequence (Seq2Seq)

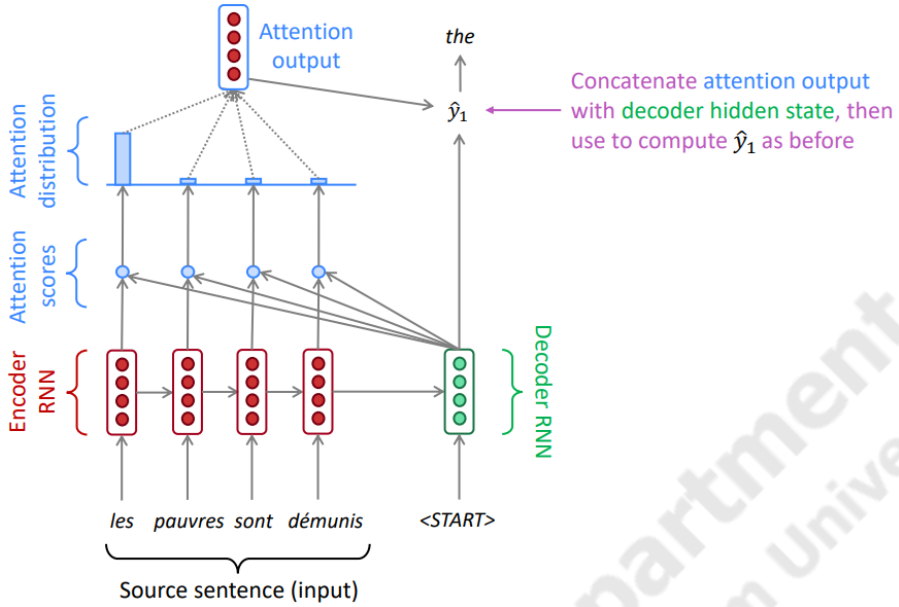
Sequence-to-Sequence (Seq2Seq) [8] หรือเรียกอีกชื่อว่า RNN Encoder-Decoder ซึ่งโมเดลนี้จะแบ่งเป็นสองส่วน เรียกว่า Encoder กับ Decoder โดยส่วน Encoder จะรับ Input เข้ามาทีละหน่วยผ่านทาง RNN และเก็บสะสม Information ที่จำเป็นไว้ จากนั้นจะผ่าน Information นี้ไปยังส่วน Decoder ซึ่งก็จะเป็น RNN อีกตัวหนึ่งที่ให้ Output ออกมาทีละหน่วย โดยดูจาก Information ที่ได้รับมา และ Output ตัวก่อนหน้า ซึ่งแผนภาพของ Seq2Seq จะแสดงได้ดังนี้



รูปภาพประกอบที่ 2.3 การทำงานของ (Seq2Seq)

โมเดล Seq2Seq จะมีปัญหาเกิดขึ้น คือการส่ง Information เป็นทอดๆ ตามสายยาวแบบนี้ อาจจะมี Information ที่จำเป็นบางอย่างสูญหายไประหว่างทางได้ เรียกว่า Attention

Computer Science Department
Faculty of Informatics, Mahasarakham University



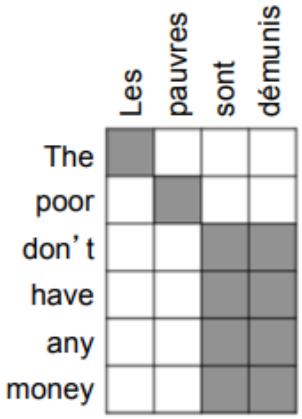
รูปภาพประกอบที่ 2.4 โครงสร้างของ Attention

การทำ Attention สำหรับ Seq2Seq สามารถแสดงได้ดังรูปด้านบนนี้ โดยเมื่อต้องการจะคำนวณ Output ที่ตำแหน่งหนึ่ง ก็จะนำ Vector ของ Decoder (q) ณ ตำแหน่งนั้น มาใช้หา Attention Score กับ Vector ของ Encoder (p) ในทุกตำแหน่ง ซึ่งถ้า Score ที่ Encoder ตำแหน่งไหนสูง หมายความว่าจะให้ความสำคัญ หรือใส่ใจกับตำแหน่งนั้นมาก การคำนวณค่านี้ก็ทำได้หลายวิธี โดยวิธีที่ง่ายที่สุดก็คือการทำ Dot Product ระหว่าง p กับ q เลย ซึ่งหมายความว่าจะใส่ใจกับตำแหน่งที่มีค่า p ใกล้เคียงกับค่า q และเมื่อได้ค่า Score ออกมาแล้ว ก็จะเอาเข้าฟังก์ชัน Softmax เพื่อแปลงเป็นค่าความน่าจะเป็น ซึ่งค่านี้จะเปรียบเสมือน Weight สำหรับ p ต่าง ๆ จากนั้นก็จะทำการหา Weight Average ของ p ออกมาเป็น Vector เดียว (r) เพื่อนำไปใช้ในการคำนวณ Output ต่อไป
เขียนสมการได้ดังนี้

$$r = \sum_i \frac{e^{pi \cdot q}}{\sum_j e^{pj \cdot q}} pi \qquad (5)$$

Attention นี้นอกจากจะแก้ปัญหาดังกล่าวที่มาแล้ว ยังสามารถแก้ปัญหา Vanishing Gradient ไปด้วยพร้อมกัน นอกจากนั้น Output ตำแหน่งต่าง ๆ ให้ความใส่ใจกับ Input ที่ตำแหน่งใด ก็เท่ากับว่าเราได้ Alignment มาฟรีๆ ซึ่งการทำ Alignment นี้ ในกรรมวิธี Machine Translation แบบดั้งเดิม ที่ไม่ใช่ Neural Machine Translation ถือว่าเป็นปัญหาสำคัญ

Computer Science Department
Faculty of Informatics, Mahasarakham University

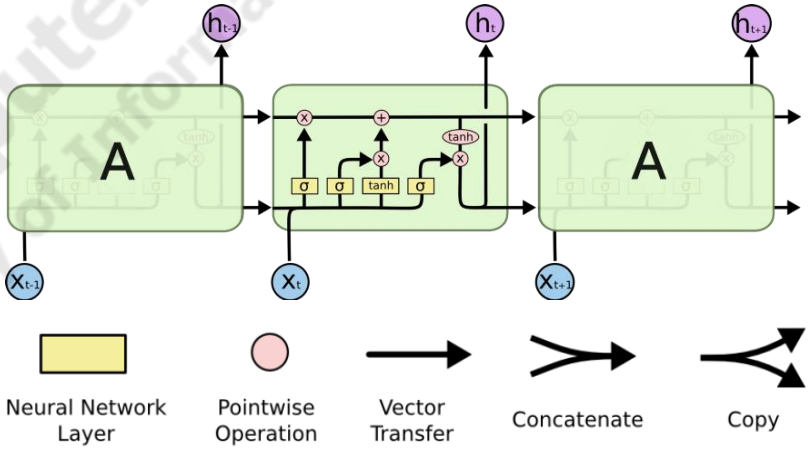


รูปภาพประกอบที่ 2.5 Neural Machine Translation

เมื่อใช้ Attention เข้ามาช่วยแล้ว ได้ผลลัพธ์ที่ดีกว่าเมื่อไม่ใช้แทบจะแน่นอน ปัจจุบัน Attention จึงเป็นเหมือนท่าบังคับพื้นฐานของการทำ Sequence-To-Sequence Learning

2.2.6 Long Short-Term Memory (LSTM)

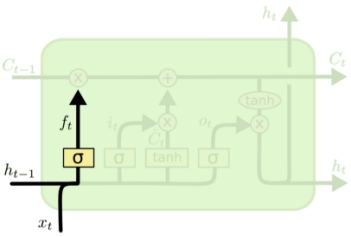
Long Short-Term Memory (LSTM) [9] เป็นเครือข่ายประสาทที่เกิดซ้ำซึ่งได้รับการแก้ไขซึ่งช่วยให้จดจำข้อมูลที่ผ่านมาในหน่วยความจำได้ง่ายขึ้น ปัญหาการไล่ระดับสีที่หายไปของ RNN ได้รับการแก้ไขแล้วที่นี่ LSTM เหมาะอย่างยิ่งในการจำแนกประมวลผลและคาดการณ์อนุกรมเวลาตามระยะเวลาที่ไม่ทราบระยะเวลา มันฝึกโมเดลโดยใช้การขยายพันธุ์ย้อนกลับ ในเครือข่าย LSTM ดังนี้



รูปภาพประกอบที่ 2.6 โครงสร้างของ Long Short-Term Memory

ในส่วนแรกจะมี Sigmoid Layer ซึ่งให้ค่าออกมาระหว่าง 0 กับ 1 จะได้ค่าออกมา ซึ่งจะนำไปใช้ในการคูณกับ State ก่อนในภายหลัง ซึ่งเป็นการปรับการใช้ State เก่า

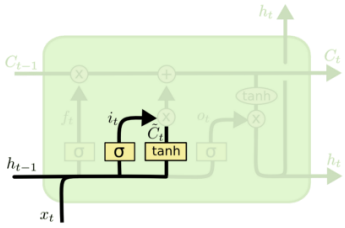
Computer Science Department
Faculty of Informatics, Mahasarakham University



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

รูปภาพประกอบที่ 2.7 การทำงานของ (LSTM)

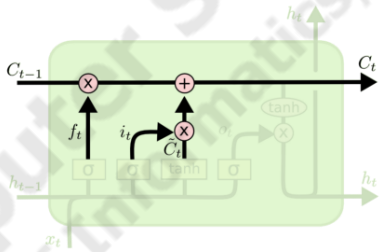
ในส่วนถัดไปเราก็จะมีการคำนวณ (ในส่วนของ tanh) แล้วนำค่านั้นไปคูณกับค่าที่ได้จาก Sigmoid Layer เพื่อตั้งค่า Weight ในข้อมูลใหม่



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

รูปภาพประกอบที่ 2.8 การคำนวณเพื่อตั้งค่า Weight

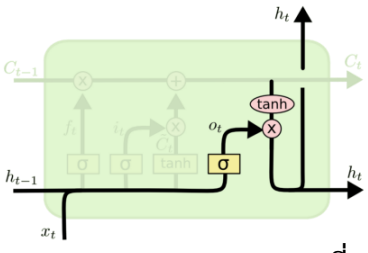
จากนั้นเราก็นำสองส่วนมารวมกัน ให้ลืมค่าเก่าบางส่วน และรับบางส่วนจากของใหม่มา จะได้ค่า Cell State



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

รูปภาพประกอบที่ 2.9 การคำนวณค่า Cell State

จากนั้นเราก็นำค่า Cell State คำนวณ (tanh) และนำค่าที่ได้มาคูณกับค่าจาก Sigmoid Layer เพื่อตั้ง Weight ให้อีกครั้ง และจะได้ออกมาเป็นค่า h(t)



$$o_t = \sigma\left(W_o \; [h_{t-1}, x_t] \; + \; b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

รูปภาพประกอบที่ 2.10 การคำนวณค่า h(t)

Computer Science Department
Faculty of Informatics, Mahasarakham University

2.3 ระบบงานที่เกี่ยวข้อง

ผลงานวิจัยของอนุรุธ เลิศปิยะ และคณะ เรื่อง “Thai Spelling Correction and Word Normalization on Social Text Using a Two-Stage Pipeline With Neural Contextual Attention” [10] ถูกนำมาใช้เพื่อปรับปรุงคุณภาพของข้อความในคอมพิวเตอร์โดยการตรวจจับและแก้ไขข้อผิดพลาด การทำงานของระบบแก้ไขข้อความในปัจจุบันในการแก้ไขข้อผิดพลาดและความแปรปรวนของคำในตำราสังคมไทย และเสนอวิธีการที่ออกแบบมาสำหรับงานนี้ เราพบว่าระบบแก้ไขข้อความภาษาไทยที่มีให้บริการในปัจจุบันนั้นมีประสิทธิภาพไม่เพียงพอสำหรับการแก้ไขข้อผิดพลาดการสะกดคำและความแปรปรวนของคำ ในขณะที่ตัวแก้ไขข้อความที่ออกแบบมาสำหรับการแก้ไขข้อผิดพลาดทางไวยากรณ์ภาษาอังกฤษประสบปัญหาการแก้ไขมากเกินไป

ดังนั้นจึงเสนอตัวแก้ไขข้อความที่ใช้ระบบประสาทด้วยโครงสร้างสองขั้นตอนเพื่อบรรเทาปัญหาของการแก้ไขมากเกินไปในขณะที่ใช้ประโยชน์จากตัวแก้ไข Seq2Seq ของระบบประสาท วิธีการนี้ประกอบด้วยตัวตรวจจับข้อผิดพลาดจากระบบประสาทและตัวแก้ไขข้อผิดพลาดของระบบประสาท Seq2Seq พร้อมความสนใจตามบริบท สถาปัตยกรรมใหม่นี้ช่วยให้เครือข่าย Seq2Seq สร้างการแก้ไขโดยยึดตามข้อความที่ผิดพลาดและบริบทโดยไม่จำเป็นต้องใช้โครงสร้างแบบ End-To-End วิธีการของเรามีประสิทธิภาพเหนือกว่าระบบแก้ไขข้อความอื่น ๆ ที่ได้รับการประเมินทั้งหมด เมื่อเปรียบเทียบกับผลลัพธ์ที่ดีที่สุดเป็นอันดับสอง (Copy-Augmented Transformer) วิธีการนี้ลดอัตราข้อผิดพลาดของคำ (WER) จาก 2.51% เป็น 2.07% ปรับปรุงคะแนนความเข้าใจการประเมินภาษาทั่วไป (GLEU) จาก 0.9409 เป็น 0.9502 สำหรับภาษาไทย งานแก้ไขข้อความและปรับปรุงคะแนน GLEU จาก 0.7409 เป็น 0

ผลงานวิจัยของจตุพร อินพานิช และวัชรพล นิลทัย เรื่อง “การแก้ไขข้อความที่ผิดพลาด” งานวิจัยนี้ได้นำเสนอ เทคโนโลยี OCR สำหรับภาษาไทย ได้มีการพัฒนาทำให้มีประสิทธิภาพสูงขึ้น จากนั้นใช้การตรวจสอบคำผิด โดยแยกตามลักษณะการตัดคำ [11] ได้แก่ การตัดคำโดยใช้พจนานุกรม การตัดคำโดยใช้กฎ และการตัดคำด้วย Deep Cut พบว่าการตัดคำโดยใช้พจนานุกรม มีความถูกต้อง 72.97% การตัดคำโดยใช้กฎ มีความถูกต้อง 80.95% และการตัดคำโดยใช้ Deep Cut มีความถูกต้อง 100%

ผลงานวิจัยของอรทัย คงธรรม และคณะ เรื่อง “Text-based LSTM Networks for Automatic Thai Love Quotes Generation on Twitter” งานวิจัยนี้ได้นำเสนอ โมเดล LSTM และปรับแก้ พารามิเตอร์ในการทำนายคำและเลือกคำโดยอัตโนมัติ [12] สร้างข้อความบน Twitter อยู่ในขั้นตอนการฝึกโมเดล ทดลองกับชุดข้อมูล 2 รูปแบบรวมถึง Word แปลงเป็นจำนวนเต็มและแปลงคำเป็นเวกเตอร์ LSTM+Word2Vec ที่ป้อน 3 คำสามารถแสดงค่าเฉลี่ยต่ำได้ การสูญเสียและอัตราการ

Computer Science Department
Faculty of Informatics, Mahasarakham University

แก้ไขการแปลที่กำหนดเป้าหมายโดยมนุษย์สามารถแสดงต่ำ อัตราการแก้ไขเฉลี่ยเช่นกัน ถึงแม้ว่าคำว่า Corpus That ใช้สำหรับบทความนี้ไม่ใหญ่แต่เมื่อเราวิเคราะห์ผลลัพธ์ ของการประเมินเป็นคะแนนที่ยอมรับได้ ความรัก คำพูดที่สร้างโดยแบบจำลองของเราสามารถอ่านและเข้าใจได้ โดยมนุษย์ งานวิจัยเกี่ยวกับการสร้างข้อความภาษาไทยอัตโนมัติคือ ยังเล็กอยู่โดยเฉพาะข้อความในด้านศิลปะ เราจะใช้ LSTM และ Word2Vec กับแต่งกลอนไทยสั้นในอนาคต

ผลงานวิจัยของฮุ่ยเหวินหวู่ และคณะ เรื่อง “Thai Word Segmentation Based On Sequence-to-Sequence Model” งานวิจัยนี้ได้นำเสนอแบบอแบบจำลองการแบ่งส่วนคำภาษาไทยแบบเรียงต่อกันที่มีประสิทธิผลมากขึ้น โมเดลนี้ใช้หน่วยความจำระยะสั้นแบบสองทิศทางและโครงข่ายประสาทเทียมแบบ Gated Recurrent [13] สำหรับเข้ารหัสอินพุตและถอดรหัสเอาต์พุตตามลำดับ นอกจากนี้ยังได้รับการประเมินใน InterBEST 2009 Thai Corpus และบรรลุผลลัพธ์ที่เหมาะสมกับชุดข้อมูลสี่ชุดในสาขาต่าง ๆ ผลการทดลองแสดง ให้เห็นว่าโมเดลนั้นใช้ง่าย มีประสิทธิภาพ และมีผลกับโดเมนที่แข็งแกร่ง ซึ่งเป็น suiTab มากกว่าสำหรับภาษาชนกลุ่มน้อยที่มีข้อมูลน้อย เราพบว่าเอนทิตี้มีอิทธิพลบางอย่างต่อผลกระทบของการแบ่งส่วนคำ ในอนาคต พิจารณาการแบ่งส่วนคำด้วยเอนทิตี้ในประโยค

Computer Science Department
Faculty of Informatics, Mahasarakham University