บทที่ 6

สรุปผลการวิจัยและข้อเสนอแนะ

6.1 สรุปผลการวิจัย

งานวิจัยนี้เริ่มจากแนวคิดที่อยากให้การติดต่อกับคอมพิวเตอร์ง่ายเหมือนการติดต่อกับมนุษย์ด้วยกันเอง แต่การโต้ตอบกับผู้ใช้ (User Interface) ระหว่างมนุษย์กับคอมพิวเตอร์ในปัจจุบันนั้นยังมีความแตกต่างจากการสื่อสารด้วยกันระหว่างมนุษย์เองค่อนข้างมาก มนุษย์ยังต้องติดต่อหรือสั่งให้คอมพิวเตอร์ทำงานด้วยคำสั่งในรูปแบบที่คอมพิวเตอร์จะสามารถเข้าใจได้

การประมวลผลภาษาธรรมชาติ (Natural Language Processing) เป็นเทคนิคหนึ่งที่ถูกนำมาใช้งานเพื่อการสื่อสารที่ดีขึ้นระหว่างมนุษย์และคอมพิวเตอร์ การประมวลผลภาษาธรรมชาติคือกระบวนการที่ทำให้ผู้ใช้คอมพิวเตอร์สามารถนำข้อมูลเข้าเครื่องในรูปแบบภาษาพูดหรือภาษาเขียนที่ผู้ใช้ใช้สื่อสารกับคนทั่วไป คอมพิวเตอร์จะตีความข้อมูลนั้นโดยจำลองวิธีการตีความของมนุษย์ การใช้งานภาษาธรรมชาติมีตัวอย่างเช่น การสั่งงานเครื่องใช้ในบ้านผ่านระบบภาษาธรรมชาติ [1] และการสร้างหุ่นยนต์สนทนาอัตโนมัติ [2], [3] เป็นต้น

หลังจากได้ศึกษาเทคนิคที่ใช้ทำหุ่นยนต์สนทนาหลายๆ เทคนิคแล้ว งานวิจัยนี้เลือกใช้ปัญญาประดิษฐ์ที่ชื่อ A.L.I.C.E Bot ซึ่งเป็นปัญญาประดิษฐ์ที่มีมาตรฐานสูงและใช้กันอย่างแพร่หลาย แต่มีข้อเสีย คือ การจับคู่ฐานความรู้เอไอเอ็มแอล นั้นจะใช้การจับคู่แบบแม่นตรง (Exact matching) ส่งผลให้เมื่อมีข้อความอินพุตผิดพลาดจากการพิมพ์เพียงเล็กน้อยก็จะไม่สามารถโต้ตอบได้ โดยการที่หุ่นยนต์สนทนาไม่สามารถคาดเดาคำผิดได้เช่นเดียวกับมนุษย์นั้นอาจก่อให้เกิดความรำคาญกับผู้ใช้งาน แทนที่จะช่วยให้การติดต่อกับคอมพิวเตอร์นั้นเป็นไปตามธรรมชาติมากขึ้น

งานวิจัยนี้จึงได้เสนออัลกอริทึมการแก้ไขคำผิดแบบไม่ตั้งใจโดยอัตโนมัติในภาษาไทย เพื่อนำไปใช้กับหุ่นยนต์สนทนา เนื่องจากหุ่นยนต์สนทนาที่ใช้เอไอเอ็มแอลเป็นฐานความรู้จะใช้วิธีการจับคู่แพทเทิร์นกฎคำถาม-คำตอบ หากมีการป้อนอินพุตที่มีความผิดพลาดเพียงเล็กน้อยก็จะทำให้ไม่สามารถโต้ตอบได้ ทั้งที่ความผิดพลาดของคำผิดในประโยคอินพุตที่ป้อนเข้าไปนั้นมนุษย์สามารถคาดเดาได้ จึงทำการหาองค์ประกอบมาช่วยในการแก้ไขคำผิด โดยได้มีการทดสอบเพื่อสังเกตพฤติกรรมในการพิมพ์ ทำให้ได้มาซึ่งรูปแบบการพิมพ์ผิดในภาษาไทยที่เหมาะสมกับโดเมน (Domain) ของการสนทนาผ่านการพิมพ์ด้วยแป้นพิมพ์ของคอมพิวเตอร์ เนื่องจากจะนำสถิติที่ได้ไปใช้ในการออกแบบสร้างอัลกอริทึมการแก้ไขคำผิดที่เหมาะสมกับหุ่นยนต์สนทนาต่อไป

ดังนั้นในงานวิจัยนี้ จึงได้แบ่งการทดลอง 2 ส่วนด้วยกัน คือ ส่วนแรกนั้นทำการหารูปแบบการพิมพ์ผิดในภาษาไทย เพื่อนำมาใช้ออกแบบอัลกอริทึม และส่วนที่สอง เป็นการทดสอบ

อัลกอริทึมที่ได้ออกแบบไว้ (ซึ่งถูกนำมารวมเข้ากับหุ่นยนต์สนทนาที่แก้ไขให้สามารถทำงานกับภาษาไทยและยืดหยุ่นต่อคำผิดลงไป) ซึ่งผลการทดลองในแต่ละส่วนสามารถสรุปได้ ดังนี้

จากการทดลองในบทที่ 3 พบว่าในภาษาไทยนั้นคำผิดส่วนใหญ่เกิดจากความผิดพลาดทั้ง 4 กรณีประกอบกันซึ่งคิดเป็น 93.54 เปอร์เซ็นต์จากคำผิดที่พบทั้งหมด นอกจากนี้ยังพบว่าความผิดพลาดทั้ง 4 กรณีนั้น สามารถนำเรียงลำดับความสำคัญตามปริมาณที่พบได้ดังนี้คือ แทนที่>เกิน>ตก>สลับ

และไม่ว่าจะเป็นกรณีความผิดพลาดใดๆ ตำแหน่งอักขระที่ผิดนั้นมักจะอยู่ตรงกลางแต่ค่อนไปทางด้านหลังเล็กน้อย คิดเป็นค่าเฉลี่ยของทุกกรณีได้เป็นตำแหน่งที่ 58.36 เปอร์เซ็นต์ของความยาวคำ

จากการทดลองในบทที่ 5 ซึ่งได้ทำการทดสอบประสิทธิภาพในการจับคู่แพทเทิร์นด้วยบทสนทนาที่มีคำผิดแบบไม่ตั้งใจทั้งหมดจำนวน 120 ประโยค พบว่าหุ่นยนต์สามารถตอบได้คิดเป็น 95 เปอร์เซ็นต์ แสดงให้เห็นว่าหุ่นยนต์สามารถทำงานได้มากขึ้นจริงหากใช้อัลกอริทึมแก้คำผิดที่ได้งานวิจัยนี้

ปัจจัยสำคัญที่ส่งผลต่อประสิทธิภาพการจับคู่แพทเทิร์นก็คือ การตัดคำหรือแบ่งขอบเขตคำ ถ้าหากโปรแกรมตัดคำที่เลือกใช้แบ่งขอบเขตของคำผิดพลาดตั้งแต่ต้น ก็จะส่งผลให้การแก้ไขคำผิดนั้นมีความผิดพลาดตามไปด้วย

6.2 ปัญหาและข้อจำกัดที่พบจากการวิจัย

การแก้ไขคำผิดในงานวิจัยนี้ใช้วิธีการค้นหาในพจนานุกรม (Dictionary Lookup) ซึ่งจัดเป็นวิธีการในกลุ่มของการแก้ไขคำผิดโดยไม่คำนึงถึงสภาพรอบข้าง (Isolated-word Error Correction) ดังนั้นจึงไม่สามารถตรวจสอบกรณีต่อไปนี้ได้ เช่น

- คำผิดที่เกิดจากการสลับอักขระระหว่างคำ เช่น **ตัดแขน** พิมพ์เป็น **ตัแดขน**
- คำผิดที่สามารถแบ่งได้เป็นคำย่อยที่ถูกได้ 2 คำ เช่น เที่ยงธรรม พิมพ์ผิด เป็น เที่ยวธรรม

สาเหตุที่ไม่สามารถแก้คำว่า **ตัแดขน** ได้นั้นเนื่องจาก โปรแกรมตัดคำที่เลือกใช้จะตัดคำนี้ออกเป็น 2 คำย่อย คือ **ตัแ+ดขน** ซึ่งอัลกอริทึมในงานวิจัยนี้ไม่ได้มุ่งเน้นการแก้คำผิดระหว่างคำ ทำให้เมื่อลองแก้คำผิดภายในขอบเขตของแต่ละคำแล้ว ก็อาจจะไม่ได้ผลลัพธ์ที่ต้องการออกมา (คือ ไม่ได้คำว่า **ตัด** และคำว่า **แขน** คืนมาดังเดิม เพราะขอบเขตของงานวิจัยนี้จะทำการแก้คำผิดภายในคำที่ได้ผลลัพธ์มาจากโปรแกรมตัดคำเท่านั้น)

ดังนั้น ข้อความภาษาไทยที่มีความหมายในพจนานุกรมเท่านั้นจะจัดว่าเป็นคำถูก และคำที่ไม่พบในพจนานุกรมจะถือว่าเป็นคำผิด (แต่ในงานวิจัยนี้จะเน้นแก้ไขเฉพาะคำผิดแบบไม่ตั้งใจเท่านั้น) และงานวิจัยนี้จะไม่ครอบคลุมถึงการพิมพ์ผิดแบบพิมพ์ตก

การแก้ไขคำผิดกรณีที่เกิดจากกรณีพิมพ์ตกนั้นจำเป็นที่จะต้องใช้โครงสร้างข้อมูลทรัยในการค้นหารายการคำใกล้เคียงคำผิด แต่ว่าโครงสร้างข้อมูลแบบทรัยนั้นต้องการใช้ทรัพยากรหน่วยความจำที่สูงระดับกิกะไบต์ (Gigabyte) ขึ้นไป หากเครื่องคอมพิวเตอร์ที่นำใช้นั้นมีประสิทธิภาพต่ำจะทำให้ช้าในการตอบสนองกับผู้ใช้

วิธีการลองแทนที่คำผิดด้วยอักขระข้างเคียงยังพบปัญหา คือ ยังคงเกิดกรณีที่ให้ผลลัพธ์มากกว่า 1 คำขึ้นได้ ทำให้ยากต่อการตัดสินใจเลือกว่าคำไหนสำคัญมากกว่ากัน สำหรับการแก้ไขคำผิดอัตโนมัติที่จะนำไปใช้กับหุ่นยนต์สนทนานั้น จะนำผลลัพธ์ที่ได้จากการแทนที่ทั้งหมดไปรวมกับคำอื่นในประโยคแล้วลองจับคู่ทุกกรณีผลคูณคาร์ทีเซียน ซึ่งคำอื่นๆ ที่อยู่ข้างเคียงในประโยคจะช่วยตัดสินใจแก้ปัญหาในเรื่องนี้ได้ (เสมือนกับว่าได้นำ context เข้ามาช่วยพิจารณา)

เมื่อทำการวิเคราะห์กรณีที่หุ่นยนต์ไม่สามารถตอบได้คิดเป็น 5 เปอร์เซ็นต์นั้น พบว่าสามารถแบ่งออกเป็น 2 สาเหตุ คือ 1.พจนานุกรมที่ใช้ยังไม่มีคำสมาส 2.ตัวอักษรที่พิมพ์ผิดนั้นไม่ได้อยู่ใกล้เคียงในระยะ 1 ปุ่มรอบตัวอักษรเดิมที่ควรจะเป็น ซึ่งรายละเอียดและตัวอย่างได้แสดงไว้แล้วในหัวข้อ 5.4

6.3 ข้อเสนอแนะ

โครงสร้างข้อมูลทรัย (Tries) มีคุณสมบัติเด่นในด้านการค้นหารายการคำใกล้เคียงคำผิดที่เกิดจากลักษณะการพิมพ์ตกได้ดี [20] ถ้าหากนำวิธีการที่นำเสนอในบทความนี้ไปรวมเข้าด้วยกันกับทรัย โดยให้ทรัยทำหน้าที่ค้นหารายการคำใกล้เคียงคำผิด แล้วนำวิธีการใหม่นี้ไปช่วยจัดลำดับรายการคำใกล้เคียง จะทำให้ได้ระบบการตรวจสอบหรือแก้ไขคำผิดที่ครอบคลุมคำผิดที่เกิดจากหลายลักษณะได้มากขึ้น

การตัดคำที่ดีทำให้สามารถระบุขอบเขตของความผิดพลาดได้ถูกต้อง และนำไปสู่ความสามารถในการแก้ไขคำผิดได้ดีกว่า ดังนั้นโปรแกรมตัดคำที่ใช้จึงมีผลต่อการทำงานของอัลกอริทึมแก้ไขคำผิดเป็นอย่างมาก แต่ถ้าหากมีโปรแกรมตัดคำตัวอื่นที่สามารถระบุขอบเขตของคำที่มีการพิมพ์ผิดในประโยคได้ดีกว่าที่ใช้อยู่ในงานวิจัยนี้ [5] ก็สามารถนำมาใช้แทนได้ ซึ่งอาจทำให้แก้ไขคำผิดได้ดีขึ้น

วิธีการแก้คำผิดแบบไม่ตั้งใจที่ได้จากงานวิจัยนี้ สามารถนำไปประยุกต์ต่อยอดใช้ให้เกิดประโยชน์ได้อย่างมาก เช่น นำไปรวมกับวิธีแก้คำผิดจากสาเหตุอื่น (เช่น คำผิดที่มีระยะแก้ไขเกิน 1 ตัวอักษร) เพื่อนำไปใช้กับงานด้านโปรแกรมประมวลคำ (Word Processing) ได้