

งอนหรือเปล่า? Are You Mad?

โปรแกรมอรรถประโยชน์เพื่อใช้ในการเอาตัวรอด

ปฏิภาณ ฝอยทอง

วิชา 968-356 NLP ระดับการศึกษาปริญญาตรี สาขาการคอมพิวเตอร์ คณะวิทยาลัยการคอมพิวเตอร์ วิทยาเขตภูเก็ต

ติดต่อผู้เขียน: ปฏิภาณ ฝอยทอง Facebook: ปฏิภาณ ฝอยทอง Email: s6630611006@phuket.psu.ac.th

บทคัดย่อ

โปรแกรม "งอนหรือเปล่า?" (Are You Mad?) เป็นระบบที่ใช้เทคโนโลยีประมวลผลภาษาธรรมชาติ (NLP) เพื่อตรวจจับว่าข้อความของคู่สนทนาเป็นการงอนหรือประชดประชันหรือไม่ โดยอาศัยการเรียนรู้จากชุดข้อมูลตัวอย่างและแบบจำลองปัญญาประดิษฐ์ ระบบสามารถวิเคราะห์โทนเสียงจากข้อความ ตรวจจับรูปแบบการใช้คำ และระบุแนวโน้มของอารมณ์ในบทสนทนาได้ ช่วยให้สามารถเข้าใจบริบททางอารมณ์ของคู่สนทนาได้แม่นยำมากขึ้น เหมาะสำหรับการใช้งานในแอปพลิเคชันแชท การวิจัยพฤติกรรมทางสังคม และการพัฒนาปัญญาประดิษฐ์ด้านอารมณ์

คำสำคัญ: NLP, การประมวลผลภาษาธรรมชาติ, การตรวจจับอารมณ์, การประชดประชัน, การงอน, การวิเคราะห์ข้อความ, ปัญญาประดิษฐ์

การรับ Input ในโปรแกรม

โปรแกรม "งอนหรือเปล่า?" (Are You Mad?) จะรับข้อความจากคู่สนทนา เพื่อนำมาวิเคราะห์ว่าข้อความนั้นเป็นการงอน หรือ การประชดประชัน หรือไม่ รูปแบบ Input

1. **ข้อความเดี่ยว (Single Sentence Input)** ผู้ใช้ป้อนข้อความที่ต้องการให้ระบบวิเคราะห์ เช่น
 - "อ้อออ ไม่เป็นไร เรามันไม่สำคัญอยู่แล้วนี่เนอะ"
 - "เธอไปเที่ยวกับเพื่อนสนุกมากเลยสินะ ลืมเราไปเลย"
2. **ชุดข้อความจากการสนทนา (Conversation Input)** รับบทสนทนา 2 ฝ่าย เพื่อให้ AI วิเคราะห์จากบริบท เช่น
 - ฝ่าย A: "วันนี้ไปดูหนังกันไหม?"
 - ฝ่าย B: "ไม่เป็นไรหรอก เธอคงอยากไปกับคนอื่นมากกว่านี้" (ประชด)

วิธีการรับ Input **พิมพ์ข้อความลงในระบบ** (ภาพที่ 1) (เช่น ผ่าน Web App, Mobile App หรือ Command Line) **ดึงข้อความจาก API** แชนบอท (เช่น LINE, Messenger) **ใช้ไฟล์ข้อความ** (.txt, .csv) เพื่อวิเคราะห์ข้อความจำนวนมาก และระบบจะนำ Input ไปผ่านกระบวนการ **ทำความสะอาดข้อมูล (Preprocessing)** ก่อนประมวลผลด้วยโมเดล NLP เพื่อให้ได้ผลลัพธ์ที่แม่นยำขึ้น

1. **ตัดคำ (Tokenization)**
 - ใช้เครื่องมือเช่น PyThaiNLP เพื่อแยกคำ
 - ตัวอย่าง: "เธอคงอยากไปกับคนอื่นมากกว่านี้" → ["เธอ", "คง", "อยาก", "ไป", "กับ", "คนอื่น", "มากกว่า", "นี้"]
2. **ลบคำที่ไม่จำเป็น (Stopword Removal)**
 - เช่น คำว่า "ก็", "นะ", "ครับ", "ค่ะ" ที่ไม่มีผลต่อความหมายของประโยค

วันนี้ไปกินข้าวกับเราทำไมดูรีบๆ จัง

ก็ไม่ได้รีบนะ แค่มีงานต้องทำตอนนี้นิดหน่อย

อ้ออ้อ งานสำคัญกว่าเราไง เข้าใจละ

ไม่ใช่อย่างนั้นนะ แค่ต้องเคลียร์ให้เสร็จเฉยๆ

ไม่เป็นไรๆ งานเธอสำคัญที่สุดอยู่แล้วนี่เนอะ เรามันแค่ตัวประกอบ

เฮ้ย อย่าคิดแบบนั้นสิ เราก็แค่งานเยอะไปหน่อยเอง

ภาพที่ 1 แสดงตัวอย่างบทสนทนาระหว่างบุคคลสองคน ในกรณีที่มีฝ่ายหนึ่งมีอาการ งอน หรือ ประชดประชัน โดยใช้คำพูดที่แฝงความน้อยใจและเสียสติ ระบบ NLP จะนำข้อความเหล่านี้มาวิเคราะห์เพื่อระบุว่าผู้พูดกำลังมีอาการแบบใด

1. แปลงคำให้เป็นรูปแบบมาตรฐาน (Text Normalization)

- ปรับคำที่ใช้ภาษาพูดให้เป็นรูปแบบมาตรฐาน เช่น
 - i. "ไปละคะ" → "ไปแล้วคะ"
 - ii. "ม่ายยยย" → "ไม่"
 - iii. "5555" → "[หัวเราะ]"

2. ลบอักขระพิเศษและอีโมจิ (Special Character & Emoji Removal)

- เช่น "😞", "😂", "😏" ที่อาจมีผลต่อความหมายแต่ไม่ได้ใช้ในการฝึกโมเดล

3. วิเคราะห์บริบทของประโยค (Contextual Analysis)

- ตรวจสอบคำที่มักใช้ในการประชด เช่น "อ้ออ้อ", "เหออ", "สุดยอดไปเลย"
- วิเคราะห์การใช้เครื่องหมายวรรคตอน เช่น
 - i. "อ้ออ้อ" (ประชด)
 - ii. "อ้อ" (ปกติ)

หลังจากทำ Preprocessing ข้อมูลที่ได้จะถูกนำไปใช้กับโมเดล NLP เพื่อวิเคราะห์ว่าประโยคนั้น เป็นการงอนหรือประชดประชันหรือไม่

การวิเคราะห์ด้วยโมเดล (Text Classification)

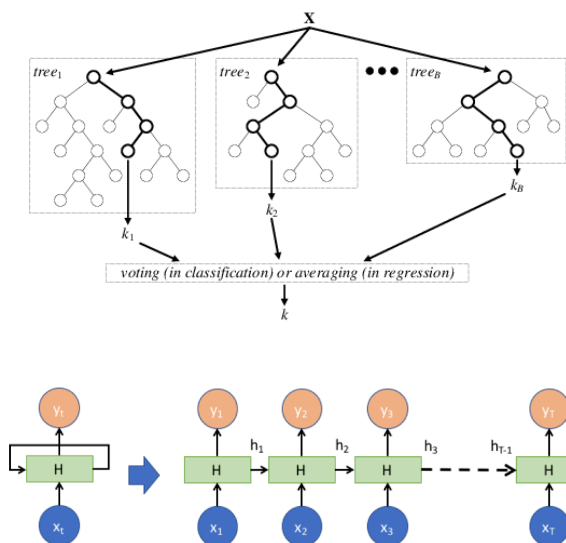
กระบวนการวิเคราะห์ว่าข้อความนั้นเป็น การงอน, การประชดประชัน, หรือ ข้อความปกติ จะใช้ เทคนิค NLP และ Machine Learning โดยมีขั้นตอนดังนี้:

1. การแปลงข้อความเป็นเวกเตอร์ (Text Vectorization)

- โมเดล NLP ไม่สามารถเข้าใจข้อความดิบได้โดยตรง จำเป็นต้องแปลงข้อความให้อยู่ในรูปของตัวเลขก่อน โดยใช้วิธีต่างๆ เช่น:
 - i. TF-IDF (Term Frequency - Inverse Document Frequency): นับความสำคัญของคำ
 - ii. Word Embeddings (เช่น Word2Vec, FastText, BERT): แปลงคำให้เป็นเวกเตอร์ที่มีความหมาย

2. การเลือกโมเดลวิเคราะห์ข้อความ สามารถใช้โมเดลได้หลายแบบตามความซับซ้อนที่ต้องการ

- โมเดลพื้นฐาน (Traditional ML Models) โมเดลเหล่านี้ประมวลผลได้เร็วและไม่ต้องใช้พลังประมวลผลสูง เช่น:
 - i. Naïve Bayes (NB) – เหมาะกับปัญหาการจำแนกข้อความ
 - ii. Support Vector Machine (SVM) – ทำงานได้ดีในข้อมูลที่มีขนาดเล็ก
- โมเดลเชิงลึก (Deep Learning) โมเดลที่สามารถจับบริบทของภาษาได้ดีขึ้น เช่น:
 - i. LSTM (Long Short-Term Memory) – จัดจำบริบทของคำในลำดับข้อความ
 - ii. BERT (เช่น wangchanBERTa) – โมเดลที่เข้าใจบริบทของภาษาไทย



ภาพที่ 2 ใช้ Machine Learning เช่น SVM, Random Forest หรือ Deep Learning เช่น LSTM, Transformers ถ้าใช้โมเดลภาษาไทย อาจลอง wangchanBERTa หรือ pythainlp

- การวิเคราะห์บริบทเพิ่มเติมนอกจากวิเคราะห์ตัวข้อความ
โมเดลอาจใช้ คุณสมบัติพิเศษ (Features Engineering) เช่น:
 - การใช้ เครื่องหมายวรรคตอน มากเกินไป ("อ้อออ อ้อ!!!") → มีแนวโน้มประชด
 - ความยาวของข้อความ ("อ้อออออ ไม่ใช่ไรเลยจ้า าวา" ยาวผิดปกติ) → อาจเป็นประชด
 - การใช้คำที่บ่งบอกอารมณ์ ("สุดยอดไปเลย", "ดีใจมาก เลยนะ", "เหรออ") → อาจเป็นประชด
- การคำนวณค่าความมั่นใจของโมเดล โมเดลจะให้ค่าความน่าจะเป็นว่า ข้อความนั้นเป็นอารมณ์แบบไหน เช่น:

ข้อความ	จน %	ประชด %	น้อยใจ %	ปกติ %
"อ้อออ ไม่ใช่ไร เรามันไม่สำคัญอยู่แล้ว"	5%	60%	35%	0%
"โอเคๆ ไม่ต้องมากก็ได้"	50%	30%	15%	5%
"เห็นเพื่อนไปกับคนอื่น ไม่ชวนเลยนะ"	40%	20%	35%	5%

การแสดงผลลัพธ์ Output

- ผลลัพธ์การวิเคราะห์ (Predicted Label): ระบบจะบอกว่ามีแนวโน้มเป็น "ประชดประชด", "จน", "น้อยใจ", หรือ "ปกติ"
- ค่าความมั่นใจ (Confidence Score): แสดงเปอร์เซ็นต์ความมั่นใจของโมเดล เช่น 75%
- แสดงผลด้วยสีหรือสัญลักษณ์: ถ้าข้อความมีแนวโน้มเป็น ประชด ประชด หรือ จน อาจใช้สีแดงหรือไอคอนเพื่อให้เข้าใจง่ายขึ้น



วันไปกินข้าวกับรอกทำไมดูรีบๆ จัง

ก็ไม่ได้รีบนะ แค่มีงานต้องทำต้องรีบหน่อย

อ้อออ งานสำคัญกว่าเราไง เข้าใจละ

ไม่ใช่อย่างนั้นนะ แค่ต้องเคสรีวิให้เสร็จเฉยๆ

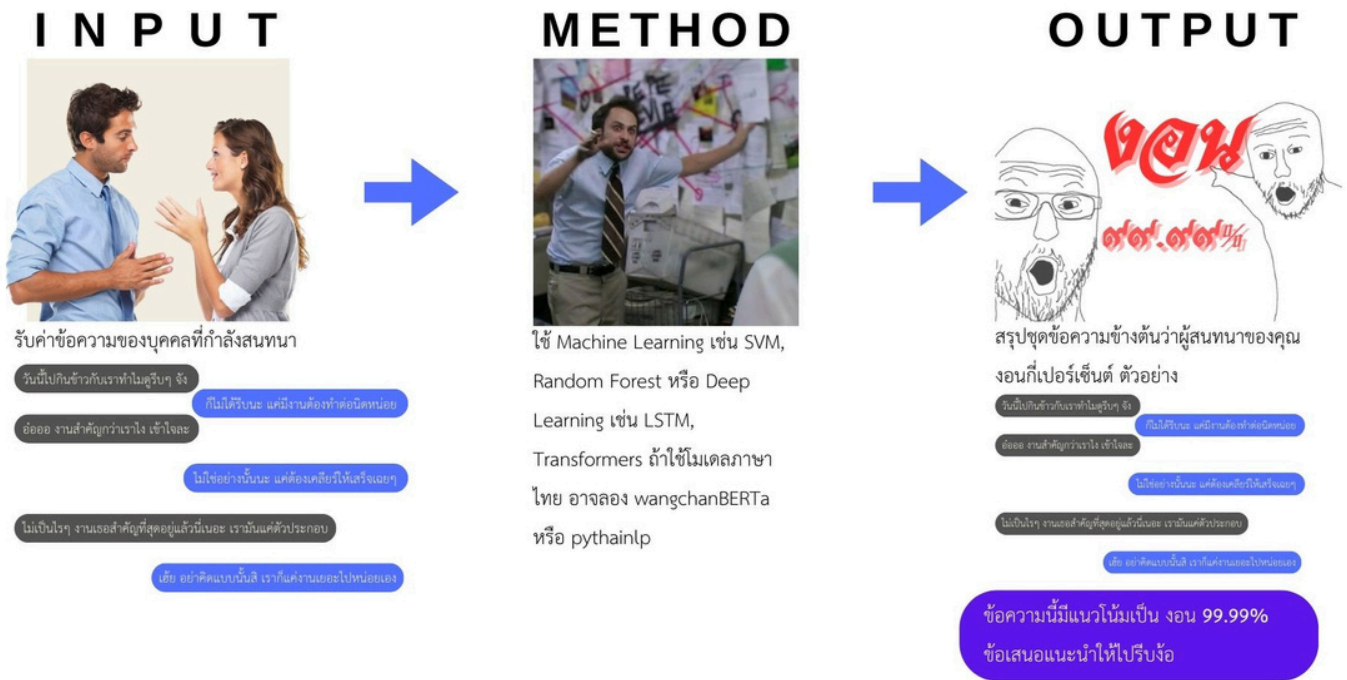
ไม่เป็นไรๆ งานสำคัญที่สุดอยู่แล้วป่ะนะ เรามันแค่ตัวประกอบ

เดี๋ย อ่าคิดแบบนั้นสิ เราก็แค่ทำงานเยอะไปหน่อยเอง

ข้อความนี้มีแนวโน้มเป็น จน 99.99%
ข้อเสนอแนะนำไปปรับจ้อ

ภาพที่ 3 แสดงตัวอย่างผลลัพธ์ของโปรแกรมวิเคราะห์อารมณ์ จากข้อความสนทนา โดยระบบทำการตรวจจับว่า ข้อความที่ บ่อนเข้ามามีแนวโน้มเป็นการจนหรือประชดประชดหรือไม่

จากตัวอย่าง โมเดลจะเลือก "ประชดประชด" เป็นคำตอบที่ดีที่สุด



ภาพที่ 4 ตัวอย่างการทำงานคร่าวๆ รับ input จาก User โดยเป็นประโยคหนึ่งประโยคหรือเป็นชุดข้อความจากนั้นส่งไปให้ Machine Learning หรือ Deep Learning วิเคราะห์โดยแสดงผล output ออกมาเป็นคำพูดหรือข้อความในเชิงความหมายว่ามีแนวโน้มของโอกาสของสิ่งที่จะเกิดและอาจจะเพิ่มคำแนะนำ

Information Extraction (IE)

โครงการเกี่ยวกับการวิเคราะห์ว่าข้อความเป็น "การถอน" หรือ "การประชดประชัน" ได้ดังนี้:

- **Named Entity Recognition (NER)** ตรวจสอบข้อมูลที่เกี่ยวข้องในบทสนทนา เช่น:
 - ชื่อบุคคล: "A", "B"
 - เวลา: "เมื่อวานนี้", "วันนี้"
 - สถานที่: "ที่ร้านกาแฟ"
 - ตัวอย่าง: ข้อความ: "อ้ออ ไม่เป็นไรหรอก เรามันไม่สำคัญอยู่แล้ว" ผลลัพธ์:
 - ประเภทของข้อความ: "ประชดประชัน"
 - อารมณ์: "เชิงลบ"

- **Relation Extraction** ตรวจสอบความสัมพันธ์ในข้อความ
 - ความสัมพันธ์ระหว่างบุคคล
 - ความสัมพันธ์ระหว่างเหตุการณ์
 - ตัวอย่าง: ข้อความ: "ถ้าเธอไม่ออกไปก็ไม่ต้องไป!"
ผลลัพธ์:
 - ความสัมพันธ์: บุคคล A → บุคคล B
 - ประเภทข้อความ: "ถอน"
- **Sentiment Analysis** วิเคราะห์อารมณ์ของข้อความว่าเป็น เชิงบวก, เชิงลบ หรือ ประชด โดยใช้โมเดล NLP เช่น ThaiGPT, wangchanBERTa
 - ตัวอย่าง: ข้อความ: "ดีจังเลยที่เธอลืมเราได้" ผลลัพธ์:
 - ประเภทข้อความ: "ประชดประชัน"
 - อารมณ์: "เชิงลบ"

Language Resources

ในการพัฒนาโปรแกรมวิเคราะห์ข้อความเพื่อระบุว่าข้อความเป็นการรอนหรือประชดประชันหรือไม่ จำเป็นต้องใช้แหล่งข้อมูลทางภาษาที่เกี่ยวข้องเพื่อช่วยในกระบวนการประมวลผลภาษาธรรมชาติ (NLP) ดังนี้

- **Keywords (คำสำคัญที่เกี่ยวข้อง)**

- **Natural Language Processing (NLP)** – การประมวลผลภาษาธรรมชาติ
- **การประมวลผลภาษาธรรมชาติ** – การใช้คอมพิวเตอร์วิเคราะห์และเข้าใจภาษา
- **Sentiment Analysis** – วิเคราะห์อารมณ์ข้อความ
- **Emotion Detection** – ตรวจจับอารมณ์ของผู้พูด
- **Sarcasm Detection** – ตรวจจับการประชดประชัน
- **Irony Detection** – การตรวจจับการเสียดสี
- **การประชดประชัน** – คำพูดที่มีความหมายตรงข้ามกับสิ่งที่ผู้พูดต้องการสื่อ
- **การรอน** – สถานการณ์ที่ผู้พูดแสดงความไม่พอใจผ่านข้อความ
- **น้อยใจ** – ความรู้สึกเสียใจที่ไม่ได้รับความสนใจหรือถูกมองข้าม
- **การวิเคราะห์ข้อความ** – การใช้ AI วิเคราะห์ประเภทและอารมณ์ของข้อความ
- **Text Classification** – การจำแนกประเภทของข้อความ
- **Named Entity Recognition (NER)** – การระบุหน่วยข้อมูลเฉพาะ

1. Corpora & Datasets (คลังข้อมูลภาษาไทย)

- **BEST 2010 (NECTEC)**
 - i. เป็นชุดข้อมูลที่รวบรวมข้อความจาก ข่าว, บทความ, บล็อก, และวรรณกรรม
 - ii. ใช้สำหรับ การตัดคำและการจำแนกประเภทข้อความ
- **Thai Social Media Texts Dataset**
 - i. รวบรวมข้อความจากโซเชียลมีเดีย เช่น Facebook, Twitter
 - ii. เหมาะสำหรับ การฝึกโมเดลให้เข้าใจภาษาพูด, สแลง, และการประชดประชัน
- **Wisesight Sentiment Corpus**
 - i. ชุดข้อมูลความคิดเห็นจากโซเชียลมีเดีย ที่มีการติดป้ายกำกับว่า บวก, ลบ, กลาง
 - ii. ใช้ช่วยโมเดลในการวิเคราะห์ อารมณ์ของข้อความ

2. Tools (เครื่องมือที่ใช้ในโครงการ)

- **PyThaiNLP** – ไลบรารี NLP สำหรับภาษาไทย รองรับ Tokenization, POS Tagging, Sentiment Analysis
- **Hugging Face Transformers** – ใช้โมเดลปัญญาประดิษฐ์เชิงลึกรองรับโมเดลเช่น *wangchanBERTa, mT5, ThaiGPT*
- **SpaCy** – เครื่องมือ NLP สำหรับการวิเคราะห์ข้อความ (ภาพที่ 6)
- **Scikit-learn** – ไลบรารีสำหรับ Machine Learning
- **TensorFlow & PyTorch** – Framework สำหรับ Deep Learning
- **LangChain** – สำหรับสร้างแอปพลิเคชันที่ใช้ LLM



ภาพที่ 5 ภาพแสดงแหล่งชุมชนออนไลน์ที่เกี่ยวข้องกับ NLP ซึ่งเป็นศูนย์กลางสำหรับนักพัฒนาและนักวิจัย

1. Communities (ชุมชนที่เกี่ยวข้อง)

- PyThaiNLP Community (Facebook Group & Discord) กลุ่มนักพัฒนาที่ใช้ NLP ภาษาไทย
- Hugging Face Community แหล่งรวมโมเดล AI และนักพัฒนา NLP
- Thai AI & Data Science Community (Facebook Group) กลุ่มสนทนาเกี่ยวกับ AI ในไทย
- Reddit NLP ชุมชนวิจัยและพัฒนา NLP ระดับสากล

2. Conferences (งานประชุมที่เกี่ยวข้อง)

- ACL (Association for Computational Linguistics) งานประชุมใหญ่ด้าน NLP และ Computational Linguistics
- EMNLP (Empirical Methods in NLP) งานประชุมเกี่ยวกับการเรียนรู้เชิงลึกใน NLP
- ICLR (International Conference on Learning Representations) งานประชุมเกี่ยวกับ Deep Learning และ NLP
- NAACL (North American Chapter of ACL) งานประชุมย่อยของ ACL ที่เน้นการประยุกต์ใช้ NLP

1. Journal Publications (วารสารวิชาการที่เกี่ยวข้อง)

- Computational Linguistics (MIT Press) วารสารชั้นนำในด้านภาษาศาสตร์คอมพิวเตอร์
- Journal of Artificial Intelligence Research (JAIR) วารสารวิจัยปัญญาประดิษฐ์ที่เกี่ยวข้องกับ NLP
- Transactions of the Association for Computational Linguistics (TACL) วารสารเกี่ยวกับ NLP และ Machine Learning
- IEEE Transactions on Affective Computing วารสารเกี่ยวกับการวิเคราะห์อารมณ์ใน NLP



ภาพที่ 6 ภาพแสดงเครื่องมือที่ใช้ในโครงการด้าน Natural Language Processing (NLP) และ Machine Learning (ML) ซึ่งช่วยในการประมวลผลและวิเคราะห์ข้อความ ได้แก่ PyThaiNLP (ไลบรารี NLP สำหรับภาษาไทย), Hugging Face Transformers, SpaCy, Scikit-learn (ไลบรารีสำหรับ Machine Learning), TensorFlow & PyTorch (Framework สำหรับ Deep Learning), และ LangChain

Document Classification

การจำแนกเอกสาร (Document Classification) เป็นกระบวนการที่ใช้ Machine Learning (ML) และ Natural Language Processing (NLP) ในการแยกประเภทของเอกสารหรือข้อความตามหมวดหมู่ที่กำหนด ในโครงการงานซึ่งเกี่ยวข้องกับ การตรวจจับ "การรอน" และ "การประชดประชัน" เราสามารถใช้การจำแนกเอกสารในลักษณะต่าง ๆ ดังนี้

1. การจำแนกประเภทข้อความตามอารมณ์ (Emotion-based Classification) ใช้สำหรับระบุอารมณ์ของ

ข้อความ เช่น หมวดหมู่:

- งอน (Annoyed/Sulking) → "ไม่ต้องมายุ่งเลย!"
- น้อยใจ (Hurt/Feeling Ignored) → "เห็นเพื่อนไปกับคนอื่น ไม่ชวนเลยนะ"
- ประชดประชัน (Sarcasm/Irony) → "เก่งจังเลยนะ ทำอะไรเองหมดเลยอะ"
- เป็นกลาง (Neutral) → "วันนี้อากาศดีนะ"
- เชิงบวก (Positive) → "ขอบคุณมากนะ ช่วยได้ เยอะเลย"

2. การจำแนกประเภทข้อความประชดประชัน (Sarcasm Detection Classification) ใช้แยกข้อความที่มี การ

ประชดประชัน ออกจากข้อความทั่วไป

- หมวดหมู่:
- ประชดประชัน (Sarcastic/Ironic) → "ว้าว! ช่วย เยอะเลยนะ นิ่งดูเฉยๆ"
 - ไม่ประชดประชัน (Non-Sarcastic) → "ขอบคุณที่ ช่วยกันนะ!"

1. การจำแนกประเภทข้อความในแชตสนทนา

(Conversational Classification) ใช้จำแนกบทสนทนาในแชตตามพฤติกรรม เช่น

หมวดหมู่:

- แสดงความน้อยใจ (Feeling Hurt) → "ไม่เห็น ถามไถ่กันเลย"
- แสดงความงอน (Sulking) → "ตามสบายเลย ไม่ ต้องสนใจกันก็ได้!"
- แสดงความประชดประชัน (Sarcastic) → "ดีจัง เลยนะ อยู่คนเดียวสบายใจมาก"
- แสดงความขอบคุณ (Gratitude) → "ขอบคุณนะ ช่วยได้เยอะเลย"

2. การจำแนกเอกสารรีวิวหรือคอมเมนต์ (Review Classification) ใช้แยก ความคิดเห็นจากโซเชียลมีเดีย, คอมเมนต์ หรือรีวิว ออกเป็นประเภทต่าง ๆ

หมวดหมู่:

- ประชดประชัน (Sarcastic Review) → "อาหาร อร่อยมาก (ประชด) ได้รอ 2 ชั่วโมงเต็มๆ"
- ไม่ประชดประชัน (Genuine Review) → "อาหาร อร่อย พนักงานบริการดี"

3. การจำแนกประเภทข้อความตามเจตนา (Intent Classification) ใช้วิเคราะห์ว่าผู้พูดมีเจตนาอะไรในการ พูดข้อความ

หมวดหมู่:

- ต้องการความสนใจ (Seeking Attention) → "ทำไมไม่ถามไถ่กันเลย?"
- ต้องการให้ฝ่ายตรงข้ามขอโทษ (Want Apology) → "เราผิดเองแหละ ขอโทษด้วยนะ"
- ต้องการยุติการสนทนา (Ending Conversation) → "ตามใจเธอเลย!"

แหล่งอ้างอิง (References)

- **NECTEC (2010).** BEST 2010: The Benchmark for Enhancing the Standard of Thai language processing. Retrieved from <https://www.nectec.or.th/corpus/>
- **Wisegight (2020).** Wisegight Sentiment Corpus: Thai Social Media Sentiment Analysis Dataset. Retrieved from <https://github.com/PyThaiNLP/wisegight-sentiment>
- **PyThaiNLP (2023).** Thai NLP tools for Tokenization, POS tagging, and Named Entity Recognition. Retrieved from <https://github.com/PyThaiNLP/pythainlp>
- **Hugging Face (2022).** wangchanBERTa: Pretrained BERT model for Thai language. Retrieved from <https://huggingface.co/airesearch/wangchanberta-base-att-spm>
- **Scikit-learn Developers (2023).** Machine Learning in Python. Retrieved from <https://scikit-learn.org/stable/>
- **Vaswani, A., et al. (2017).** Attention Is All You Need. Advances in Neural Information Processing Systems (NeurIPS), 30. Retrieved from <https://arxiv.org/abs/1706.03762>
- **Hugging Face (2023).** Transformers: State-of-the-art Natural Language Processing. Retrieved from <https://huggingface.co/docs/transformers/index>
- **TensorFlow Developers (2023).** TensorFlow: An Open Source Machine Learning Framework for Everyone. Retrieved from <https://www.tensorflow.org/>
- **PyTorch Developers (2023).** PyTorch: Tensors and Dynamic Neural Networks in Python with Strong GPU Acceleration. Retrieved from <https://pytorch.org/>
- **Hochreiter, S., & Schmidhuber, J. (1997).** Long Short-Term Memory. Neural Computation, 9(8), 1735–1780. Retrieved from <https://www.bioinf.jku.at/publications/older/2604.pdf>