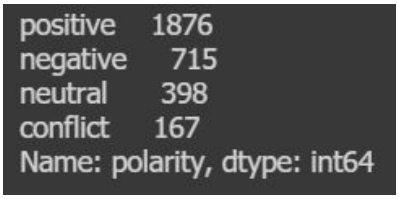
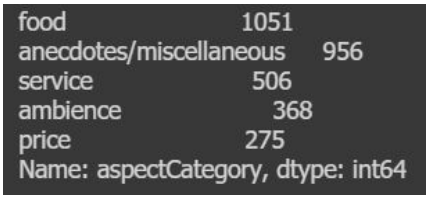
Experimental Setup

เนื่องจากข้อมูลมีขนาดกลาง คือ 3156 ประโยคจึงแบ่งสัดส่วนเป็น dev 20% และ train 80% ของข้อมูลทั้งหมด

ประกอบกับ Label distribution ไม่สมดุลเท่าที่ควร เนื่องจากแต่ละ label มีจำนวนที่ต่างกันมากๆ ทั้ง polarity label และ aspect label ดังรูป





ดังนั้น จึงเหมาะกับสัดส่วน 80 : 20 มากที่สุด training data จะได้ไม่น้อยเกินไป เพื่อให้โมเดลได้เรียนรู้ในตอนเทรนมากที่สุด

Model

ทดลองทั้งหมด 3 รูปแบบ คือ

Logistic regression(1)

Feature : [tokenize] + [contractions หรือ การเปลี่ยนรูปคำย่อต่างๆ เช่น is’nt ---> is not] + [unigram และ bigram] และแปลงให้เป็น sparse feature matrix

Feature นี้น่าจะเป็นผลดีต่อประสิทธิภาพของโมเดล เพราะ การเปลี่ยนคำย่อทำให้มีรูปเต็มของ not ในประโยค โมเดลสามารถเรียนรู้ได้ง่ายขึ้นใน label negative หรือ conflict ส่วนการทำ unigram และ bigram น่าจะทำให้โมเดลทำนายจากบริบทของแต่ละคำในประโยคได้มากขึ้น และการแปลงให้เป็น sparse feature matrix จะทำให้ค่าส่วนใหญ่ใน matrix เป็น 0 ประหยัดเนื้อที่และเวลาในการประมวลผลมากกว่า dense feature matrix

Logistic regression(2)

Feature : [tokenize] + [contractions หรือ การเปลี่ยนรูปคำย่อต่างๆ เช่น is’nt ---> is not] + [กรองให้เหลือแต่ คำนาม คำคุณศัพท์ คำกริยา และ คำกริยาวิเศษณ์ด้วย nltk pos tagger] และแปลงให้เป็น sparse feature matrix

Feature นี้น่าจะเป็นผลดีต่อประสิทธิภาพของโมเดล เพราะ การเปลี่ยนคำย่อทำให้มีรูปเต็มของ not ในประโยค โมเดลสามารถเรียนรู้ได้ง่ายขึ้นใน label negative หรือ conflict ส่วนการกรองให้เหลือเพียง คำนาม คำคุณศัพท์ คำกริยา และคำกริยาวิเศษณ์เพราะคำเหล่านี้สำคัญในการสื่อถึง sensiment รวมถึง aspect มากกว่าคำชนิดอื่นๆ และการแปลงให้เป็น sparse feature matrix จะทำให้ค่าส่วนใหญ่ใน matrix เป็น 0 ประหยัดเนื้อที่และเวลาในการประมวลผลมากกว่า dense feature matrix

Neural network

Feature: [tokenize] + [contractions หรือ การเปลี่ยนรูปคำย่อต่างๆ เช่น is’nt ---> is not] + [กรองให้เหลือแต่ คำนาม คำคุณศัพท์ คำกริยา และ คำกริยาวิเศษณ์ด้วย nltk pos tagger] คล้ายๆกับ logistic regression (2) และเลือกใช้ Pre-train Word Embedding เนื่องจากเห็นว่าเป็นข้อมูลขนาดใหญ่ รวมถึงมีคลังของศัพท์ว่าคำไหนใช้ในบริบทไหน น่าจะทำให้โมเดลเรียนรู้จากคำเหล่านั้นได้มากขึ้น แต่อย่างไรก็ตามเนื่องจากจำนวนข้อมูลในEmbedding นั้นใหญ่เกินไปจึงเลือกเฉพาะคำที่มีอยู่ในไฟล์เทรนมาหา word embedding

เลือกใช้ Neural network แบบ Bidirectional LSTM เพราะมีความสามารถในเรื่อง classification

Hidden Layers: มีการใช้Hidden Layers หลายชั้นเพื่อให้โมเดลมีการ back Propagate เพราะคาดว่าจะทำให้โมเดลได้เรียนรู้จากความผิดพลาดและปรับปรุงให้ได้ผลลัพท์ที่แม่นยำที่สุด นอกจากนี้มีการใช้ Drop Out ระหว่าง hidden layer เพื่อลดโอกาสในการเกิด Overfit เท่าที่จะทำได้

Results

โมเดลแรกได้ผลแย่สุด เพราะอาจไม่เกี่ยวกับบริบท โมเดลต่อมามีการกรองคำ คำประเภทคำนาม กริยา คุณศัพท์ กริยาวิเศษณ์จึงน่าจะมีผลมาก และสุดท้ายใช้ Model ที่ซับซ้อนขึ้นมาจึงมีค่ามากที่สุด

Loggistic regression Uni+Bi(1):

=== CLASSIFICATION : ASPECT ===

	class name	precision	recall	F1-score	support
0	food	0.849	0.778	0.812	203
1	price	0.727	0.133	0.225	60
2	service	0.866	0.558	0.678	104
3	ambience	0.769	0.282	0.412	71
4	anecdotes/miscellaneous	0.744	0.794	0.768	194
5	MACRO AVG	0.791	0.509	0.579	632
6	MICRO AVG	0.801	0.630	0.705	632

=== CLASSIFICATION : SENTIMENT ===

	class name	precision	recall	F1-score	support
0	positive	0.721	0.895	0.799	306
1	negative	0.530	0.280	0.366	125
2	neutral	0.533	0.216	0.308	74
3	conflict	0.429	0.375	0.400	24
4	MACRO AVG	0.553	0.442	0.468	529
5	MICRO AVG	0.672	0.631	0.651	529

=== CLASSIFICATION : OVERALL ===

	precision	recall	F1-score	support
0 MICRO AVG	0.539	0.424	0.475	632

Logistic regression-Unigram(2):

=== CLASSIFICATION : ASPECT ===

	class name	precision	recall	F1-score	support
0	food	0.878	0.778	0.825	203
1	price	0.889	0.267	0.410	60
2	service	0.881	0.567	0.690	104
3	ambience	0.882	0.423	0.571	71
4	anecdotes/miscellaneous	0.773	0.789	0.781	194
5	MACRO AVG	0.860	0.565	0.655	632
6	MICRO AVG	0.837	0.658	0.737	632

=== CLASSIFICATION : SENTIMENT ===

	class name	precision	recall	F1-score	support
0	positive	0.761	0.863	0.809	306
1	negative	0.596	0.496	0.541	125
2	neutral	0.595	0.297	0.396	74
3	conflict	0.222	0.083	0.121	24
4	MACRO AVG	0.543	0.435	0.467	529
5	MICRO AVG	0.704	0.662	0.682	529

=== CLASSIFICATION : OVERALL ===

	precision	recall	F1-score	support
0 MICRO AVG	0.586	0.460	0.516	632

Neural Network:

=== CLASSIFICATION : ASPECT ===

	class name	precision	recall	F1-score	support
0	food	0.847	0.897	0.871	203
1	price	0.727	0.133	0.225	60
2	service	0.942	0.625	0.751	104
3	ambience	0.826	0.535	0.650	71
4	anecdotes/miscellaneous	0.846	0.680	0.754	194
5	MACRO AVG	0.838	0.574	0.650	632
6	MICRO AVG	0.855	0.672	0.753	632

=== CLASSIFICATION : SENTIMENT ===

	class name	precision	recall	F1-score	support
0	positive	0.834	0.820	0.827	306
1	negative	0.634	0.568	0.599	125
2	neutral	0.437	0.419	0.428	74
3	conflict	0.154	0.083	0.108	24
4	MACRO AVG	0.515	0.473	0.490	529
5	MICRO AVG	0.714	0.671	0.692	529

=== CLASSIFICATION : OVERALL ===

	precision	recall	F1-score	support
0 MICRO AVG	0.608	0.478	0.535	632

Conclusion

เลือกใช้โมเดล Neural Network เนื่องจากเป็นโมเดลที่แม่นยำมากที่สุด คิดว่าได้ผลดีกว่า logistic regression เนื่องจาก neural network เป็นโมเดลที่ซับซ้อนกว่า เปรียบเสมือน logistic regression หลายๆอันมารวมกัน และ word-embedding ใช้ดูความเชื่อมโยงทางบริบทได้ดีกว่า unigram ผสมกับ bigram ด้วย แต่คะแนนไม่ห่างมากอาจเนื่องมาจากการ overfitting ของข้อมูล (ค่า validation accuracy มักจะต่ำกว่าค่า training accuracy เมื่อดูจาก history)