Comparing Aspect-Based Sentiment Analysis Methods Across Variations In Training Data Size

Jane Hung
2021 April 10

Abstract

Aspect-based sentiment analysis (ABSA) can be a difficult NLP problem space for reasons such as the lack of relevant tagged corpora, the need to represent context and domain knowledge within the algorithms, and the challenge of assessing two different types of problems - aspect extraction (AE) and aspect sentiment classification (ASC). This research seeks to tackle logistical issues around training data size and overall tagged corpus size to inform practitioners of the significant benefit of deep learning models (BERT) over rule-based mechanisms in the ABSA space, regardless of the constraints driven by manually tagging a relevant corpus. The study concludes that training data size and therefore overall tagged corpus size does not have an effect on BERT performance, and the two modalities do not converge in practice.

Keywords

aspect-based sentiment analysis; rule-based; BERT

Introduction

Aspect-based sentiment analysis (ABSA) is an NLP technique that includes two major steps: aspect extraction (AE) and aspect sentiment classification (ASC). The first step includes isolating the terms of interest within a sentence whereas the second step entails identifying the related opinion around the extracted terms. ABSA is a markedly more difficult task than sentence-level sentiment analysis because each sentence may have multiple terms each with an associated sentiment. However, ABSA can provide valuable context around pain point drivers in software support tickets or well-liked features in a restaurant, for example.

One major issue in ABSA is embedding domain specific knowledge to ascertain relevant terms and sentiments. For example, BERT is commonly used with great success in NLP tasks, but since it is trained on objective Wikipedia articles, it may not produce the same gain when used in sentiment analysis tasks due to the subjective nature of the problem.[2,4] Therefore, there is a need to fine-tune ABSA methods to a specific domain in order to be effective, which requires a significant amount of domain knowledge data.

Underlying this issue is the need for large amounts of tagged training data in order to proceed with these domain-tuning tasks. As part of this analysis, I explored whether

there is a minimum training data size requirement to building deep learning models using BERT. The goal of this analysis is to inform whether rule-based methods for ABSA outperform deep learning models when training data is limited.

Approach

Data

The analysis utilizes a tagged corpus from the International Workshop on Semantic Evaluation (SemEval) that centers on laptop reviews. Entities are tagged within each sentence according to the IOB strategy, where “B” determines the beginning of an entity, “I” determines whether the following words are part of that entity, and “O” defines words as not part of an entity.[8] This corpus is further tagged with a sentiment polarity with three classes - negative, neutral, and positive. This corpus was chosen due to the wealth of research completed on SemEval data that would allow fair comparisons outside of this analysis. Furthermore, due to the lack of tagged data in this space, I sought to choose a usable dataset that would inform logistics in future ABSA projects.

Baseline

To develop a baseline, I used rule-based tools that incorporate an external tagged corpus and may be ideal in cases where relevant tagged domain data is difficult to find. Practically speaking, this approach may be ideal where time and resources are limited.[10]

To create a baseline model for the dual tasks of AE and ASC, I first designed a noun phrase chunker that tagged each word with their part-of-speech (POS) and assigned consecutive nouns to a noun phrase chunk, which formed an entity of interest. This initial model takes a sentence and then labels tokens with IOB designations.[8] Building off this naive approach, I further utilized SemEval guidelines for ABSA annotations to elucidate whether business rules would provide a performance lift.[7]

After developing an AE mechanism, I explored using the sentiment lexicon, VADER (Valence Aware Dictionary and sEntiment Reasoner), to assign a positive, neutral, negative sentiment categorization to an entity given a sentence. Since it is fairly uncommon in this dataset to see multiple entities within a sentence as well as multiple entities with different polarities, this research used sentence-level sentiment classification to attribute a sentiment polarity to an entity, which forms the ASC naive approach.[6]

ML Models

BERT encodings formed the foundation of baseline model improvements due to their prevalence in existing ABSA research as well as their ability to provide contextualized representations.[1]

For AE, I fine-tuned all $BERT_{BASE}$ (uncased) layers for token classification whereas for ASC, I fine-tuned the embeddings for sequence classification (see Appendix).[5]

Results

To evaluate the models using the full dataset, I utilized the SemEval evaluation procedure that focuses on entity-level performance rather than token-level performance.[3,9] I used the partial match strategy to add flexible constraints around entity detection, which in practice may prevent information loss without the added noise of spurious predictions.

AE Model	Results			
	Precision	Recall	F1-Score	Counts
POS Tagger + Regex Parser	0.27	0.75	0.39	Correct: 422 Partial: 192 Missed: 75 Spurious: 1336
POS Tagger + Regex Parser + SemEval Rules	0.27	0.64	0.38	Correct: 327 Partial: 223 Missed: 140 Spurious: 1094
$BERT_{BASE}$	0.80	0.81	0.80	Correct: 478 Partial: 124 Missed: 68 Spurious: 72

The poor performance of the naive AE model elucidates the need for machine learning methods that can incorporate contextual meaning. It has imbalanced precision and recall, suggesting that the model is overzealous in entity detection, and due to the high number of spurious and missed predictions, this model may prove useless in practice.

Notably, when adding in business rules as dictated by the SemEval annotation guidelines, model performance does not improve, but there are fewer spurious entities identified. As shown in the above table, a major issue with these naive models is a

penchant for extracting entities that are not tagged in the text, leading to a higher recall value. In particular, these models perform well in cases where named entities (e.g. product names, such as “Mac Book Pro”) and homographs (e.g. “caught a chill” (noun) vs. “just chill” (verb)) are not present. SemEval rules specifically dictate that product names are not selected as entities in ABSA, which is logic that the baseline models could not discern well. This concept however uncovers the difficulty with standardizing entity labeling within a corpus. In particular, these named entities perhaps should be considered true entities since they may be the main topic of a review. To improve on the homograph issue, I migrated to a methodology that does not rely on rule-based POS tagging, which may better ascertain context in entity extraction.

The $BERT_{BASE}$ implementation widely outperforms the rule-based models in precision, recall, and F1-score. Curiously, this model fairs similarly against the initial naive model in counts of entity error types aside from the lower count of spurious entities, which suggests this model provides more useful entity sets. This model still has trouble with homographs, where for example, it cannot identify the word, “use,” as an entity in the below sentence.

```
True: {'ports', 'design', 'keyboard'}

Predicted: {'use', 'ports', 'design', 'keyboard'}

Sentence:
['[CLS]','i','like','the','design','and','ease','of','use','with','the','k
eyboard',',','plenty','of','ports','.','[SEP]','[PAD]',...,'[PAD]']
```

Furthermore, the BERT model has difficulty deciphering the full entity, “legacy programs,” and instead only recognizes the word, “programs,” which suggests that the model does not have enough computer domain knowledge to recognize the impact of the word, “legacy.” As such, this further supports existing research in data domain tuning in BERT.[4]

ASC Model	**Results**	
	Accuracy	**Macro-F1**
VADER	0.63	0.57
$BERT_{BASE}$	0.75	0.69

For the ASC models, I used accuracy and macro-F1 to evaluate multi-class classification performance. Moreover, I assessed the number of mispredictions resulting in the most severe error type, i.e. when the sentiment is predicted negative and is actually positive and vice versa). As shown below, the naive ASC model has 40 total predictions that fall under this characterization.

VADER MODEL

True:	Predictions:	Count (Prop):
negative	neutral	45 (0.188285)
	positive	**22 (0.092050)**
neutral	negative	32 (0.133891)
	positive	56 (0.234310)
positive	negative	**18 (0.075314)**
	neutral	66 (0.276151)

In one example, the naive model has difficulty assessing sarcasm and incorrectly classified the below entity as positive when it is actually negative, which showcases logic that is difficult to embed in a rule-based system.

Entity: Apple "Help"

Sentence: Apple "Help" is a mixed bag.

BERT MODEL

True:	Predictions:	Count (Prop):
negative	neutral	14 (0.08750)
	positive	**12 (0.07500)**
neutral	negative	50 (0.31250)
	positive	45 (0.28125)

positive	negative	**18 (0.11250)**
	neutral	21 (0.13125)

The above indicates that even though BERT trumps rule-based methods, it still demonstrates issues with incorrectly classifying the opposing sentiment (30 severe mispredictions). For example, “voice recording” below should be a negative sentiment, but the BERT model construes this as a positive sentiment, which is indicative of a known problem in NLP where outside knowledge allows us to associate “sounds like the interplanetary transmissions in the “Star Wars” saga” with poorly quality voice recordings.

```
Entity: voice recording

Sentence: Also, in using the built-in camera, my voice recording for my
vlog sounds like the interplanetary transmissions in the "Star Wars" saga.
```

Below, another issue is handling opposing sentiments within a sentence as well as the temporal component, which directly conflicts with our previous assumptions. It may be prudent to embed a higher number of datapoints with this characteristic in order to better predict these edge cases.

```
Entity: portable computing

Sentence: The criticism has waned, and now I'd be the first to recommend
an Air for truly portable computing.
```

Finally, iterating through the various training data sizes yielded no appreciable difference in BERT performance, such that at every tested training dataset size, BERT performance did not converge to baseline levels (see Appendix). This suggests that rule-based methods may not be impactful particularly when BERT encodings can provide exceptional performance even at small training data sizes.

Conclusions

The analysis clearly demonstrates the benefit of BERT encodings over rule-based methods for ABSC. In terms of AE, BERT encodings provided far fewer spurious predictions. However, although BERT encodings improved ASC performance, there are clearly issues with multiple entities, multiple polarities, and contextual/domain knowledge. As discussed, some vital next steps may be to embed data domain tuning and incorporate variations in the corpus to assess problem areas in this research.

References

1. Akbar Karimi, Leonardo Rossi, Andrea Prati. 2021. Improving BERT Performance for Aspect-Based Sentiment Analysis. https://arxiv.org/abs/2010.11731
2. Chi Sun, Luyao Huang, and Xipeng Qiu. 2019. Utilizing BERT for Aspect-Based Sentiment Analysis via Constructing Auxiliary Sentence. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, volume 1, pages 380–385.
3. David S. Batista. 2018. Named-Entity evaluation metrics based on entity-level. http://www.davidsbatista.net/blog/2018/05/09/Named_Entity_Evaluation/
4. Hu Xu, Bing Liu, Lei Shu, and Philip S. Yu. 2019. BERT Post-Training for Review Reading Comprehension and Aspect-based Sentiment Analysis. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, volume 1, pages 2324–2335.
5. Mark Butler. 2021. BERT T5 NER. https://github.com/datasci-w266/2021-spring-main/blob/master/materials/Bert/BERT_T5_NER_2_3_030521.ipynb
6. Qingnan Jiang, Lei Chen, Ruifeng Xu, Xiang Ao, Min Yang. 2019. A Challenge Dataset and Effective Models for Aspect-Based Sentiment Analysis. *In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing,* pages 6280–6285. https://www.aclweb.org/anthology/D19-1654.pdf
7. SemEval 2014 Task 4 Annotation Guidelines. https://alt.qcri.org/semeval2014/task4/data/uploads/semeval14_absa_annotationguidelines.pdf
8. Steven Bird, Ewan Klein, Edward Loper. Chapter 7: Extracting Information from Text. *Natural Language Processing with Python.* http://www.nltk.org/book/ch07.html#ref-ie-postag
9. Xu Liang. 2020. Entity Level Evaluation for NER Task. *Towards Data Science Blog.* https://towardsdatascience.com/entity-level-evaluation-for-ner-task-c21fb3a8edf
10. Youngseok Choi, Habin Lee. 2017. Data properties and the performance of sentiment classification for electronic commerce applications. *Inf Syst Front.* https://link.springer.com/content/pdf/10.1007/s10796-017-9741-7.pdf

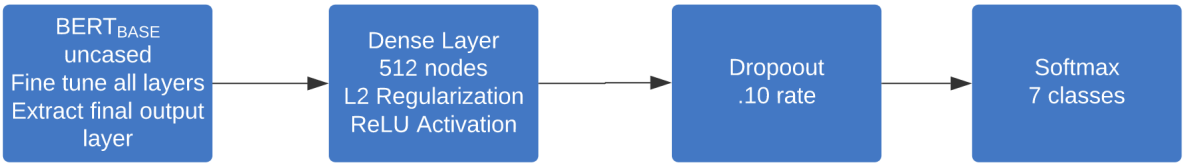
Appendix

Modeling

For the AE model, I exploded the IOB tags as shown below.

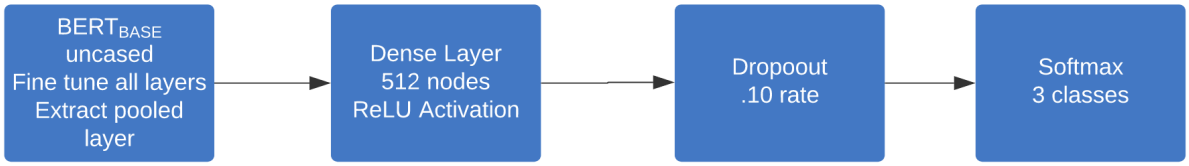
tag	cat
B	0
I	1
O	2
[nerCLS]	3
[nerPAD]	4
[nerSEP]	5
nerX	6

AE Model



Cross-entropy loss
Adam activation with 5E-7 learning rate
10 epochs
8 batch size

ASC Model



Cross-entropy loss
Adam activation with 3E-5 learning rate
5 epochs
16 batch size

Training Size Experiments

		Aspect Extraction								
		Precision			Recall			F1-Score		
Training Data Size	Tagged Corpus Size	Baseline	Baseline +SemEval	BERT	Baseline	Baseline +SemEval	BERT	Baseline	Baseline +SemEval	BERT
2895	3845	0.2656	0.2662	0.8012	0.7518	0.6337	0.806	0.3925	0.3749	0.8036
1000	1950	0.2656	0.2662	0.7211	0.7518	0.6337	0.7019	0.3925	0.3749	0.7113
100	1050	0.2656	0.2662	0.82022	0.7518	0.6337	0.7969	0.3925	0.3749	0.8084

		Aspect Sentiment Classification			
		Macro-F1		Accuracy	
Training Data Size	Tagged Corpus Size	Baseline	BERT	Baseline	BERT
2163	2951	0.57	0.69	0.63	0.75
1000	1788	0.57	0.69	0.63	0.74
100	888	0.57	0.69	0.63	0.75

Evaluation Strategy

According to SemEval, we evaluate AE model performance according to entity-level precision, recall, and F1-score rather than token-level.[3] I implemented the partial evaluation schema, which allows a partial boundary entity match over the surface string and is a more lenient evaluation strategy. I define different categories of error according to the Message Understanding Conference (MUC) as shown below:

- **Correct (COR)** : both are the same;
- **Incorrect (INC)** : the output of a system and the golden annotation don't match;
- **Partial (PAR)** : system and the golden annotation are somewhat "similar" but not the same;
- **Missing (MIS)** : a golden annotation is not captured by a system;
- **Spurius (SPU)** : system produces a response which doesn't exist in the golden annotation;

Scenario	Golden Standard		System Prediction		Evaluation Schema			
	Entity Type	**Surface String**	**Entity Type**	**Surface String**	**Type**	**Partial**	**Exact**	**Strict**
III	brand	TIKOSYN			MIS	MIS	MIS	MIS
II			brand	healthy	SPU	SPU	SPU	SPU
V	drug	warfarin	drug	of warfarin	COR	PAR	INC	INC
IV	drug	propranolol	brand	propranolol	INC	COR	COR	INC
I	drug	phenytoin	drug	phenytoin	COR	COR	COR	COR
I	Drug	theophylline	drug	theophylline	COR	COR	COR	COR
VI	group	contraceptives	drug	oral contraceptives	INC	PAR	INC	INC

$$\text{POSSIBLE}(POS) = COR + INC + PAR + MIS = TP + FN$$

$$\text{ACTUAL}(ACT) = COR + INC + PAR + SPU = TP + FP$$

$$\text{Precision} = \frac{COR + 0.5 \times PAR}{ACT} = \frac{TP}{TP+FP}$$

$$\text{Recall} = \frac{COR + 0.5 \times PAR}{POS} = \frac{COR}{ACT} = \frac{TP}{TP+FP}$$

ASC model performance is determined by accuracy and macro F1-score, which is appropriate for the multiclass classification problem (i.e. positive, negative, neutral).