Aggregation is Not All You Need

Generalizing Improvements in Aspect-Based Sentiment Analysis

Vaibhav Beohar and **Julian Hicks** and **Dasa Ponnappan**
School of Information, UC Berkeley, Fall 2021
{vbeohar, julian.hicks, epdasa}@berkeley.edu

Abstract

We explore recent improvements in Aspect Based Sentiment Analysis (ABSA), which is a subfield of sentiment analysis. Despite widespread adoption of sentiment analysis, ABSA is a challenging task because it involves examining the type of sentiments as well as sentiment targets expressed in product reviews, both of which are domain-dependent tasks. We examine a recent paper by Karimi et al. (1) and find that using newer and generalized models in place of domain-specific models can improve performance on sentiment classification while impairing performance on sentiment target identification.

1 Introduction

ABSA studies consumer opinions on market products and involves examining sentiment valences as well as sentiment targets expressed in product reviews. In our work, we explore the novel BERT-based architectures proposed by (1) for two main ABSA tasks, namely Aspect Extraction (AE) and Aspect Sentiment Classification (ASC), in order to improve their model's performance.

1.1 Aspect Extraction

One of the two core tasks in ABSA, Aspect extraction (AE) attempts to find aspects on which reviewers have expressed opinions as explained by (2). In supervised settings, it is typically modeled as a sequence labeling task, where each token from a sentence is labeled as one of {Begin, Inside, Outside}. A continuous chunk of tokens that are labeled as one "B" and followed by zero or more "I"s forms an aspect. The authors of (1) build upon the work of (3) to adapt BERT's general language models and to incorporate domain word embeddings to improve the BERT's performance.

1.2 Aspect Sentiment Classification

Aspect sentiment classification (ASC) classifies sentiment polarities (positive, negative, or neutral) expressed on an aspect extracted from a review sentence. There are two inputs to ASC: an aspect and a review sentence mentioning that aspect. As explained by (1), the representation for this element is embodied in the architecture of the BERT model. For each sequence as input, there are two extra tokens that are used by the BERT model:

$$[CLS], w1, w2, ..., wn, [SEP] \quad (1)$$

The sentiment of a sentence is represented by the [CLS] token representation in the final layer of the architecture. The class probability is, then, computed by the softmax function. However, it is important to note that the sentiment of a sentence often differs from the sentiment of the term itself. For example, see the sentence:

> "While the smoothies are a little big for me, the fresh juices are the best I have ever had!"

In this review, the sentiment expressed towards "smoothies" is negative, while the sentiment of "juices" is positive, as is the sentiment of the overall sentence. ASC is therefore more specific and cannot rely solely on [CLS] token sentiment.

2 Related work

2.1 BERT Models and Architectures

In recent research, BERT has been one of key innovations towards progress on contextualized representation learnings (4)(5)(6). BERT adopts a fine-tuning approach that requires almost no specific architecture for each end task. This is desirable as an intelligent agent should minimize the use of prior human knowledge in the model design.

As mentioned earlier, in this project we attempt to build upon a novel BERT-based architecture introduced in (1), which itself builds upon work that further trains BERT for domain-specific tasks in (7) and (3). This enriches word and sentence level representations using additional domain-specific restaurant and laptop data by post-training BERT models, which they call BERT-PT. Following on this enrichment, aggregation layers which extract information from the four final layers of BERT are added. This builds on work which shows that hidden layers of deep networks, in particular BERT, can provide region-specific information useful for various tasks.

2.1.1 Parallel and Hierarchical Aggregation

As described by (8) BERT and deep models can capture knowledge of the language as they grow. The initial to middle layers of BERT were shown to extract syntactic information, whereas the language semantics are represented in higher layers.

In the architecture presented by (1), the information in the final four hidden layers is exploited using two similar methods: parallel aggregation and hierarchical aggregation. In parallel aggregation, prediction is performed on a BERT layer fed with the outputs of each hidden layer and then aggregated. In hierarchical aggregation, the outputs of the new BERT layers are fed into the input of the following new BERT layer.

Additionally, in the aspect-extraction task, the outputs from these layers are modified using conditional random fields in order to capture sequential information.

2.2 Alternative BERT-based models

2.2.1 RoBERTa

RoBERTa (Robustly Optimized BERT Pretraining Approach) is a further development of BERT and has achieved better performance than base BERT on many tasks(9). This is due to additional training on the masked language model task, omitting training on the next sentence prediction task and by using a byte-pair encoder for embeddings, similar to GPT-2. We hypothesize that by using a model which generally shows better performance than BERT on most tasks we should yield an ABSA model which additionally performs better on AE and ASC tasks and would potentially be more generalizeable.

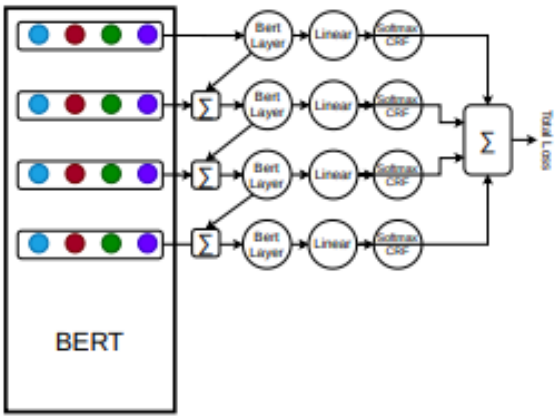

Figure 1: A diagram from (1) illustrating the H-SUM architecture. P-SUM is identical except that the output of each new BERT layer does not feed the layer below it.

2.2.2 SpanBERT

SpanBERT, another version of BERT, uses a pre-training method that is designed to better represent and predict spans of text, thereby aiming to improve pre-training by representing and predicting spans(10). Given the nature of AE as a span-indentification task and the possible interpretation of ASC as an information-extraction task, we propose that a model more attuned to spans of text could yield better performance on these ABSA tasks.

3 Proposed models and research questions

Our work in this paper is twofold:

- Reconstruction and validation studies (running existing models such as BERT-PT)
- Enhancement and research studies (fine-tuning new models and architectures)

Additionally we aim to answer questions in the following two areas:

Generalization. Does the application of parallel or hierarchical aggregation layers (as proposed by Karimi et al) generalize to other versions of BERT that were trained on datasets other than domain-specific BERT-PT?

Aspect Extraction (AE)				
	Train		Test	
Dataset	Sent.	Asp.	Sent.	Asp.
Laptop	3045	2358	800	654
Rest16	2000	1743	676	622

Table 1: Laptop and restaurant datasets for AE. Sent.: Sentences; Asp.: Aspects; Rest16: Restaurant dataset from SemEval 2016

Aspect Classification (ASC)						
	Train			Test		
Dataset	Pos	Neg	Neu	Pos	Neg	Neu
Laptop	987	866	460	341	128	169
Rest14	2164	805	633	728	196	196

Table 2: Laptop and restaurant datasets for ASC. Pos, Neg, Neu: Number of positive, negative, and neutral sentiments, respectively; Rest14: Restaurant dataset from SemEval 2014

Complication. Do PSUM and HSUM architectures yield performance improvements with additional outer layers (as opposed to the current four layers)?

4 Design and implementation

In order to answer these questions, we aimed to maintain continuity with (1) and (3) by using the same data and initial codebase. This meant that our modelling was mainly in PyTorch. Our datasets were also the same, viz. Laptop (LPT14) and restaurant (RST16) datasets from SemEval 2014 and 2016 for AE and the Laptop (LPT14) and restaurant (RST14) datasets from SemEval 2014 for ASC.

As we pursued revisions to the above model, we used the HuggingFace transformers library (11) to enable rapid prototyping of new models. Our source code is available for further research and analysis[1].

We ran our experiments on a Google Cloud environment machine configured at a base `n1-standard-16` architecture and an NVIDIA Tesla T4 GPU, using batches of 16 for all our models. For training, the Adam optimizer was used and the learning rate was set to `3e5`. We used 150 examples from the distributed training data for validation using scripts from (1) and (3).

4.1 Generalized model implementation

To test if these layers provide deeper performance improvements beyond that achieved on BERT-PT, we replaced the base BERT-PT models first with base BERT as a comparison, and then with two alternative models, discussed above: **RoBERTa** and **SpanBERT**.

[1]https://github.com/vbeohar/BERT-for-ABSA-Generalized-UCBerkeley

Additionally, we replicated results from (1) with a slightly different version of **BERT-PT**, which was pretrained on a larger corpus of reviews across domains outside laptop and restaurant reviews.

4.2 Complication model implementation

Based on insights from error analysis of RoBERTa in particular, we hypothesized that accessing more than the four last layers of BERT architectures could provide more information related to structure and grammar.

To investigate, we replicated the P-SUM and H-SUM architectures with eight instead of four layers. We preserved all hyperparameters and used the same BERT-PT models as bases in order to isolate the potential performance gains from doubling layer counts.

5 Results

5.1 Replication & Confirmation Models

To begin, we attempted to replicate past work in this field by re-running BERT-PT without further pre-training on the restaurant and laptop datasets prior to task training (BERT-PT*). We saw similar performance to BERT-PT in past published work.

After having run this model, we noticed that the base version of BERT (`bert-base-uncased`) had not been run with the P-SUM or H-SUM architecture applied. Since we hoped to show that aggregation architectures like P-SUM and H-SUM improved performance regardless of domain-specificity, we trained BERT on our AE and ASC tasks with the P-SUM architecture applied. We saw a strong improvement versus base BERT across all tasks, indicating that aggregation architectures do improve performance on ABSA tasks absent domain-specific pretraining.

			AE		ASC			
			Laptop	Rest16	Laptop		Rest14	
Model	**Aggregation**	**Agg. Layers**	F1	F1	Acc	MF1	Acc	MF1
BERT	None	–	79.28	74.10	75.29	71.91	81.54	71.94
BERT-PT	None	–	84.26	77.97	78.07	75.08	84.95	76.96
BERT-PT	P-SUM	4	**85.94**	**81.99**	**79.55**	**76.81**	**86.30**	**79.68**
BERT-PT	H-SUM	4	**86.09**	**82.34**	**79.40**	76.52	**86.37**	**79.67**
BERT-PT*	None	–	84.07	77.64	77.52	74.36	82.07	72.04
BERT	P-SUM	4	83.14	77.76	77.74	74.23	84.41	76.74
SpanBERT	P-SUM	4	82.41	77.53	76.43	72.99	83.25	75.0
RoBERTa(5)	P-SUM	4	81.28	78.88	78.21	75.01	84.86	77.68
RoBERTa(5)	H-SUM	4	81.96	79.48	78.46	75.44	84.53	77.06
RoBERTa(9)	H-SUM	4	82.59	80.61	**80.23**	**77.61**	**86.25**	**79.71**
BERT-PT	P-SUM	8	84.95	81.13	**79.43**	**76.83**	**85.90**	**78.87**
BERT-PT	H-SUM	8	**85.86**	**82.38**	79.10	76.37	**86.20**	**79.25**

Table 3: Performance of base BERT, Domain-Specific BERT (BERT-PT) and other BERT-based models (RoBERTa and SpanBERT) under different aggregation architectures. Model results presented as the average of 9 runs, except BERT - P-SUM, SpanBERT, and RoBERTa(5 epochs) - P-SUM, of which we only had results for 5 runs at time of submission. **Bold** figures indicate performance within 1 percentage point of the best observed result, and an **underline** indicates our best observed result. Acc: Accuracy, MF1: Macro-F1.

5.2 SpanBERT

After obtaining encouraging results with the PSUM architecture configured on the outer layers of a plain-vanilla BERT model, we decided to explore this option on SpanBERT as well. Our intuition being that SpanBERT, unlike BERT, was trained with masked random contiguous spans rather than random individual tokens – and as such provided improved predictions of spans of texts.

However, after running predictions on 5 trials with 4 epochs each, the SpanBERT PSUM model yielded less than satisfying results as compared to the plain-vanilla PSUM BERT (let alone more advanced RoBERTa and domain-specific BERT-PT models).

An error analysis on the ASC laptop dataset revealed that of those misclassified samples, most of the samples were originally `neutral` (57%) – with a majority of them being classified as `positive`).

To us this indicated that either SpanBERT needed more domain specific training to correctly perform coreference resolution or was confusing sentiments without their contextual representations (for example, `Needs a CD/DVD drive and a bigger power switch` has two aspects `CD/DVD drive` and `power switch`, with `neutral` and `negative` sentiments. In this case, SpanBERT misclassified both sentiments as negative and neutral respectively, effectively flipping the order of the sentiments for the aspects present).

5.3 RoBERTa

When SpanBERT failed to outperform our baseline comparisons, we hoped to yield additional improvements with RoBERTa, a newer and generally better-performing model. We attempted to select the proper number of training epochs by inspecting the training loss and validation loss generated by the model. The results for 9 epochs of training for each task, which were initially performed without inspecting performance on test data, are presented in figure 2. These results seem to indicate that 4 or 5 epochs would be ideal in minimizing both training and validation losses, but we saw stronger results on our test dataset for the model trained on 9 epochs, which we had performed blindly first in order to generate the model loss figures above. This likely is due to us inspecting loss rather than accuracy or F1 scores in our attempts to determine training epoch length.

RoBERTa's AE errors seem to indicate limits to the performance of non-domain-specific

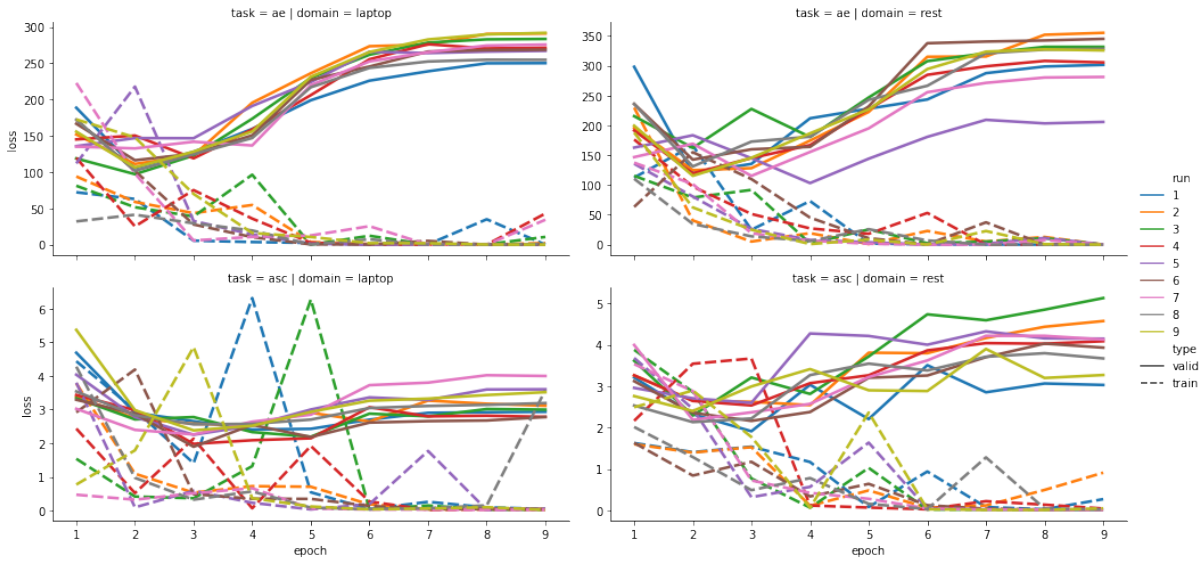

Figure 2: Validation and training losses for RoBERTa - H-SUM over 9 different runs.

models. Inspection of a sample of errors on aspect extraction revealed that many were due to improper handling of proper or domain-specific nouns. For example, in the sentence `Try the Chef's Choice for sushi as the smoked yellowtail was incredible`, our RoBERTa model identified `Chef's Choice`, `sushi` and `smoked yellowtail` as aspects, but yielded an error since the full aspect was `Chef's Choice for sushi`. Similarly, in `has enough storage for most users and many ports`, our model fails to identify `ports` as an aspect of a laptop. Since these domain-specific errors seem common, it is unlikely that we will be able to improve this model's ability to perform aspect extraction without more domain-specific training.

We additionally randomly selected 50 misclassified examples from the ASC task. Of those examined, roughly 26% appear to be due to cue words with positive or negative valence grammatically unrelated to the term in question. For example, in `Stick to the items the place does best, brisket, ribs, wings, cajun shrimp is good, not great`, the model incorrectly identifies `ribs` as having a negative sentiment, despite the `not great` clause clearly referring to the `cajun shrimp`. This indicated that a potential area for improvement would be structuring our model to increase information about regions within sentences and grammatical structure.

5.4 Complication Model Results and Discussion

Upon examining the results of various flavors of BERT (BERT-PT, BERT, SpanBERT, RoBERTa) from table 3 we can make following inferences.

- BERT-PT (pretrained on domain specific tasks) on HSUM and PSUM (85.94 F1 and 86.09 F1) outperforms most of the flavors of BERT, including best performing RoBERTa model HSUM (81.96 4 epochs and 82.59 with 9 epochs) for AE tasks
- Above results of HSUM and PSUM BERT-PT demonstrate that domain specific post-trained model is more accurate for aspect extraction. This shows that aspect extraction task benefits from domain specific training as compared to plain-vanilla BERT or other flavors of post-trained BERT models
- When comparing BERT-PT 4-layer HSUM and PSUM with BERT-PT 8-layer PSUM and HSUM models - it is evident that the addition of more layers does not result in increased scores for either AE or ASC tasks

6 Areas for future research

Our work indicates that better performance on ABSA tasks depends on strong domain knowledge for the Aspect Extraction task but stronger general models for the Aspect Sentiment Classification task. Further progress may be gleaned by pursuing the following questions:

- Can a BERT model which is pre-trained to better understand parts-of-speech or other grammatical structures help yield enhanced accuracy?
- Can further tweaking/permuting the outer BERT layers improve performance over H-SUM and P-SUM architectures? For example, are there better and worse hidden layers from which to extract information? Are there better ways to aggregate information from BERT's hidden layers to maximize meaning?
- Can BERT-based aggregation architectures be combined with other state-of-the-art innovations like Automated Concatenation of Embeddings (ACE) (12) to produce even stronger results? How could such an approach be efficiently implemented?

7 Conclusion

In this paper, we delved deeper into two questions: whether generalized BERT models yield equivalent/or better results as compared to architectures proposed by (1); and whether generalized BERT versions or those with additional layers in HSUM and PSUM aggregation architectures yield equivalent/or better results.

By adding aggregation architecture to base BERT, we saw that even non-specialized language models can improve performance with aggregation methods. By using models like SpanBERT and RoBERTa in the place of base BERT, we saw that improved BERT models are able to match or exceed the performance of domain-specific models in aspect sentiment classification with similar architectures, provided they are robustly post-trained. Finally, in failing to improve results for aspect extraction, we found that domain-specific training is valuable in performance of the aspect extraction task.

References

[1] A. Karimi, L. Rossi, and A. Prati, "Improving bert performance for aspect-based sentiment analysis," 2021.

[2] M. Hu and B. Liu, "Mining and summarizing customer reviews," pp. 168–177, 08 2004.

[3] H. Xu, B. Liu, L. Shu, and P. S. Yu, "Bert post-training for review reading comprehension and aspect-based sentiment analysis," 2019.

[4] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, (New Orleans, Louisiana), pp. 2227–2237, Association for Computational Linguistics, June 2018.

[5] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, (Melbourne, Australia), pp. 328–339, Association for Computational Linguistics, July 2018.

[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *NAACL*, 2019.

[7] A. Karimi, L. Rossi, and A. Prati, "Adversarial training for aspect-based sentiment analysis with bert," 2020.

[8] G. Jawahar, B. Sagot, and D. Seddah, "What does BERT learn about the structure of language?," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, (Florence, Italy), pp. 3651–3657, Association for Computational Linguistics, July 2019.

[9] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized BERT pretraining approach," *CoRR*, vol. abs/1907.11692, 2019.

[10] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, and O. Levy, "Spanbert: Improving pre-training by representing and predicting spans," 2020.

[11] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush, "Huggingface's transformers: State-of-the-art natural language processing," 2019.

[12] X. Wang, Y. Jiang, N. Bach, T. Wang, Z. Huang, F. Huang, and K. Tu, "Automated concatenation of embeddings for structured prediction," *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021.