Aspect-based Sentiment Analysis using BERT

Momal Ijaz
im.momil@knights.ucf.edu

1. ABSTRACT

The problem of sentiment analysis is well-known and highly valued because of it's applications in almost any industry or business. Aspect based sentiment analysis(ABSA) is a more deeper and refined version of the basic sentiment analysis, in which a given text piece is analyzed to extract different aspects and features of the subject entity, and the sentiment associated with the corresponding aspect. ABSA can be done two ways by extracting feature and opinion word from the text and analyzing sentiment of the extracted opinion or it can be performed using category classification.

This paper discusses category classification based ABSA implementation using pretrained language model BERT by mapping the category classification task as sequence pair classification problem and addresses the SemEval task1 and task3 together. A comparative analysis of different version of model reveals domain-specific fine tuned uncased-BERT base model was able to identify the categories with high scores than original pretrained BERT.

2. INTRODUCTION

The demand of sentiment analysis in almost every startup, business or service, which in anyway interacts with it's client in an online setup and gets their reviews could use the unlimited potential of this simple yet powerful application. Businesses can monitor the customer feedback, preferences, problems with the product, and a lot of other useful insights that can help them value their customer needs and grow.

Aspect based sentiment analysis focuses on single aspect of a particular product and tries to understand user's feedback's sentiment towards a particular aspect of the product. This approach is very much needed given the diversity of emotions and approaches every user takes while reviewing an entity, which makes it hard for current basic sentiment analysis systems to classify a given rich opinionated text in a single sentiment category. For example a review about a restaurant "The food was great but expensive, I liked the drinks but their presentation was weird. The overall ambience of the place was very smooth and inviting but their service sucks". Classifying this review as positive,negative,neutral,conflict or any other sentiment category still misses a lot on all the underlying detailed information about various discussed aspects like food and drinks quality, prices, ambience, restaurant in general and ambience of the place. Here comes ABSA to classify or extract all the underlying aspects and their corresponding polarities, to give the user a more fine and deeper and dive into user experience and mass issues.

SemEval is an ongoing series of evaluations of computational semantic analysis solutions. Every year a bunch of tasks are published with some variation and related annotated datasets are released too. Given increased popularity and applications of ABSA it has been made part of research as a regular annual task since 2014. SemEval 2016 task 5 hosted ABSA, provided data-sets were a restaurant and laptop and a combined reviews data-set. These data-sets are provided in multiple languages and annotated for category classification based ABSA. ABSA sub-task 1 deals with aspect category classification, which is to determine all the underlying features of an entity discussed in a given review. Task 2 is about opinion target extraction, which deals with extraction of the words from the review which are refer to a particular aspect class and Task 3 is sentiment polarity classification which deals with prediction of sentiment of all extracted aspect categories.

In this work, task 1 and task 3 are mapped using a combined pre-trained language model BERT. BERT has shown great success for a bunch of varying NLP tasks in a lot of works. Original BERT takes two sentences as input and gives encoded hidden layer sized vectors for each tokens, along with a next sentence prediction binary output. Mapping category classification based ABSA as sequence pair classification using BERT, allows to solve task 1 and task 3 together. Original idea for this architecture was proposed in [1]. Later sections of the paper talks in detail about the related work, data-set, prepossessing, model architecture, prediction pipeline, evaluations over different versions of model in comparison with original scores from [1] as baseline, issues with the architecture, followed by future work and conclusion.

3. RELATED WORK

A lot of solutions have been submitted in SemEval and in various journals for approaching ABSA. [1] talks about mapping ABSA as a sequence classification for BERT problem by taking review and aspect based sentence pairs and their

ACM ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

proposed combined model was able to out perform SOTA over bunch of SemEval tasks. [2] Also convert the aspect to an auxiliary sentence but using a slightly different approach of question answering and natural language inference. [3] takes a highly interesting approach of converting user reviews to a large reading comprehension data-set and answer user questions from that corpus, kind of true exposure to user experience for new users without having to dig into review sea of a product before buying. They also perform ABSA on the created dataset by using a novel fine tuning approach for BERT but the novel approach is not tested on other datasets.

[4] takes the feature extraction based approach for ABSA where the author suggested a POS tagging and dependency parsing rule-based technique for extracting feature/ aspect work along with the associated opinion word. The author then takes a domain specific scores based approach for rating the corresponding the opinion word as positive or negative, to tell if the opinion expressed towards the discussed feature of the entity was positive or negative or neutral. This method does not require a domain based training of any model or pre-defining of the related aspects or features of the entity because a noun phrase of the format 'entity-adjective' can extract feature based opinion from any product review e.g. "The camera was good" or "The garlic sauce was super smooth ", the proposed technique is capable of extraction feature-opinion pairs form both reviews as [[Camera],[Good]] and [[Garlic Sauce],[smooth]]. Then author uses a domain-specific score to get the sentiment of the extracted opinion word. Despite afore-mentioned pros, this technique was only capable of extracting feature based sentiments and failed on phrases context capturing for long reviews like "The camera was good but it had poor zoom quality", the technique wont be able to judge what "it" is referring to in the second sentence of the review. Also the author used a poor and domain specific sentiment score judgement technique and the proposed evaluation metrics were calculated for a task specific close-sourced dataset. Although, a little exploration revealed this technique could be polished and used for opinion target extraction task 2 of SemEval, with a domain trained CNN for extracted sentiment classification.

[5] a solution from Stanford used deep learning based ABSA solution for first time since it officially became a SemEval task in 2014, before this solution it was all SVMs at ABSA. Authors proposed a novel architecture to combine aspect classification and sentiment prediction, and seemed to perform well on selected benchmarks. [6] focuses on a sub task of aspect based sentiment analysis that is aspect target sentiment classification, which is a fancier name for the feature opinion word extraction and calculating polarity of the associated opinion word. Authors proposed a better approach for domain specific fine tuning of pre-trained language models like BERT for boosting its performance on ABSA and surpassing strong baselines like XLNet and BERT-base. The proposed approach for domain specific fine tuning of this solution, could be a possible future improvement for contextual model training approach currently used in this work.

4. DATA

Dataset used for performing all corresponding experiments is SemEval 2016 Task 5 restaurants English language. Training data is publicly availble for download from ***here***. The

Review	Category	Polarity	Target
Price is high but food is good, specially prawn. I will come back.	FOOD#Quality	Positive	Food,prawns
	FOOD#Prices	Negative	Price
	RESTAURANT#Genral	Positive	NULL

Figure 1:
Single data sample from Restaurant reviews dataset - SemEval 2016T5

	GENERAL	PRICES	QUALITY	STYLE& OPTIONS	MISCELLANEOUS
RESTAURANT	✓	✓	✗	✗	✓
FOOD	✗	✓	✓	✓	✗
DRINKS	✗	✓	✓	✓	✗
AMBIENCE	✓	✗	✗	✗	✗
SERVICE	✓	✗	✗	✗	✗
LOCATION	✓	✗	✗	✗	✗

Figure 2:
Aspect-Entity categories of Restaurant dataset from SemEval 2016T5

data-set is available for two varying sub-tasks text level and sentence level. The sentence level data-set is chosen as it is annotated for each review sentence. There are three versions of review data-set available based on entity or main product of review, i.e. laptop reviews, restaurant reviews and laptop+restaurant reviews combined. Each version of sentence level data-set is annotated for all three afore-mentioned three sub-tasks of SemEval's Task5 of 2016. Chosen data-set has 2508 unique reviews for training and 8000 reviews for testing. The test data-set was split into sets of 4000 for validation and testing. A single sample from dataset looks like figure 1.

Each sample has a review text, a category of aspect, an associated polarity and a target word, that discussed the selected category, is given. We are going to focus on category and polarity labels. The category label is a pre-defined set of 12 categories for Restaurant data-set decided by the SemEval community. Where the capitalized word is entity and after symbol / the aspect of that entity follows. Details could be seen in SemEval 2016 Restaurant dataset documentation, a table showing all the 12 categories for our dataset is given in figure 2. Drinks,food are usually discussed in terms of their price, quality and presentation, whereas ambience,location and service are discussed generally. Also few reviews talk about restaurant overall in terms of price or generally.

5. METHODS

5.1 ABSA as Sequence Pair Classification

BERT is a famous pre-trained language model that is actually made by stacking encoders from famous self-attention based model, transformers. BERT is a bi-directional encoder representation from transformers, which takes two input sentences as input and pass them through a bunch of

stacked transformer encoders, as the input and output size of each transformer block is similar there is no need of input or output reshaping in between encoder blocks. Output of original BERT is two folded, both outputs contributed in pre-training of BERT for it to understand language better. Used vocabulary was word piece with 30k words and special tokens like [CLS] for sentence start, [SEP] for sentence end, [UNK] for words out of vocabulary, [PAD] for padded tokens and [MASK] for masked tokens. First output is NSP or next sentence prediction, which tells if the second sentence follows the first sentence or not, a binary classifier.This task helps the model learn about context. The second output is for masked language modelling, which masks random words in input sentence and network tries to predict the masked word. The number of output tokens is similar to the number of input tokens for both sentences. This task helps the model learn about language in general, making it able to suggest sensible words for masked tokens.

Sequence pair classification is a famous task that can be performed using BERT. In this task we pass two input sentences to BERT and it is expected to classify them in one of the pre-defined categories, those categories can be sentiment, aspect or entire entity, depending upon given data-set. For modelling ABSA as sequence pair classification, the first sentence is selected as review text and the second sentence is formed by transforming the original category data labels from the data-set into a sentence format. That is done by transforming "FOODquality" into "Food, Quality" or generally "Entity, aspect". The target labels or classes to classify this input sentence pair is three sentiments i.e. positive,neutral,negative and a new label 'unrelated', which means that this entity-aspect pair is not discussed in this review. So for a given entity-aspect pair category and a review sentence, we classify the sentence pair into all related classes, hence a multi-class classification problem, because each review can talk about more than one entity-aspect pair of the subject. So given a review and category, if the category is discussed in the review, model gives the sentiment associated with the category otherwise gives unrelated label.

5.2 Data Prepossessing

The trial, test and train datasets are given in XML format. Excluding the target label column the remaining three data columns i.e. reviews text,category and polarity were read into python pandas data-frame. Later all the category samples for test,validation and train data-set are converted sentence format as mentioned earlier. The new polarity class unrelated is added in data by assuming the categories not given in original review annotation as unrelated and adding them to the given review with unrelated label. So each review sentence is repeated 12 or number of classes times in the dataset, in pair with each category label, with sentiment(if available) or unrelated polarity label.

Later, HuggingFace transformer library is used for getting BERT-based case and uncased version for experimentation. Using built-in tokenizers the BERT special tokens are added, for padding or truncation based input sequence adjustment, the max length is chosen to be 80, as most of the reviews in dataset are 40-50 tokens or words long, so on safe side, max length of 80 is chosen, corresponding max length distribution for input reviews is shown Figure 3.

These fixed length sequences are transformed into BERT

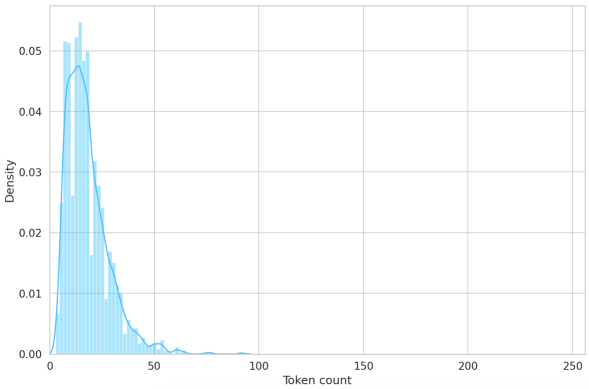

Figure 3:
Training Reviews Token length distribution

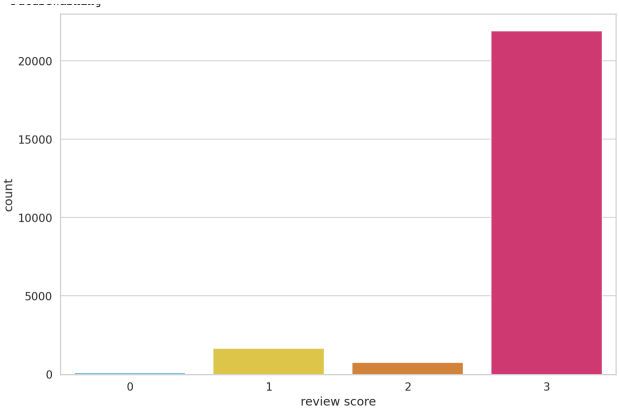

Figure 4:
Training Labels imbalance

specific input features i.e. Input ids(vocabulary index of sequence tokens), Token Type ids(token id to tell which input sentence it belongs to) and attention mask(1 for original token, 0 for pad).

5.2.1 Class Imbalance

Adding unrelated label to data increased the already present bias in the dataset, and made the classes highly imbalanced. Figure 4 shows the imbalance of class labels in the training data-set, similar trends are in the testing and validation set too.

This skewness is adjusted by assigning weights to each label using this formula and are fed to loss function for penalizing the parameters updation from each class by it's weights. Higher frequency lower the weight, hence the class 'unrelated' has smallest weight and 'neutral' has highest weight.

$$Class_A weight = \frac{Total samples}{Class_A frequency * Total Classes}$$

5.3 Model Architecture

Core model used for performing the task was BERT, which is a stacked version of transformer's encoders, famous architecture by [7]. For performing sequence classification a linear

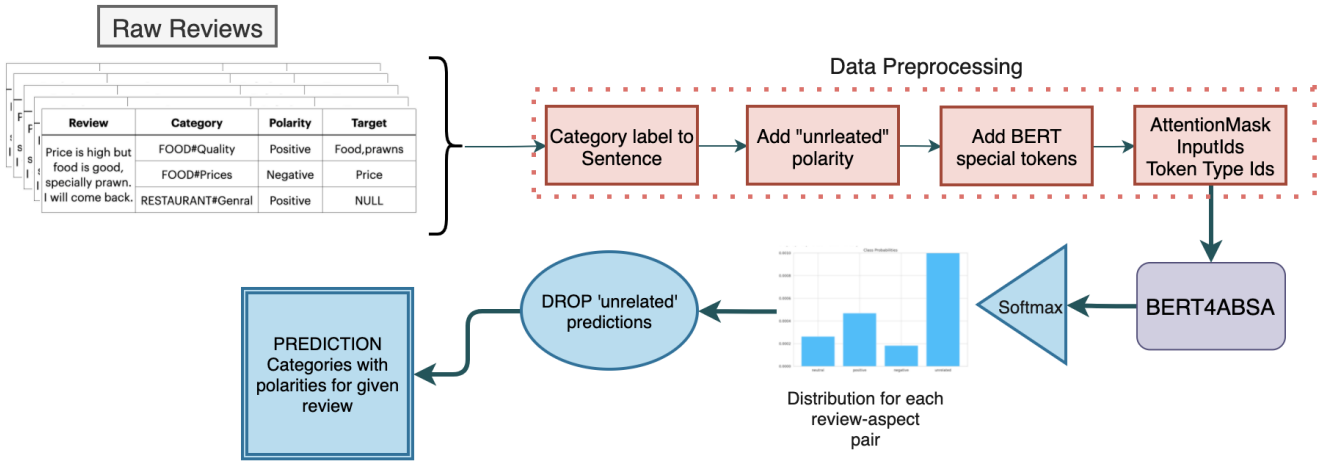

Figure 5: BERT4ABSA: Complete Pipeline

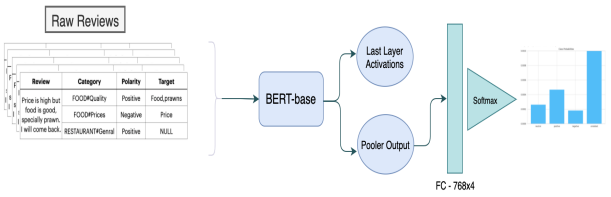

Figure 6:
BERT4ABSA: BERT-base with custom Sequence Classification Head for ABSA

layer is added on top of pooler output of BERT-base, which comes with default 12 heads for multi-head attention, 30K vocabulary words, 110M pre-trained trainable or freezable parameters and a hidden layer size of 768. From huggingface implementation, acesses to two type of layer output is available using any supported pre-trained version of BERT, pooler output and raw final layer activations. The later is [Number of tokens x Hidden layer size] activations from last layer of BERT-base. Whereas the pooler output is of size [1 x Hidden Layer size], this layer acts like a CNN pooling layer and passes the activations from a fully connected linear layer and applies Tanh activation to give a combined highlighted learned encoding for both input sequences, to be used as features for performing deeper required task.

A fully connected linear layer of size [Pooler out x Number of classes] followed by a drop out and softmax activation, to transform the scores into probabilities was used, as can be seen in figure 3. Complete pipeline for training and prediction is shown in figure 5.

5.4 Training Hyperparameters

- BERT, the original paper recommended some hyper parameter for fine-tunning BERT for any desired NLP task, those are, number of epochs=2,4,6 or 8 with some specified ranges for learning rates and batch size. All versions of our models were trained on a total of 4 epochs. Batch size was kept fixed at 16.
- Optimizer of choice was AdamW, with a learning rate of 2e-5.
- A scheduler was used to dynamically update the learning rate as the model trains, with zero warm up steps.
- Loss function of the model was standard cross entropy loss, with calculated training weights for each label, to penalize the contribution from each class to learning(parameter updation) by it's frequency in dataset, for dealing with class imbalance.

6. EVALUATION

As this was a classification problem, typical classification metrics i.e. precision, F1, recall and accuracy were used. But as mentioned before even the test data suffers from high class imbalance, all the reported scores and ac curacies were calculated using weighted sample performance (Micro Average) of all versions of models. Micro Average scores are used to show a model efficiency if the data suffers from high class imbalance, in this approach each sample is given a weight, these weights are calculated using the formula given in data processing section, Weights for each class for the test data are as below: The test-

Label	Weight
Unrelated	0.27
Positive	3.5
Negative	10.5
Neutral	45

ing data scores are reported in the following discussion, all variants of model tried are given with their corresponding precesion, f-1, recall and accuracy scores in figure 7.

First four rows are using BERT-base original pre-trained model with cased or uncased version. Frozen means encoder

Model	Precesion	Recall	F-1	Accuracy
BERT_Cased Frozen	0.36	0.53	0.43	0.53
BERT_Cased UnFrozen	0.73	0.62	0.59	0.62
BERT-Uncased Frozen	**0.73**	**0.68**	**0.67**	**0.68**
BERT-Uncased UnFrozen	0.42	0.56	0.48	0.56
Contx. BERT Uncased UnFrozen	0.18	0.36	0.24	0.36
Contx. BERT Uncased Frozen	0.69	0.61	0.58	0.61
BaseLine Original Paper	*0.77*	*0.75*	*0.43*	*0.95*

Figure 7:
BERT4ABSA different model variants vs. Original Baseline

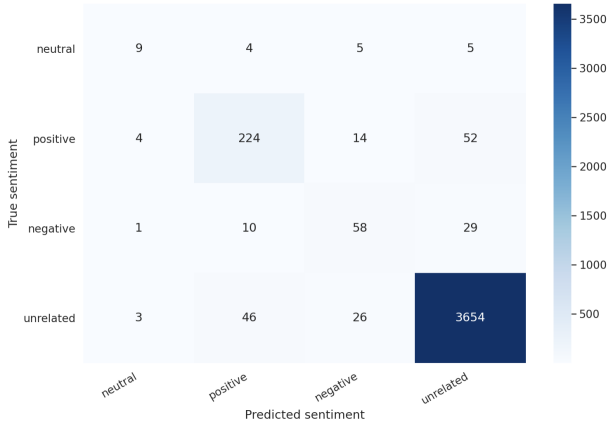

Figure 8:
Confusion Matrix: BERT-base-Uncased Frozen

parameters were fixed during training and unfrozen means encoder parameters were allowed to be updated during training. Row 5 and 6, are bert-base uncased model with frozen and un-frozen encoder parameters respectively. Contx. stan for contextual, these two BERT models were first fine tuned for restaurant reviews domain using masked language modelling, and then the base model domain-pretrained model was used as core BERT in BERT4ABSA model. Last row contains the scores obtained from original paper's experiments using the same architecture over the same dataset, which are taken as baseline across different variants tried.

Also confusion matrix and detailed classification report for the BERT-Uncased Frozen model can be seen in figure 8 and 9.

7. DISCUSSION

First four variants of the models, BERT-cased was expected to show better performance, as the capitalized opinion words like "BAD" or "GREAT" are meant to show more sentiment then non-capitalized sentiment words. But weirdly

	precision	recall	f1-score	support
neutral	0.96	0.39	0.56	1389.507469177246
positive	0.71	0.76	0.74	1082.6279096603394
negative	0.57	0.59	0.58	798.3597679138184
unrelated	0.58	0.98	0.73	1038.9692186415195
accuracy			0.66	4309.464365392923
macro avg	0.70	0.68	0.65	4309.464365392923
weighted avg	0.73	0.66	0.65	4309.464365392923

Figure 9:
Classification Report: BERT-base-Uncased Frozen

the BERT-base-cased performance was some near or worse than the BERT-base-uncased. Also a general trend of improvement in model performance was noticed whenever encoder parameters were allowed to be tuned i.e. unfrozen models. BERT-base uncased seemed to perform well in terms of precision and F-1 scores but recall was lower than original paper's scores, and surprisingly, F-1 scores were higher, but that could be some pre-processing fault or random error, as the hyper parameter choice and model architecture and domain specific fine tuning techniques details were not very clear in the original paper and could definitely be improved to get better results.

Contextual models were trained using domain fine tuned versions of the base BERT. Finetunning was done on original bert-base-uncased model. All the review texts were concatenated in form of a corpus and random MASKs were added in the data, to let the bert guess them, Model parameters were of-course allowed to be updated in this drill, and this domain -fine tuned models were used as base model in BERT4ABSA model, and further finetuned on sequence classification task, to give results for both contextual models in row 5 and 6. Due to poor finetunning technique and lack of training corpus,(only 2000 unique review statement, of length 80) the domain-fientuned BERT models didn't performed any better than the originally pretrained bert-base-uncased model.

8. FUTURE WORK

Except for the vanilla version of the BERT i.e. bert-base-uncased,all other model variants did'nt showed any better or closer performance to original baseline and that is because of a couple of potential issues,that could be potential improvements, are summarized below:

- The BERT-base model has fewer parameters and is relatively simpler model since no experimentation was done with larger and stronger BERT variants, that may result better performance.
- Training BERT-base on same dataset for masked language modeling task was a poor domain fine-tuning technique and other better variants can be explored.
- A better technique for dealing with class imbalance in dataset could be explored, like decreasing abundant class samples or augmenting rare class samples.
- Rule-based feature extraction techniques could be a potential baseline to compare the value added by using complex language model to the ABSA task.

9. CONCLUSION

This work explored Aspect Based Sentiment Analysis which is a deeper and more challenging task of sentiment analysis, dealing with aspect based sentiment extraction of information from the input text. A pretrained language model BERT, with it's various variants is experimented against a baseline from [1], which is from where the idea for model architecture is taken. Vanilla bast-base-uncased performed well on the task, but future improvements are potential hope to take this work towards better results.

Code is available here:

BERT-ABSA

Trained Models

SemEval2016-Training-Test

References

[1] M. Hoang and O. A. Bihorac, *ABSA using BERT*, Sweden. Chalmers University of Technology, Sweden: SemEval, 2019.

[2] C. Sun and L. Huang, *Utilizing BERT for ABSA via Auxiliar sentences* . Fudan University, China: Cornell University Pub., 2019.

[3] H. Xu and B. Liu, *BERT Post-Training for Review Reading Comprehension and ABSA*. University of Illinois Chicago,US: Cornell University Pub., 2019.

[4] N. Chockalingam, *Simple and Effective Feature based ABSA on Product Reviews using Domain Specific Sentiment Scores*. POLIBITS vol.57, 2018.

[5] B. Wang, M. Liu, *Deep Learning for Aspect-Based Sentiment Analysis*. Department of Electrical Engineering, Stanford University.

[6] A. Reitzler, S. Stabinger *Adapt or Get Left Behind: Domain Adaptation through BERT Language Model Finetuning for Aspect-Target Sentiment Classification*. DeepOpinion.ai,Austria, 2019.

[7] A. Vaswani, N. Shahzeer *Attention is all you need*. GoogleBrain, GoogleResearch,NeurIPS,CA,USA 2017.