Approach for Aspect Based Sentiment Analysis

Introduction

Aspect-based sentiment analysis(or ABSA) is the task of identifying fine-grained opinion polarity towards a specific aspect in a given context/text/review.

It is a variant of sentiment analysis, but much more insightful.

In the given dataset,there were 3 classes-Negative,Neutral and Positive

Literature survey-

There has been a substantial amount of research done in the area of ABSA,most of it in the past 10 years.

Generally,in most of the methods,there are 2 parts of achieving the above task-

1.Aspect Extraction(opinion mining)

2.Sentiment Analysis.

However,in our dataset,since the aspect itself was already given,,hence the first part wasn't mandatory.

All the approaches that have been used for ABSA can be broadly divided into three categories-

1.Rule based (or syntactic) approach.

2.Traditional machine learning approaches(SVM,Random forest)

3.Deep Learning approaches(mainly LSTM's/RNN's)

The 3rd approach is what I have used for thai project,as LSTM's have been the best when it comes to state of the art results on ABSA.

Syntactic approaches-

Syntactic approaches mainly worked by extracting the relationship between words in a sentence ,semantic similarities and general grammatical rules.

In order to use rule based approaches,data preprocessing become incredibly important,with steps like negation removal,stemming and stop words removal playing a crucial role.

After some research on this,Angiani et.al concluded that Basic processes and Stemming give the best results[1](http://ceur-ws.org/Vol-1748/paper-06.pdf)

After baic preprocessing,aspect extraction is usually done using a POS tagger.Aspects are usually a noun or noun-phrases but this ideology is not uniform[2](https://sentic.net/aspect-parser.pdf).

For eg,take the below review-

The product design was amazing but overall it was a disappointing buy.

Let's understand the linguistic pattern of aspects in this statement.

disappointing buy: [disappointing — Adjective, buy — Noun].

design amazing: [design — Adjective, amazing — Noun].

What is the pattern here? Extract the aspect from the text if a noun is followed by an adjective.

This is just one of the examples of how rule based approaches can work.

However,as can be expected,this approach is not foolproof.

For eg-one of the statements given in our train set is-

unable to load money using wallets (phonepe/ola money /freecharge/payzapp).

Over here,the adjective(unable) is nowhere near the noun.

This is just one example and almost all rule based approaches have this fallacy of either too many rules or too many exceptions.

Some other papers that employ a syntactic approach are-

Yuanbin et al. [3]

Qiu et al.[4]

Smeaton et al[5]

Maharani et al[6]

Traditional ML approaches-

In recent years, there has also been a focus on using traditional ML approaches for ABSA.

The majority of success has been achieved using SVM's with techniques like Co-clustering Naive Bayes and tree based approaches also having their fair share.

For the task of aspect term extraction on large movie reviews, Asha S Manek et al.[8] compared five feature selection algorithms including SVM and Naive Bayes (NB) classifiers and found that SVM is giving the best results with the Gini Index feature selection algorithm

Nurulhuda et al. [9] proposed a hybrid sentiment classification model that used SVM along with an Association Rule Mining (ARM) technique. Feature selection methods like PCA (Principal Component Analysis), Latent Sentiment Analysis, random project, etc. are applied with a heuristic combination of Parts Of Speech (POS) and also for the extraction of implicit aspects, the authors used Stanford Dependency parser in this work

There has also been work on using a combination of traditional ML and neural networks ,for eg-HILATSA[10] by Karminan et al,which uses a combination of lexicon-based and machine learning in which lexicons and machine learning algorithms like SVM, logistic regression and RNN, are used to extract sentiments from tweets in Arabic language.

There has also been work on Aspect Category Sentiment Analysis(ACSA),which works mainly by dividing aspects in to categories and then classifying each aspect into a predefined topic.This works mainly using LDA for topic modeling

Some papers which use this approach are-

Xinaghua et al[11].

Reinald et al[12]

Deep Learning Approaches-

This has been the approach which has proven to be the most successful in recent years

Most papers use LSTM's/RNN's with a few authors also focussing on using CNN and GRU for ABSA.

Ravindra Kumar et al. [13] applied CNN for the task of ABSA and also stochastic optimization was done in their work. Here semantic feature extraction was done by developing ontologies and word-level embedding is done using word2vec.

Sayed Mahdi Rezaeinia et al. [14] proposed Improved Word Vectors (IWV), an approach that improved the accuracy of pre-trained word embedding's - Word2Vec and Glove.

The approach with most success came out to be LSTM based techniques due to their ability to learn long term dependencies.

Earlier approaches like Xia Ma et al. [15] and Irum Sindhu et al [16] used LSTM's which mainly ahd two layers that focussed on extracting aspects in first layer and then sentiment classification in the second one.

However the real success came with attention based approaches and this is what I have mostly used in my model implementations.

There have been quite a few papers on using Attention based LSTM's for ABSA,with best one's being-

Ruidan He et al[17]

Mohammad Al-Smadi et al. [18]

Minche Song et al[19]

The papers which I have implemented in this project are-

[Aspect Level Sentiment Classification with Attention-over-Attention Neural Networks Binxuan Huang, Yanglan Ou and Kathleen M. Carley][20]

[BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding Jacob Devlin Ming-Wei Chang Kenton Lee Kristina Toutanova][21]

[LCF: A Local Context Focus Mechanism for Aspect-Based Sentiment Classification by Biqing Zeng 1ORCID,Heng Yang 2,*ORCID,Ruyang Xu 2,Wu Zhou 2 andXuli Han 2ORCID][22]

My Approaches-

The sentence is represented as ->w=[w1,w2,w3,...wn]

The aspect is represented as->t=[a1,a2,a3...am]

After applying basic preprocessing steps of converting all letters to lower case and special characters removal(mainly using regex),I mainly applied 3 approaches for ABSA.

1.Non BERT models

I used the **NLTK's WordPunktTokenize**r to tokenize the sentences and create a vocabulary(dictionary) of all the words

After this I created a word embedding tensor of all words using **Glove (the embedding dimensions being 300).**

This embedding tensor served as the weights for my model's embedding layer-

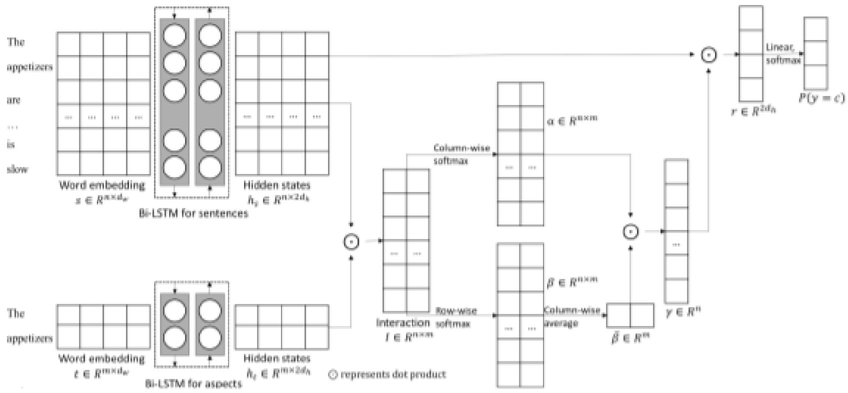
nn.Embedding.from_pretrained(embedding_matrix,freeze=true).

From here on I had 3 models-

1.BI-LSTM(LSTM_imp) with a single layer of LSTM and no dropout.-A single BIdirectional LSTM layer applied on both text and aspect,followed by dot product of outputted tensors and a final linear layer.

2.BI-LSTM with 2 layers and a dropout of 0.2.-Similar as above except for a dropout layer of dropout 0.2 and batch norm layer.

3.Attention over attention-



After the BI-LSTM,AOA mainly works by a column-wise softmax and row-wise softmax on the obtained dot product from BI-LSTM.Through this, we get target-to-sentence attention α and sentence-to-target attention β. After column-wise averaging β, we get a target-level attention β¯ ∈ Rm, which indicating the important parts in an aspect target. The final sentence-level attention γ ∈ Rn is calculated by a weighted sum of each individual target-to-sentence attention α, given by equation.

The L2 regularization coefficient is set to 10−4 and the dropout keep rate is set to 0.2[20]

I use a 80:20 ratio for train and validation sets.

2.BERT models-

The Bert models were based on the paper [LCF: A Local Context Focus Mechanism for Aspect-Based Sentiment Classification by Biqing Zeng 1ORCID,Heng Yang 2,*ORCID,Ruyang Xu 2,Wu Zhou 2 andXuli Han 2ORCID]**.**

For creating the vocabulary dictionary I used the BERT tokenizer based on **"bert-base-uncased".**

BERT uses special tokens [CLS] and [SEP] to understand input properly. [SEP] token has to be inserted at the end of a single input. When a task requires more than one input such as NLI and Q-A tasks, [SEP] token helps the model to understand the end of one input and the start of another input in the same sequence input. [CLS] is a special classification token and the last hidden state of BERT corresponding to this token (h[CLS]) is used for classification tasks

The model takes 4 inputs-

```
concat_bert_indices = tokenizer.text_to_sequence('[CLS] ' + text+ ' [SEP] ' + aspect + " [SEP] ")

concat_segments_indices = [0] * (text_len + 2) + [1] * (aspect_len + 1)

concat_segments_indices = pad_and_truncate(concat_segments_indices, tokenizer.max_seq_len)

text_bert_indices = tokenizer.text_to_sequence("[CLS] " + text + " [SEP]")

aspect_bert_indices = tokenizer.text_to_sequence("[CLS] " + aspect + " [SEP]")
```

One of the most important things is how to determine whether a contextual word belongs to the local context of a specific aspect or not. In order to solve this problem, this paper proposes SRD(*Semantic-Relative Distance*), which aims at assisting models to capture local contexts.

For my training I have set SRD to 3,which in layman terms represents how far are the words ,which can have an effect on aspect context.

There are two modes for training in Local context focus bert models-

1**.CDM mechanism:** Constructing the mask matrices M for the input sequence according to SRD s

2.**CDW mechanism**: Constructing the weighting matrices W for the input sequence according to SRDs

CDM/CDW are applied on concat_bert_indices and text_bert_indices,followed by a dropout layer.

After this we concat the outputs from the two dropout layers,followed by a linear layer,Bert self attention layer,a bert pooler and a final linear layer which outputs the class probabilities

In all these models,I have used CrossEntropyLoss and Adam optimizer,with a Pytorch implementation.

2nd Approach-

Question Answer approach in combination with normal sentiment analysis-

In this approach ,I take the aspect and convert that aspect to a question in the form of-

'How is [aspect]?"

For eg- if the aspect is "camera",then the question becomes- How is camera?

For context,I use the text column of csv file?

Using Hugging face's pretrained model and tokenizer(bert-large-uncased-whole-word-masking-finetuned-squad),I get an answer based on the given aspect and context.

I then pass this answer through a normal sentiment analysis model,which outputs the sentiment that is expressed towards a given aspect in the given text.

3rd approach-.**Syntactic Techniques-**

This was a rule based approach ,wherein I devised a couple of rules for ABSA.

Steps-

1.Preprocessing done mainly using spacy.

2.Devising the rules for aspect extraction.

a.Extract the aspect from the text if a **noun** is followed by an **adjective.**

b.Extract the aspect from the text if an **adjective** is followed by an **adverb** followed by a **compound noun(noun followed by a noun).**

3. Remove Redundant Aspects:There could be few redundant aspects like ["narrative style", "liked narrative style", "really liked narrative style"].

4.Once I had extracted the aspects,I used VADER sentiment analyser to find out the sentiment expressed towards the given aspect in the text.

Metrics Table-

Model	Accuracy	F1_score (average=macro)
LSTM_imp(BI-LSTM)	58.6	56.2
LSTM_imp(dropout)	60.2	56.5
AOA	66.7	64.5
LC-Bert(CDM)	68.6	65
LC-Bert(CDW)	71.8	68.6
QA approach	62.8	61.6
Syntactic approach	56.6	54.8

Error analysis and ablation study-

As is evident from the above table,LC Bert models seem to be performing the best,with AOA close on heels.

1.Effects of SRD on LC Bert models-

I have used a SRD(Semantic Relative Distance) value of 3.However,as we keep on increasing the value of SRD,sentences in which aspect and sentiment describing words are far off ,perform much better,whereas when SRD is less,sentences with aspect and it's sentiment describing words being close enough perform much better.

2.Ablation study on LC Bert models-

LC Bert variation(CDW)	Accuracy	F1-score
No Dropout layer	67.4	64.2
No self attention	63.5	59.8
BatchNorm1D applied	70.4	68
Linear_single layer used instead of Linear_double	70.4	67.6

3.QA models seem to be heavily biased towards Negative class,with far more instances being classified negative.

For eg-

Text:I love tea but hate coffee.

Aspect:tea

This gives output negative

On the other hand-

Text:I really love tea but not coffee.

Aspect:tea

This gives positive output.

The QA model seems to focus a lot more on words with heavily negative connotations(like the word hate).

Conclusion and improvements-

The LC Bert model and the AOA models seem to be performing the best.

I am yet to experiment on the models with certain techniques like Gradient clipping,apply cyclic learning rates and until now have used random weight initialisation for my model layers.

While using Glove,certain words weren’t available in the Glove dictionary and for those,until now I have done random initialisation but that can be improved using semantic similarity concepts and Fasttext.

Also ,in these models,I have using zeros for padding and truncating of sentences and aspects which have size less than maximum length,but in order to improve accuracy,I aim to replace zeros with mean/mode.