Aspect Based Sentiment Analysis In Text Reviews

Yashaswi Tripathi

Aspect Based Sentiment Analysis In Text Reviews

DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Technology
in
Computer Science

by

Yashaswi Tripathi

[Roll No: CS-1825]

under the guidance of

Dr. Utpal Garain
Professor
Computer Vision and Pattern Recognition Unit

Dr. Debapriyo Majumdar
Assistant Professor
Computer Vision and Pattern Recognition Unit

Indian Statistical Institute
Kolkata-700108, India

July 2020

To my family and my guides

CERTIFICATE

This is to certify that the dissertation entitled **"Aspect Based Sentiment Analysis In Text Reviews"** submitted by **Yashaswi Tripathi** to Indian Statistical Institute, Kolkata, in partial fulfillment for the award of the degree of **Master of Technology in Computer Science** is a bonafide record of work carried out by him under our supervision and guidance. The dissertation has fulfilled all the requirements as per the regulations of this institute and, in my opinion, has reached the standard needed for submission.

Utpal Garain
Professor,
CVPR Unit,
Indian Statistical Institute,
Kolkata-700108, INDIA.

Debapriyo Majumdar
Assistant Professor,
CVPR Unit,
Indian Statistical Institute,
Kolkata-700108, INDIA.

Acknowledgments

I would like to show my highest gratitude to my advisors, *Dr. Utpal Garain* and *Dr. Debapriyo Majumdar*, Computer Vision and Pattern Recognition Unit, Indian Statistical Institute, Kolkata, for their constant support and encouragement. This work would not have been possible without their consistent motivation and timely guidance.

I would also like to thank *Debjyoti Paul* , M-Tech CS 2014-16 , Indian Statistical Institute, Kolkata, currently working as Data Scientist 2 at Amazon for his valuable suggestions and discussions.

I would like to thank my friends for continuous motivation and suggestions. Last but not the least, I am very much thankful to my family for their everlasting support.

Yashaswi Tripathi
Indian Statistical Institute
Kolkata - 700108 , India.

Abstract

Sentiment analysis plays an important role in e-commerce, as it allows the industries to better understand the customer experience and its brand value. Aspect Based Sentiment Analysis (ABSA) is a fine-grained version of sentiment analysis. ABSA not only focuses on analysing opinions in a given review but also looks into the several aspects and their sentiments thus giving a much clearer understanding. Aspect extraction is a crucial part of this ABSA task on which much attention has not been paid until recent years. Limited number of training data has made the task further challenging. This project addresses the problem of extraction of aspects from review comments and thereby attempts to improve the state of the art results in ABSA. For language modeling, BERT is used and it's finetuned on a novel *Neurosyntactic* model architecture. POS and dependency tags are used along review comments for extraction of aspect terms. Experiments conducted on SemEval dataset show that the proposed architecture achieves the state of the art results on the dataset.

Keywords: *ABSA, Aspect Detection, Neurosyntactic architecture, POS tags, DEP tags, BERT.*

List of Figures

List of Tables

Contents

Chapter 1

Aspect Based Sentiment Analysis

Sentiment Analysis [13] is an active field of research in Natural Language Processing and deals with opinions in text. Sentiment analysis is applied in multiple areas such as product reviews, customer feedback, political comments etc. Large organizations perform sentiment analysis to understand consumer experience and product reputation, monitor brand and analyse public opinion while Aspect Based Sentiment Analysis (ABSA) involves breaking down a review sentence into smaller fragments and thus providing a more granular and accurate insights of the review sentence. For example, consider this restaurant review : *"The food was great but the service was poor."* In this example there are more than one sentiment and more than one aspect (topic) in a single sentence, so to label the whole review as either positive or negative would not be correct. So here ABSA comes into picture which will extract the aspect terms from this sentence and then assigns sentiment polarity for each aspect. In this instance, Food and Service are two aspect terms and sentiment polarity associated with them are Positive ('The *food* was great") and Negative ("*Service* was poor") respectively.

Aspect term extraction and aspect polarity classification are two sub-tasks of Aspect Based Sentiment Analysis task.

1.1 Aspect Term Extraction

Aspect extraction (AE) a core task in ABSA, aims to find aspects on which reviewers have expressed opinions on [10]. Approaches used to extract aspects are finding noun phrases, using opinion and target relations, supervised learning and topic modelling. AE tasks are mainly classified into two categories with respect to the approaches taken, first is dictionary based or rule based approaches [21, 50, 40, 22] and second is deep learning based methods [20, 41]. In supervised learning, it is modelled as a

sequence labelling task, first a review is converted to separate tokens and then inferred whether the token belongs to any aspect. The tokens are labelled as one of Begin, Inside, Outside. A continuous chunk of tokens that are labelled as one B and followed by O's or more I's until O forms an aspect. In example 1.1 battery life is the aspect term.

The battery life was good
O **B** **I** O O

Figure 1.1: BIO Tagging.

1.2 Aspect Sentiment Classification

Aspect Sentiment Classification (ASC) is a subsequent task of AE, which aims to classify the sentiment polarity of the extracted aspect terms of the review sentence as positive, negative or neutral. The most commonly used deep neural network architectures for ASC task were recurrent neural networks (LSTMs) [37, 38, 33, 7, 14] and convolutional neural networks [47, 49, 5] until the introduction of BERT [6]. BERT based models and its variations achieved state of the art results on the aspect-based sentiment classification task.

1.3 Problem Statement

Given a review sentence our goal is to identify the aspects/aspect phrases of given target entities and the sentiment expressed towards each aspect. So, we divide our problem into two sub-problems first the aspect extraction and second aspect sentiment classification.

1.3.1 Aspect Extraction

Given a review with pre-identified entities (laptops or restaurants), identify the aspect terms present in the review sentence and return a list containing all the distinct aspect terms. For entity "laptop" *hard-disk, boot-time, battery-life* etc are the aspect terms while for target entity "restaurant" *food, service, location* are examples of the aspect terms.

Consider the following laptop text review :

Boot time *is super fast, around anywhere from 35 seconds to 1 minute.*

Given the review statement and target entity as *laptop* we need to extract the aspect term i.e. **boot time**.

1.3.2 Aspect Sentiment Classification

Given a review sentence with pre-identified aspect terms from the subtask 1 determine whether the polarity of each aspect term is positive, negative or neutral.

Consider the example :

I liked the service, but not the food

Here, our desired output is {service : **positive**; food : **negative**} where service is the aspect positive is the sentiment, food is the aspect negative is the sentiment.

1.4 Our Contribution

We are proposing three different simple BERT-based architectures which achieves comparable results to the state of the art results for the ABSA task :

- A Neurosyntactic model architecture for the task of finetuning-based aspect extraction. We leveraged Part of Speech and Dependency features of review sentences along with the review sentences to enhance the state of the art results on Vanilla BERT model.
- We also experimented on the feature-based approach model architecture, concatenating BERT, dependency and part of speech embeddings for the downstream task.
- A Neurosyntactic model architecture for the task of aspect sentiment polarity prediction. We leveraged the pre-trained aspect extraction model to get richer aspect embeddings as compared to Vanilla BERT .We finetuned a BERT base model and the pre-trained AE model parallelly to enhance the state of the art results on Vanilla BERT based aspect sentiment classification model.

1.5 Thesis Organization

The rest of the thesis is organized as follows. In Chapter 2, we discuss the previous related works in the field of aspect based sentiment classification and give a detailed

description of all the datasets used. In Chapter 3, we discuss the deep learning architectures and mechanisms used for the experiments. In Chapter 4 feature based approach is discussed followed by the final finetuning based models we are proposing in Chapter 5 for both Aspect extraction and Aspect Sentiment Classification task. Chapter 6 concludes our thesis and outlines future work followed by bibliography.

Chapter 2

Previous Work and Datasets

2.1 Previous Work

Saeidi et al. (2016) [26] introduced the task of targeted aspect based sentiment analysis (TABSA), to identify the fine-grained opinion polarity towards a particular aspect associated with a given target. The task has two sub steps: the first step is to determine the aspects associated with each target and the second step is to predict the polarity of aspects for a given target. The earliest work on (T)ABSA mostly depended on feature engineering [38, 12], followed by neural network-based methods [17, 42, 32, 34, 39] which achieved higher accuracy. In [30], several methods are explored for constructing an auxiliary sentence and transform (T)ABSA into a sentence pair classification task and finetune pre-trained BERT model to achieve the state-of-the-art results on the task. Here, the task is set as a 3-class classification problem, where the input is the sentence s, a set of target entities T and a fixed aspect set A = {general, price, transit, location, safety} to predict the sentiment polarity y = {positive, negative, none} over the full set of target aspect pairs {(t, a) : t belongs to T, a belongs to A} while in ABSA, the target-aspect pairs {t, a} becomes only aspects a.

In paper [45] a novel post-training approach on the popular language model BERT is explored to enhance the performance of fine-tuning of BERT for AE and ASC. Fine-tuning BERT directly on the end task that has limited tuning data faces both domain challenges and task awareness challenge. To enhance the performance of AE and ASC, we need to reduce the bias introduced by non-review knowledge (e.g., from Wikipedia corpora) that is used for pre-training BERT and fuse domain knowledge (DK) from unsupervised domain data and task knowledge from supervised Machine Reading Comprehensions (MRC) [25, 24] task but out-of-domain data.

To post-train on domain knowledge, the two novel pre-training objectives same as of BERT are used that is masked language model (MLM) and next sentence prediction (NSP). The former predicts randomly masked words and the latter detects whether two segments of the input are from the same document or not. MLM is crucial for injecting review domain knowledge for example, in pre-training in the Wikipedia domain, BERT learns to guess the [MASK] in "The [MASK] is bright" as *"sun"*. But in a laptop domain, it could be *"screen"*. While the objective of NSP encourages BERT to learn contextual representations beyond word-level. The total loss of the domain knowledge post training is given as

$$L_{DK} = L_{MLM} + L_{NSP} \tag{2.1}$$

where the loss function of MLM is represented as L_{MLM} and the loss function of next segment prediction be L_{NSP}. To post-train BERT on task-aware knowledge, we use SQuAD [25], which is a popular large-scale MRC dataset thus the resultant joint loss of post-training is given as

$$L = L_{DK} + L_{MRC} \tag{2.2}$$

Algorithm describes one training step where it takes one batch of data on domain knowledge (DK) D_{DK} and one batch of MRC training data D_{MRC} to update the parameters Θ of BERT.

Algorithm 1: Post-training Algorithm

Input: $\mathcal{D}_{\text{DK}}$: one batch of DK data;
$\mathcal{D}_{\text{MRC}}$ one batch of MRC data;
u: number of sub-batches.

$\nabla_{\Theta}\mathcal{L} \leftarrow 0$
$\{\mathcal{D}_{\text{DK},1}, \ldots, \mathcal{D}_{\text{DK},u}\} \leftarrow \text{Split}(\mathcal{D}_{\text{DK}}, u)$
$\{\mathcal{D}_{\text{MRC},1}, \ldots, \mathcal{D}_{\text{MRC},u}\} \leftarrow \text{Split}(\mathcal{D}_{\text{MRC}}, u)$
for $i \in \{1, \ldots, u\}$ **do**
  $\mathcal{L}_{\text{partial}} \leftarrow \frac{\mathcal{L}_{\text{DK}}(\mathcal{D}_{\text{DK},i}) + \mathcal{L}_{\text{MRC}}(\mathcal{D}_{\text{MRC},i})}{u}$
  $\nabla_{\Theta}\mathcal{L} \leftarrow \nabla_{\Theta}\mathcal{L} + \text{BackProp}(\mathcal{L}_{\text{partial}})$
end
$\Theta \leftarrow \text{ParameterUpdates}(\nabla_{\Theta}\mathcal{L})$

Figure 2.1: Post-training Algorithm. [45]

After this the obtained post-trained model is finetuned on the two end tasks Aspect Extraction(AE) and Aspect Sentiment Classification. For AE input is provided as a single sentence with m words constructed as x = $([CLS], x_1, \ldots\ldots, x_m, [SEP])$. After this we obtain h = Bert(x) and then apply a dense layer and a softmax for each position of the sequence, to predict labels of each token we take argmax at each position. For ASC task our input is in a form of sentence pair where the first sentence is the aspect term with m tokens and the second sentence is the review itself represented as x = $([CLS], q_1, \ldots, q_m, [SEP], d_1, \ldots\ldots, d_n[SEP])$. After h = BERT(x), we use the representations of [CLS] $h_{[CLS]}$, which is considered as the aspect-aware representation of the whole input and then apply softmax followed by argmax to predict the sentiment of the review sentence. For both the tasks cross entropy is used as the loss function. AE requires intensive domain knowledge for example, knowing that *"screen"* is a part of a laptop. Thus, post-training BERT on domain specific review is critical for AE. The paper attained state of the art results on Aspect term extraction task on SemEval dataset 2.2 on subtask 1 for both the domains laptop and restaurant that is a f1 score of 84.26 and 77.97 respectively. As a subsequent task of AE, aspect sentiment classification (ASC) predicts sentiment polarities as {Positive, Negative, Neutral} for extracted aspect terms also achieved state of the art results, Macro F1 score for laptop and restaurant domain are 75.08 and 76.96 respectively.

2.2 Datasets Description

SemEval (Semantic Evaluation) is an ongoing series of evaluations of computational semantic analysis systems. Aspect Based Sentiment Analysis (ABSA) was introduced as a SemEval task in 2014 (SE-ABSA14) Task 4 providing benchmark datasets of English reviews and a common evaluation framework [19]. The datasets were annotated with aspect terms (e.g. "hard disk", "pizza") and their polarity for laptop and restaurant reviews.

Two domain-specific datasets for laptops and restaurants, consisting of over 6K sentences with fine-grained aspect-level human annotations have been used for training. Benchmark restaurant domain dataset on subtask 1 (slot 2) of SemEval-2016 Task 5 and laptop domain dataset on subtask 1 of SemEval-2014 Task 4 with labelled aspect words are used to conduct the experiments in chapters 4 and 5.

2.2.1 Laptops reviews

This dataset consists of over 3K English sentences extracted from customer reviews of laptops.

	AE	ASC
Laptop	SemEval14 Task4	SemEval14 Task4
Training	3045 S./2358 A.	987 P./866 N./460 Ne.
Testing	800 S./654 A.	341 P./128 N./169 Ne.
Restaurant	SemEval16 Task5	SemEval14 Task4
Training	2000 S./1743 A.	2164 P./805 N./633 Ne.
Testing	676 S./622 A.	728 P./196 N./196 Ne.

Figure 2.2: Summary of datasets on aspect extraction (AE) and aspect sentiment classification (ASC). S: number of sentences; A: number of aspects; P:number of positive; N:number of negative; Ne: number of neutral polarities.

Dataset format

The sentences in the datasets are annotated using XML tags. The example 2.3 illustrates the format of the annotated sentences of the laptop's dataset.

```
</sentence>
- <sentence id="2225">
    <text>I also enjoy the fact that my MacBook Pro laptop allows me to run Windows 7 on it by using the VMWare program.</text>
  - <aspectTerms>
      <aspectTerm to="75" from="66" polarity="positive" term="Windows 7"/>
      <aspectTerm to="109" from="95" polarity="neutral" term="VMWare program"/>
    </aspectTerms>
</sentence>
- <sentence id="2644">
    <text>It's so much easier to navigate through the operating system, to find files, and it runs a lot faster!</text>
  - <aspectTerms>
      <aspectTerm to="60" from="44" polarity="positive" term="operating system"/>
      <aspectTerm to="88" from="84" polarity="positive" term="runs"/>
      <aspectTerm to="31" from="23" polarity="positive" term="navigate"/>
      <aspectTerm to="75" from="65" polarity="positive" term="find files"/>
    </aspectTerms>
</sentence>
- <sentence id="1365">
    <text>Purchased a Toshiba Lap top it worked good until just after the warrenty went out.</text>
  - <aspectTerms>
      <aspectTerm to="72" from="64" polarity="negative" term="warrenty"/>
    </aspectTerms>
</sentence>
```

Figure 2.3: Raw Laptops Dataset.

2.2.2 Restaurants reviews

The restaurants dataset consists of 350 review texts (2000 sentences) annotated with 2499 EA, OTE, polarity tuples where E stands for Entity, A stands for Attribute. E and A should be chosen from predefined inventories of entity types (e.g. RESTAURANT, FOOD, DRINKS) and attribute labels (e.g. PRICES, QUALITY,

STYLE_OPTIONS). OTE stands for Opinion Target Expression, it is defined by its starting and ending offsets. When there is no explicit mention of the entity, the slot takes the value "NULL". The possible values of the polarity field are: "positive", "negative", "neutral".

Dataset format

The sentences in the datasets are annotated using XML tags. The example 2.4 illustrates the format of the annotated sentences of the restaurant's dataset.

```
  </sentence>
- <sentence id="1004293:1">
      <text>We, there were four of us, arrived at noon - the place was empty - and the staff acted like we were imposing on them and they were very
    - <Opinions>
          <Opinion to="80" from="75" polarity="negative" category="SERVICE#GENERAL" target="staff"/>
      </Opinions>
  </sentence>
- <sentence id="1004293:2">
      <text>They never brought us complimentary noodles, ignored repeated requests for sugar, and threw our dishes on the table.</text>
    - <Opinions>
          <Opinion to="0" from="0" polarity="negative" category="SERVICE#GENERAL" target="NULL"/>
      </Opinions>
  </sentence>
- <sentence id="1004293:3">
      <text>The food was lousy - too sweet or too salty and the portions tiny.</text>
    - <Opinions>
          <Opinion to="8" from="4" polarity="negative" category="FOOD#QUALITY" target="food"/>
          <Opinion to="60" from="52" polarity="negative" category="FOOD#STYLE_OPTIONS" target="portions"/>
      </Opinions>
  </sentence>
- <sentence id="1004293:4">
      <text>After all that, they complained to me about the small tip.</text>
    - <Opinions>
          <Opinion to="0" from="0" polarity="negative" category="SERVICE#GENERAL" target="NULL"/>
      </Opinions>
  </sentence>
```

Figure 2.4: Raw Restaurants Dataset.

In the sentences of both datasets, there is an entry in the xml file for each occurrence of an aspect term. For example, if the previous sentence contained two occurrences of the aspect term *performance*, there would be two entries, which would be identical if both occurrences had negative polarity. If a sentence has no aspect terms, there is no entry in its annotations, and similarly for the aspect categories in the restaurant's dataset.

Chapter 3

Preliminaries

We will give overview of the major deep learning architectures, models and mechanisms that will be used in the later chapters.

3.1 Recurrent Neural Network

A recurrent neural network [28] is a generalization of feedforward neural networks where the output from previous step are fed as input to the current step. The traditional neural networks, failed in cases where the input size was not fixed and in cases like sentence translation where it requires information of the previous words to predict the current word and hence there is a need to remember the previous words. Thus, RNN came into existence, which solved this issue with the help of its internal state (memory) to process variable length sequences of inputs. RNN is recurrent in nature as it performs the same function for every input of data while the output of the current input depends on the past one computation. This makes them applicable to tasks such as unsegmented, connected handwriting recognition, machine translation, speech recognition etc.

3.2 Long Short Term Memory

The biggest disadvantage of recurrent neural network was gradient vanishing problem which was partially solved by using Long Short-Term Memory (LSTM) [8] networks, a modified version of recurrent neural networks by allowing gradients to also flow unchanged. A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. Intuitively, the cell is responsible for keeping track of the dependencies between the elements in the input sequence. The input gate controls the extent to which a new value flows into the cell, the forget gate controls the extent to which a value remains in the cell and the output gate controls the extent to which

the value in the cell is used to compute the output activation of the LSTM unit. The activation function of the LSTM gates is often the logistic sigmoid function. To sum it up, the cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell.

3.3 Bidirectional - Long Short Term Memory

Unidirectional Long Short-Term Memory (LSTM) networks only preserve information of the past because the only inputs it has seen are from the past. While in bidirectional LSTM (BiLSTM) [27] your inputs will run in two ways, one from past to future and one from future to past. The LSTM that runs forward preserve information from past and the LSTM that runs backwards preserve information from the future and using the two hidden states combined you are able in any point in time to preserve information from both past and future. Thus, they can understand the context better than unidirectional LSTMs.

3.4 Attention Mechanism

A Sequence-to-sequence model (seq2seq) [31] is a deep learning model that takes a sequence of items (words, time series, etc) and outputs another sequence of items. They have achieved a lot of success in tasks like text summarization, machine translation, and image captioning. The seq2seq model is composed of an encoder-decoder architecture (encoder and decoder both use some form of RNNs.) [4], where the encoder processes the input sequence and encodes/summarizes the information into a context vector of a fixed length. This representation is considered as a good summary of the entire input sequence. The decoder is then initialized with this context vector, using which it starts generating the transformed output. A critical and apparent disadvantage of this fixed-length context vector design is that it is not capable of remembering longer sequences. Often it forgets the earlier parts of the sequence once it has processed the entire sequence. The attention mechanism solves this problem by allowing the model to focus only on the relevant parts of the input sequence as needed. An attention model is different from a classic sequence-to-sequence model in two main ways [3]: First, the encoder passes a lot more data to the decoder. Instead of just passing the last hidden state of the encoding stage, it passes all the hidden states to the decoder. Second, an attention decoder does an extra step before producing its output. In order to focus on the relevant parts of the input in this decoding time step, the decoder does the following, first it looks at the set of encoder steps it received where each encoder states is most associated with a certain word in the input sequence, after this calculate a score for each hidden state using a predefined procedure and at last multiply each hidden state by its softmaxed score, thus amplifying

hidden states with high scores, and drowning out hidden states with low scores.

3.5 Transformers

The Transformer [36] model architecture relies entirely on an attention mechanism to draw global dependencies between input and abstains from recurrence. The Transformer allows for significantly more parallelization and is the state of the art in translation quality.

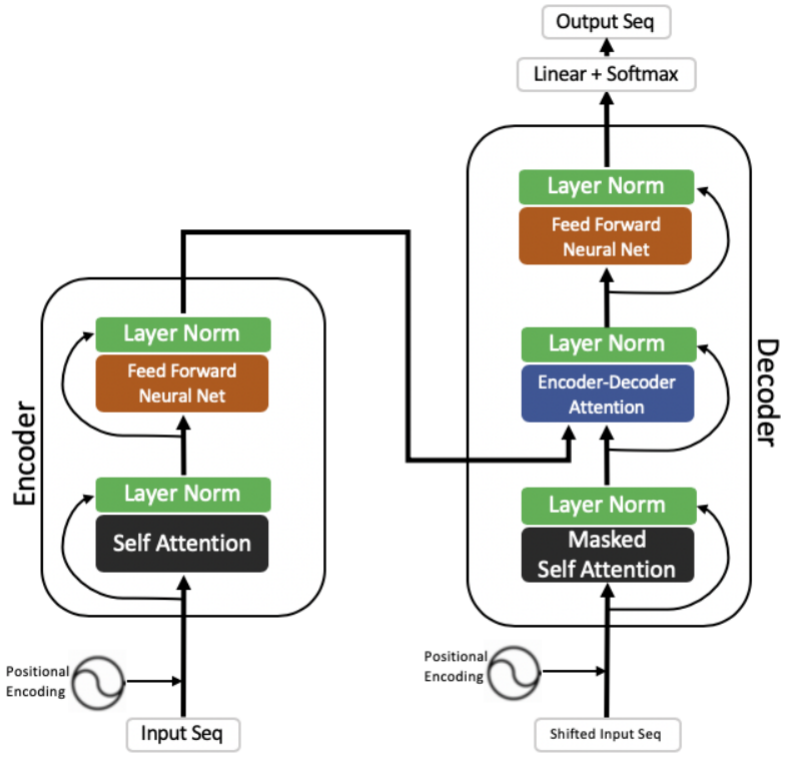

Figure 3.1: The transformer - model architecture. [36]

Transformer has an encoding component and a decoding component with connections between them. The encoding component is a stack of encoders similarly decoding component is stack of decoders of the same number as encoder. The paper [36] uses stack of 6 encoders and 6 decoders. Each encoder is identical and can be broken down into sub-layers, first is the self-attention layer and second is feed forward neural network. The decoder also has both the layers as in encoder but also has an attention layer between them that helps the decoder in focusing on relevant parts of the input sentence.

3.6 BERT

Recent advances in Natural Language Processing (NLP) have been dominated by the combination of Transfer Learning methods with large-scale Transformer language models. Bidirectional Encoder Representations from Transformers (BERT) [6] is an example of large-scale Transformer language model which is one of the most popular NLP approach to transfer learning.

BERT is a language representation model which can be pretrained on deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. As a result, the pre-trained BERT model can be finetuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without making substantial task specific architecture modifications.

BERT is a multi-layer bidirectional Transformer encoder. The two models introduced in the paper are BERT base- 12 layers (transformer blocks), 12 attention heads, and 110 million parameters and BERT large- 24 layers, 16 attention heads and, 340 million parameters.

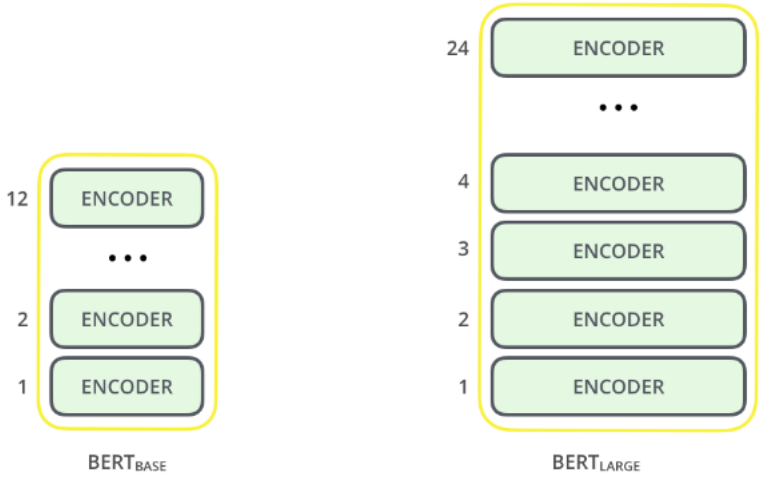

Figure 3.2: BERT models. [2]

BERT is pretrained on Masked Language Modelling (MLM) and Next Sentence Prediction (NSP) tasks. Language Modeling is the task of predicting the next word given a sequence of words while in MLM instead of predicting every next token, a certain percentage of input tokens is masked at random and only those masked tokens are predicted. Next sentence prediction task is a binary classification task in which, given

a pair of sentences, it is predicted if the second sentence is the actual next sentence of the first sentence.

The input representation used by BERT is able to represent a single text sentence as well as a pair of sentences in a single sequence of tokens. Each input sequence begins with [CLS] and ends with [SEP] token and in case a pair is fed then to separate, a [SEP] token is inserted in between both the sequence. BERT uses wordpiece tokenisation strategy to perform tokenisation. BERT handle OOV words by breaking them into sub -words present in its vocabulary. Maximum possible sequence length input for BERT is 512 tokens.

Feature-based and **Fine-tuning** are two existing strategies for applying pre-trained language representations to downstream tasks. Examples of feature-based approach, such as ELMo [18], include the pre-trained representations as additional features in a task-specific architecture while in the fine-tuning approach, such as the Generative Pre-trained Transformer (OpenAI GPT) [23], all the pre-trained layers along with the task-specific parameters are trained simultaneously.

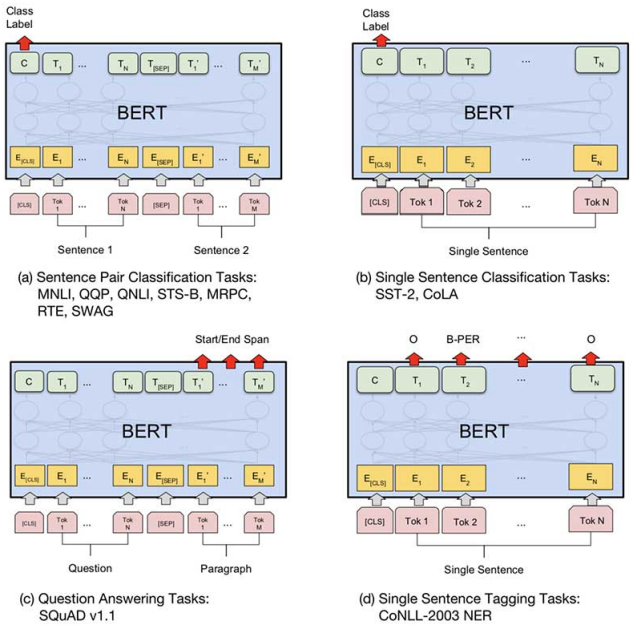

Figure 3.3: Architecture diagram and different fine-tuning techniques for BERT model. [6]

Self-attention mechanism in the Transformers makes fine-tuning straightforward thus, allowing BERT to model many downstream tasks. Finetuning procedure for sentence classification task involves the final hidden state of the [CLS] token taken as the

fixed-dimensional pooled representation of the input sequence and then is fed to the classification layer for sentence classification similarly in sequence tagging task, the final hidden states of every input token is fed to the classification layer to get a prediction for every token. For question answering task the question becomes the first sentence and paragraph the second sentence in the input sequence and two new parameters are learned during fine-tuning a start vector and an end vector indicating answer span.

3.7 Named Entity Recognition

Named entity recognition (NER) – also called entity identification or entity extraction is an information extraction technique that automatically identifies named entities in a text and classifies them into predefined categories. Entities can be names of people, organizations, locations, times, quantities, monetary values, percentages, and more.

An example of how NER works :

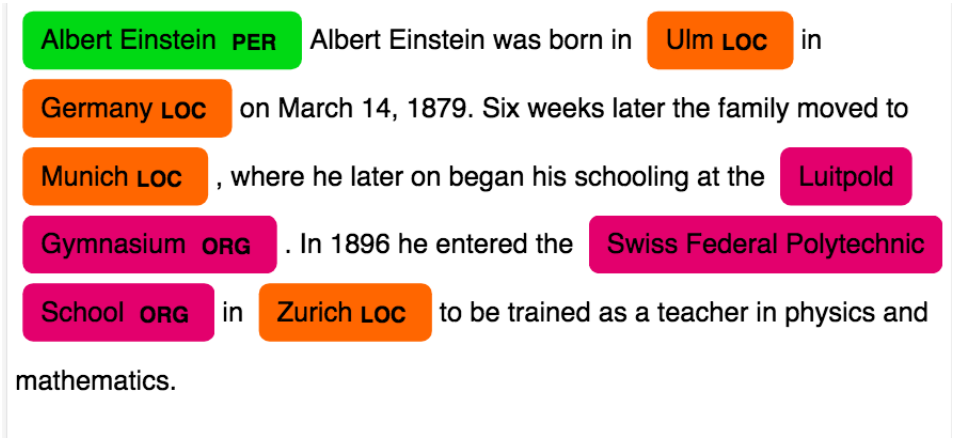

Figure 3.4: Example of NER.

NER systems take an unannotated block of text, as shown in example 3.4 and produce an annotated block of text that highlights the names of entities. In this example, a person name, two organization names and three locations have been detected and classifed.

Chapter 4

Feature Based Approach For Aspect Extraction

4.1 Motivation

In the BERT paper [6] under the section *Feature-based Approach with BERT* they have proposed a feature based approach for CoNLL-2003 Named Entity Recognition (NER) task [35]. Figure 4.1 shows the results obtained by applying the feature-based approach where the activations from one or more layers are extracted without fine-tuning any parameter of BERT then the obtained contextual embeddings are given as input to a randomly initialized two-layer 768-dimensional BiLSTM followed by a classification layer. It is observed that concatenating the embeddings obtained from the top four hidden layers of the pre-trained BERT, is only 0.3 F1 [6] behind fine-tuning the entire model. This experiment proved that BERT can also give good results for feature-based approaches as for the finetuning based approaches. We propose a feature based training architecture where along with the BERT contextual embeddings we concatenate the Part of Speech (POS) and Dependency (DEP) information of each word to extract the aspect terms.

Layers	Dev F1
Finetune All	96.4
First Layer (Embeddings)	91.0
Second-to-Last Hidden	95.6
Last Hidden	94.9
Sum Last Four Hidden	95.9
Concat Last Four Hidden	96.1
Sum All 12 Layers	95.5

Figure 4.1: Feature-based approach for NER task [6]

4.2 Architecture

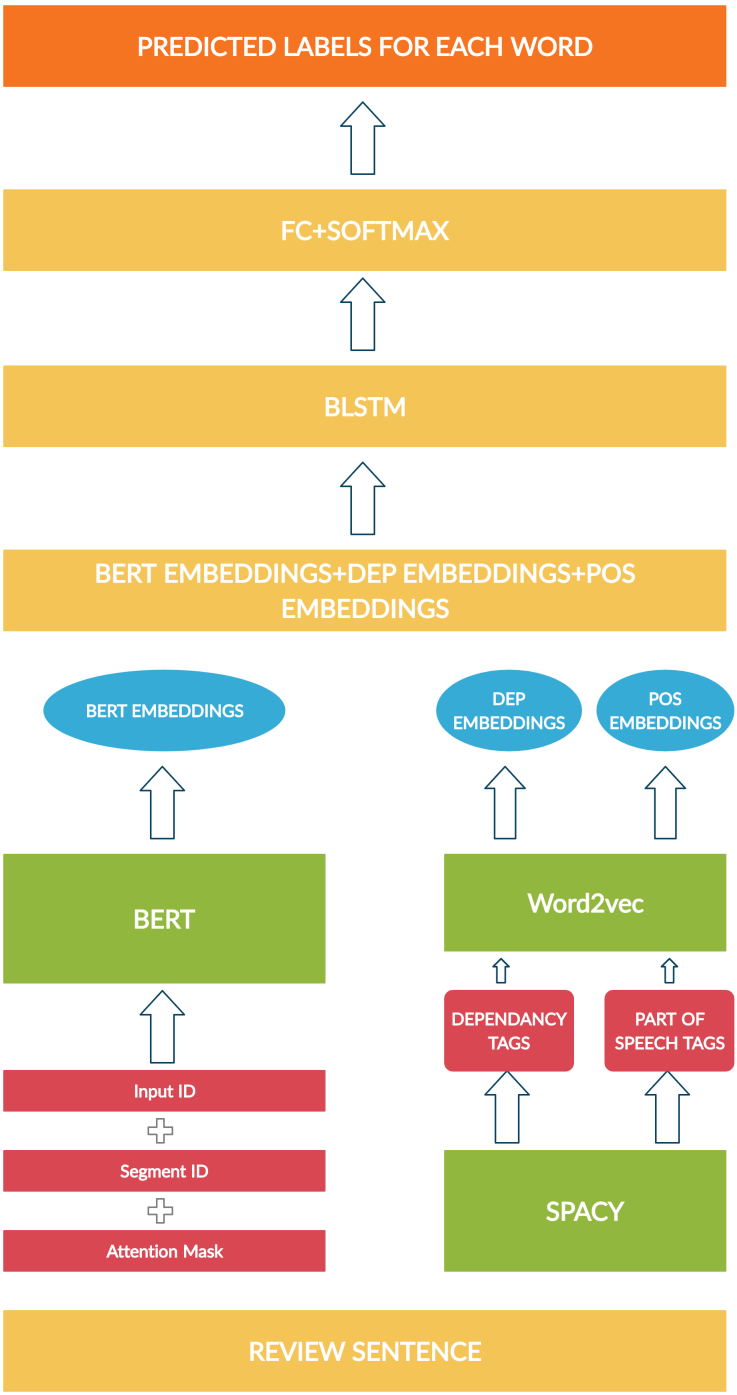

Figure 4.2: Feature based model architecture.

Details of the model architecture illustrated in Figure 4.2 are as follows :

- Review sentences are fed to BERT base consisting of 12 transformer blocks layers, 12 attention heads, and 110 million parameters.
- In parallel review sentences are also fed to spacy model [9]. In simple words spacy model is an english multi-task CNN trained on OntoNotes [43] which assigns context-specific token vectors, dependency parse, POS tags, and named entities.
- Word2vec [16] is a two-layer neural network, here we are using skip - gram architecture with window size of 3 and output POS and DEP embeddings of dimension 6 and 12 respectively.
- Two-layer 786 (768 + 12 + 6) dimensional BiLSTM followed by a classification layer (fully connected layer followed by softmax).

4.3 Method

4.3.1 BERT Embeddings

The input sentence with m words is constructed as x = $([CLS], x_1, \ldots\ldots, x_m, [SEP])$ where each x_i is obtained by passing a review sentence through a BERT base uncased tokenizer. Each above constructed sequence is encoded to produce three input embeddings namely input ids, segment ids and attention mask. Input ids correspond to the hash value of each token in the BERT base uncased vocabulary. Segment ids (token_type_ids) has a value of 0 corresponding to first sequence and 1 corresponding to second sequence in case of sequence pair input. Attention mask distinguishes real tokens with padding tokens and has value 1 for real tokens and 0 for padding tokens. These three are given as input to BERT for each sequence, after h = BERT(x) we get the embeddings from the last layer of BERT.

4.3.2 POS And DEP Embeddings

We will get POS tags and DEP tags for each review sentence using spacy's general purpose, small english model trained on written web text (blogs, news, comments). We will use the obtained list of POS tags and DEP tags for all the review sentences in the training set as training corpus to train two skip-gram Word2Vec models respectively. Thus, using the trained models we will get the POS and DEP embeddings for each token.

4.3.3 Concatenation

Concatenation here is a challenging task as tokenizer used for generating POS tags and DEP tags is different than the tokenizer used for generating BERT embeddings. Thus, the POS and DEP embeddings are not consistent with BERT embeddings. For example, consider the input sequence as *"we had champagne and caviar and felt like princesses"* when passed through BERT tokenizer outputs tokens ['we', 'had', 'champagne', 'and', 'ca', '## via', '## r', 'and', 'felt', 'like', 'princess', '## es'] (out of vocabulary words are broken into sub-words following ##) while when the same sentence is passed through spacy's tokenizer it outputs ['we', 'had', 'champagne', 'and', 'caviar', 'and', 'felt', 'like', 'princesses'] so as we can see that shape of BERT embedding for the sentence would be (12, 768). (12 indicates number of tokens produced by BERT tokenizer) and for POS and DEP embeddings shape would be (9, 6) and (9, 12) respectively (9 indicates number of tokens produced by spacy tokenizer). So, if we want to concatenate them, we need to make them consistent that is number of tokens must be same. To ensure the same we will only consider only the embeddings for very first token in case when a single word is broken down into multiple sub-tokens in both the cases i.e. spacy tokenizer or BERT tokenizer. This will make the number of tokens consistent for both the tokenizers that will be equal to number of words in a sequence. Thus, now we can concatenate the embeddings in order as BERT + DEP + POS. The resulting embedding thus obtained is of size 768 + 12 + 6 = 786.

4.4 Training

The concatenated embedding is passed as input to the BiLSTM neural network with two hidden states followed by a fully connected layer followed by softmax operation to obtain a probability distribution for the three classes {Begin, Inside, Outside}. We train the model for 10 epochs with train batch size equal to 16. Adam optimizer [11] is used with learning rate equal to 1×10^{-4}. Loss is calculated using Cross-entropy with parameter ignore index equal to the label of padding embeddings, so that padding doesn't makes any effect on model training that is model only learns labels for real tokens.

4.5 Results

Evaluation is reported as f1 scores calculated using official evaluation script [19]. The results are shown in the table 4.1 for laptop and restaurants datasets. The first row depicts results obtained by using BERT finetuning based approach for classification task while the second row depicts the results obtained when using concatenated em-

Domain	**Laptop**	**Restaurant**
Methods	**F1**	**F1**
BERT	79.28	74.1
Feature based approach(using DEP& POS Info.)	75.25	71.03

Table 4.1: Test Set Results (round off upto 2 decimal places) of experiments on feature-based architecture

beddings as mentioned in section 4.3.3 for the classification task. We got f1 scores of 75.25 and 71.03 for laptops and restaurant datasets which is close to the finetuning based approach, thus proving that feature based approaches are as good as finetuning based approaches for aspect extraction task.

Due to hardware constraint we could not use extract feature method of BERT to get the embeddings of last four layers and concatenate them (results are reported based on only last layer embeddings) which would have further improved the scores as mentioned in [6] for NER task.

Chapter 5

Neurosyntactic Model for Aspect Based Sentiment Analysis

5.1 Neurosyntactic Model for Aspect Extraction

5.1.1 Motivation

The architecture of BERT for token classification task as mentioned in [44] or Aspect extraction task as mentioned in [45] take a single sentence as input and output label for each token. While in a Question Answering task [6] a sequence pair is given as input to BERT with first sequence as Question and the second sequence as the context paragraph, the model predicts a start and an end token from the paragraph that most likely answers the question.

We are proposing a Neurosyntactic approach, where the syntactic information is enforced in neural model (BERT) and the existing relation of synactic and semantic in the neural model (BERT) is retrained with the explicit syntactic information provided through Part of Speech and Dependency embeddings. The proposed Neurosyntactic model utilise the question answering task BERT sequence pair model architecture where the input review sentence is mapped to it's dependency parsed string form (with DEP and POS) to better identify the aspect terms in review sentence using the sentence semantic and structural information. The proposed model gives very good results on SemEval datasets 2.2 as compared to Vanilla BERT model for Aspect Extraction task.

5.1.2 Architecture

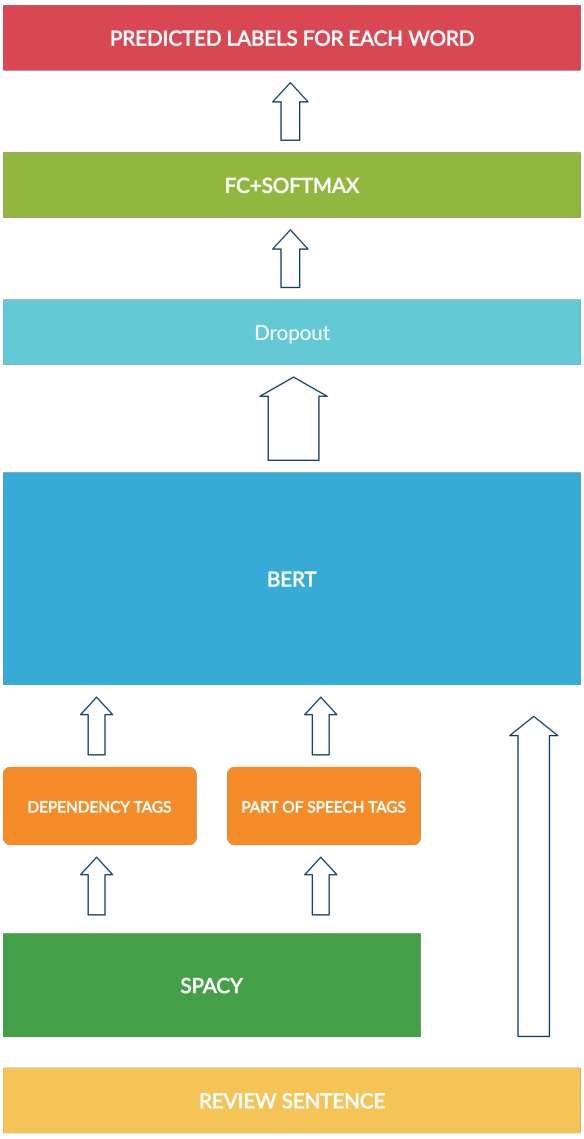

Figure 5.1: Neurosyntactic Model Architecture for AE.

The architecture of the Neurosyntactic model for AE is shown in Figure 5.1. The review sentence is passed through *spacy* model to obtain DEP and POS tags. These DEP and POS tags along with the review sentence is input to BERT base uncased model [44] as a sequence pair x = $([CLS], q_1, \ldots\ldots, q_m, [SEP], d_1, \ldots\ldots, d_n, [SEP])$, where $(q_1 \ldots\ldots\ldots q_m)$ now is DEP + POS tags of the review sentence and $(d_1, \ldots\ldots, d_n)$ is the review sentence containing that aspect. The BERT tokenizer encodes the sequence pair as input ids, segment ids and attention mask before inputting it to BERT

model. After h = BERT(x), followed by a dropout [29] layer we apply a dense layer and a softmax for each position of the sequence pair. Softmax is applied along the dimension of labels {B, I, O} for each position followed by taking argmax functions to predict labels.

5.1.3 Training

For aspect extraction task, sub-words except the first one beginning with ## in BERT is invalid in the labelling space, whereas NAACL paper [45] treats them as 'I'. We are following the same convention for our experiment. For example, the input is "we had champagne and caviar and felt like princesses" with labels as {O, O, B, O, B, O, O, O, O} where champagne, caviar are aspect terms for the entity restaurants. After BERT tokenisation we get tokens as ['we', 'had', 'champagne', 'and', 'ca', '##via', '##r', 'and', 'felt', 'like', 'princess', '##es'] now sub-words starting with ## don't have any labels so this becomes errant. Following the standard practice we label the starting sub-token of the aspect terms as 'B' and all the following sub-words (starting with ##) as 'I' while for not an aspect term label all sub-words as 'O'. So, the modified labels are {O, O, B, O, B, I, I, O, O, O, O, O}. The labels corresponding to DEP, POS tags and padding tokens are set as -1. While training we completely ignore tokens with label equal to -1. Thus, we make predictions only for the tokens of review sentence. In each epoch, the loss function is the averaged cross entropy calculated for all non-ignored indexes across all positions of a sequence given as

$$-\sum_{c} \mathbb{1}(X, l) \log(\Pr(l \mid X)) \tag{5.1}$$

where $\mathbb{1}(X, l)$ is the binary indicator {0 or 1} if class label l is the correct classification for X, and Pr(.) is probability that X is labelled as class l by our model. Here l belongs to set of all non-ignored class labels. The network is trained with dropout equal to 0.1 and Weight decay (form of regularization, after calculating the gradients) set as 0.01 and 0.00 [45] for the weight and bias parameters respectively. BertAdam [44] optimizer is used with learning rate equal to 3×10^{-5}. and warmup proportion equal to 0.1. We calculate learning rate for each step as :

$$lr\_this\_step = learning\_rate * warmup\_linear(global\_step/t\_total, warmup\_proportion) \tag{5.2}$$

where warmup_linear [45] is defined as :

```
def warmup_linear(x, warmup=0.002):
    if x < warmup:
        return x/warmup
    return 1.0 - x
```

Training is done for 8 epochs, (although training always converges by 3rd epoch) with batch size equal to 16. Total training time is 30 minutes on NVIDIA Tesla K80 GPU.

5.1.4 Results

The following table 5.1 show the results on laptop and restaurants test sets.

Domain	**Laptop**	**Restaurant**
Methods	**F1**	**F1**
DE-CNN [46]	81.59	74.37
BERT [6]	79.28	74.1
Neurosyntactic Model for AE	**81.0**	**75.84**
BERT-PT [45]	84.26	77.97

Table 5.1: Test Set F1 scores (round off upto 2 decimal places) for AE

We observed an improvement of **+1.72 F1** for laptop dataset and **+1.74 F1** for restaurants dataset as compared to BERT and an increase of **+1.47 F1** from DE-CNN for restaurants dataset using a simple architecture variation, without using sophisticated industry resources. Results are calculated using official evaluation script [19] averaged over 5 runs for 5 different seed values.

5.2 Neurosyntactic Model for Aspect Sentiment Classification

5.2.1 Motivation

Aspect based sentiment analysis provides more detailed information than general sentiment analysis, as it aims to predict the sentiment polarities of the given aspects in a review sentence. Aspect sentiment analysis is a second subtask of ABSA followed by aspect extraction. Traditional method for ABSA using BERT [45, 30] takes aspect term and review sentence as input pair for the classification task.

In this chapter, we would propose a novel Neurosyntactic model for ASC with parallel model architecture using the pretrained aspect extraction model along with BERT base uncased model to predict the aspect sentiment polarity. The intuition behind this architecture can be understood as follows. Improvement in AE task leads to improvement in ABSA task as AE is a core sub-task of the latter. So we use our pretrained Neurosyntactic AE model which has been proven to perform better than BERT (Table 5.1) for aspect extraction along with BERT to get improved results on the task.

5.2.2 Architecture

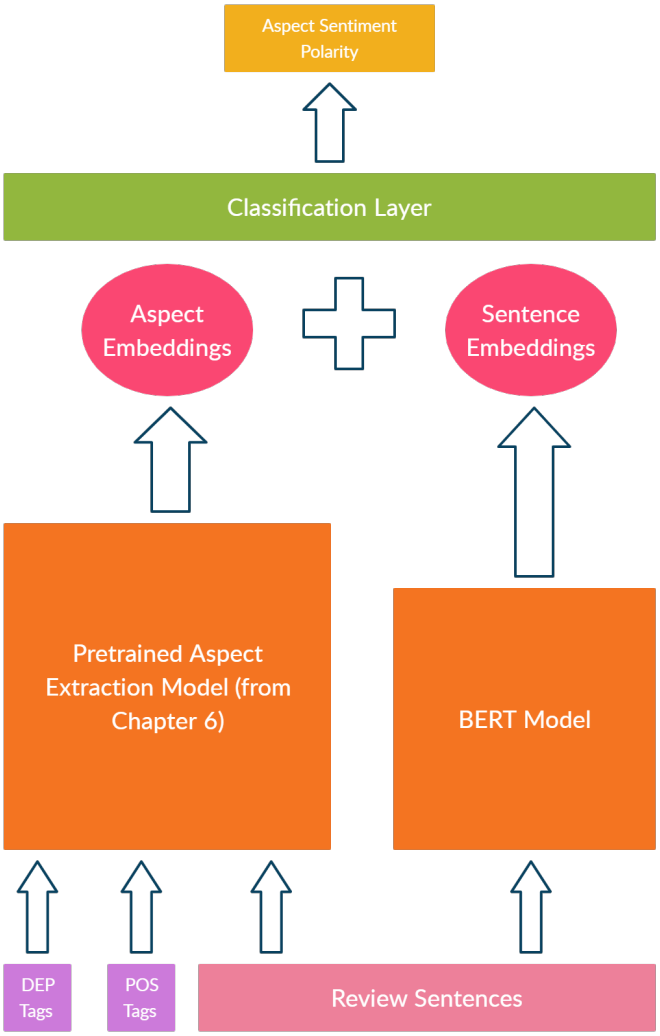

Figure 5.2: Neurosyntactic Model Architecture for ASC.

The architecture of the Neurosyntactic model for ASC is shown in Figure 5.2. We are training two models parallelly the first model is a pretrained aspect extraction model from section 5.1 which takes DEP and POS tags of the review sentence along with the review sentence as input while the second model is a BERT base uncased model for which input is review sentences. Followed by this is a classification layer with input size as 768 and output size equal to the number of labels {positive, negative, neutral} i.e. 3.

5.2.3 Training

Training is performed in a parallel fashion for both the BERT based models. For pretrained AE model we get the embeddings for given aspect word tokens of the given review sentence from the pre-classification layer and then sum them to get an embedding of size 768. For example, aspect term for a given review sentence is *appetizers* that will be broken into sub-tokens by BERT tokenizer as ['app', '##eti', '##zers'], so we need to get embeddings of all these aspect term tokens and sum them to get a representation for aspect embeddings. While in case of BERT base model we get the embeddings corresponding to the [CLS] token (it is the first token of the sequence) as a representation of sentence embedding. *"The first token of every sequence is always a special classification token ([CLS]). The final hidden state corresponding to this token is to be used as the aggregate sequence representation for classification tasks." (from the BERT paper [6]).* Both embeddings i.e. aspect embeddings and [CLS] embeddings obtained via two parallel models are summed and fed to classification layer (fully connected layer followed by softmax) to predict the sentiment polarity of a given aspect term in a review sentence. Both the parallel models along with the classification layer is finetuned with the loss function as cross entropy. Training is done with learning rate 3×10^{-5}. for 8 epochs with batch size of 32 for laptop dataset and batch size of 16 for restaurant dataset.

5.2.4 Results

Domain	**Laptop**		**Restaurant**	
Methods	**Acc.**	**MF1**	**Acc.**	**MF1**
BERT [6]	75.29	71.91	81.54	71.94
Neurosyntactic Model for ASC	**76.18**	**72.17**	**82.23**	**73.50**
BERT-PT [45]	78.07	75.08	84.95	76.96

Table 5.2: Test Set Results (round off upto 2 decimal places) for ASC task

The results for the aspect based sentiment classification is presented in Table 5.2. Training for the proposed architecture was not possible within given hardware support, so the results are presented using feature based approach for pre-trained AE model i.e. during the whole training pre-trained AE model is not finetuned. Given, the limitation then, also our model surpassed the MF1 scores of Vanilla BERT model by a good margin. There is an increase of **+0.89** and **+0.69** in Accuracy and an increase of **+0.26** and **+1.56** in Macro-F1 scores for laptop and restaurant domains respectively as compared to BERT. Thus, proving the efficiency of the model for the ASC task. Results would improve further with a good margin if trained as proposed in sub-section 5.2.3.

Chapter 6

Conclusion and Future Work

We proposed a Neurosyntactic model architecture for the task of aspect extraction and aspect sentiment classification. Dependency and Part of Speech information help us in better understanding syntax and structure of a given sentence, thus adding them as features to our model along with the review sentences resulted in better aspect extraction score. We are able to achieve a decent increase in F1 scores for both the domains as compared to Vanilla BERT. Our model is comparable to the current state of the art BERT-PT model [45]. BERT-PT is post-trained on 1,151,863 Amazon laptop reviews [15] and 700K reviews from Yelp reviews [1] for restaurant categories to increase domain knowledge, it is also trained on SQuAD 1.1 [25] that comes with 87,599 training examples from 442 Wikipedia articles to increase task awareness knowledge before finetuning on the downstream task while our model is not post-trained on any data. Thus, comparable results to the SOTA BERT-PT model shows a great advantage of our proposed model over it. We have used a machine with 8 GB RAM and core i5 intel processor along with free version of google Colab to run all the experiments. Due to limited resources we were not able to experiment with BERT Post Trained models, with larger training batch sizes, concatenated BERT embeddings of last four layers of BERT, larger versions of BERT, Neurosyntactic model for ASC (Parallel model architecture) proposed in section 5.2, which all could have lead to improvement in results.

Further work may include Post-Training BERT model with Dependency/POS tags and review sentence as input pair and then using the Post-trained model for aspect extraction task and experimenting with XL-Net [48], OpenAI GPT [23] as the pre-trained models.

Bibliography

[1] Yelp-dataset-challenge, https://www.yelp.com/dataset

[2] Alammar, J.: The illustrated bert, elmo, and co. (how nlp cracked transfer learning) (2018), http://jalammar.github.io/illustrated-bert/

[3] Alammar, J.: The illustrated transformer (2018), https://jalammar.github.io/illustrated-transformer/

[4] Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. CoRR abs/1409.0473 (2015)

[5] Chen, T., Xu, R., Wang, X.: Improving sentiment analysis via sentence type classification using bilstm-crf and cnn. Expert Systems with Applications (11 2016)

[6] Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)

[7] Fan, F., Feng, Y., Zhao, D.: Multi-grained attention network for aspect-level sentiment classification. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 3433–3442. Association for Computational Linguistics, Brussels, Belgium (Oct-Nov 2018), https://www.aclweb.org/anthology/D18-1380

[8] Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation 9(8), 1735–1780 (1997), https://doi.org/10.1162/neco.1997.9.8.1735

[9] Honnibal, M., Johnson, M.: An improved non-monotonic transition system for dependency parsing. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. pp. 1373–1378. Association for Computational Linguistics, Lisbon, Portugal (September 2015), https://aclweb.org/anthology/D/D15/D15-1162

[10] Hu, M., Liu, B.: Mining and summarizing customer reviews. pp. 168–177 (08 2004)

[11] Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization (2014), http://arxiv.org/abs/1412.6980, cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015

[12] Kiritchenko, S., Zhu, X., Cherry, C., Mohammad, S.: NRC-canada-2014: Detecting aspects and sentiment in customer reviews. In: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014). pp. 437–442. Association for Computational Linguistics, Dublin, Ireland (Aug 2014), https://www.aclweb.org/anthology/S14-2076

[13] Liu, B.: Sentiment analysis and opinion mining. vol. 5 (05 2012)

[14] Ma, D., Li, S., Zhang, X., Wang, H.: Interactive attention networks for aspect-level sentiment classification. CoRR abs/1709.00893 (2017), http://arxiv.org/abs/1709.00893

[15] McAuley, J., Yang, A.: Addressing complex and subjective product-related queries with customer reviews (12 2015)

[16] Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: Bengio, Y., LeCun, Y. (eds.) 1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings (2013), http://arxiv.org/abs/1301.3781

[17] Nguyen, T.H., Shirai, K.: PhraseRNN: Phrase recursive neural network for aspect-based sentiment analysis. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. pp. 2509–2514. Association for Computational Linguistics, Lisbon, Portugal (Sep 2015), https://www.aclweb.org/anthology/D15-1298

[18] Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. CoRR abs/1802.05365 (2018), http://arxiv.org/abs/1802.05365

[19] Pontiki, M., Galanis, D., Papageorgiou, H., Androutsopoulos, I., Manandhar, S., AL-Smadi, M., Al-Ayyoub, M., Zhao, Y., Qin, B., De Clercq, O., Hoste, V., Apidianaki, M., Tannier, X., Loukachevitch, N., Kotelnikov, E., Bel, N., Jiménez-Zafra, S.M., Eryiğit, G.: SemEval-2016 task 5: Aspect based sentiment analysis. In: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016). pp. 19–30. Association for Computational Linguistics, San Diego, California (Jun 2016), https://www.aclweb.org/anthology/S16-1002

[20] Poria, S., Cambria, E., Gelbukh, A.: Aspect extraction for opinion miningwith a deep convolutional neural network. Knowledge-Based Systems 108 (06 2016)

[21] Poria, S., Cambria, E., Ku, L.W., Gui, C., Gelbukh, A.: A rule-based approach to aspect extraction from product reviews. SocialNLP 2014 (01 2014)

[22] Qiu, G., Liu, B., Bu, J., Chen, C.: Opinion word expansion and target extraction through double propagation. Computational Linguistics 37(1), 9–27 (2011), https://www.aclweb.org/anthology/J11-1002

[23] Radford, A.: Improving language understanding by generative pre-training (2018)

[24] Rajpurkar, P., Jia, R., Liang, P.: Know what you don't know: Unanswerable questions for squad. CoRR abs/1806.03822 (2018), http://arxiv.org/abs/1806.03822

[25] Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P.: Squad: 100, 000+ questions for machine comprehension of text. CoRR abs/1606.05250 (2016), http://arxiv.org/abs/1606.05250

[26] Saeidi, M., Bouchard, G., Liakata, M., Riedel, S.: SentiHood: Targeted aspect based sentiment analysis dataset for urban neighbourhoods. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers. pp. 1546–1556. The COLING 2016 Organizing Committee, Osaka, Japan (Dec 2016), https://www.aclweb.org/anthology/C16-1146

[27] Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing 45(11), 2673–2681 (1997)

[28] Sherstinsky, A.: Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. CoRR abs/1808.03314 (2018), http://arxiv.org/abs/1808.03314

[29] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research 15, 1929–1958 (06 2014)

[30] Sun, C., Huang, L., Qiu, X.: Utilizing BERT for aspect-based sentiment analysis via constructing auxiliary sentence. CoRR abs/1903.09588 (2019), http://arxiv.org/abs/1903.09588

[31] Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. CoRR abs/1409.3215 (2014), http://arxiv.org/abs/1409.3215

[32] Tang, D., Qin, B., Feng, X., Liu, T.: Target-dependent sentiment classification with long short term memory. CoRR abs/1512.01100 (2015), http://arxiv.org/abs/1512.01100

[33] Tang, D., Qin, B., Feng, X., Liu, T.: Effective LSTMs for target-dependent sentiment classification. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers. pp. 3298–3307. The COLING 2016 Organizing Committee, Osaka, Japan (Dec 2016), https://www.aclweb.org/anthology/C16-1311

[34] Tang, D., Qin, B., Liu, T.: Aspect level sentiment classification with deep memory network. CoRR abs/1605.08900 (2016), http://arxiv.org/abs/1605.08900

[35] Tjong Kim Sang, E.F., De Meulder, F.: Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In: Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003. pp. 142–147 (2003), https://www.aclweb.org/anthology/W03-0419

[36] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I.: Attention is all you need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems 30, pp. 5998–6008. Curran Associates, Inc. (2017), http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf

[37] Vo, D.T., Zhang, Y.: Target-dependent twitter sentiment classification with rich automatic features. In: IJCAI (2015)

[38] Wagner, J., Arora, P., Cortes, S., Barman, U., Bogdanova, D., Foster, J., Tounsi, L.: DCU: Aspect-based polarity classification for SemEval task 4. In: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014). pp. 223–229. Association for Computational Linguistics, Dublin, Ireland (Aug 2014), https://www.aclweb.org/anthology/S14-2036

[39] Wang, B., Liakata, M., Zubiaga, A., Procter, R.: TDParse: Multi-target-specific sentiment recognition on twitter. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers. pp. 483–493. Association for Computational Linguistics, Valencia, Spain (Apr 2017), https://www.aclweb.org/anthology/E17-1046

[40] Wang, B., Wang, H.: Bootstrapping both product features and opinion words from Chinese customer reviews with cross-inducing. In: Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-I (2008), https://www.aclweb.org/anthology/I08-1038

[41] Wang, W., Pan, S.J., Dahlmeier, D., Xiao, X.: Coupled multi-layer attentions for co-extraction of aspect and opinion terms. In: AAAI (2017)

[42] Wang, Y., Huang, M., Zhu, X., Zhao, L.: Attention-based LSTM for aspect-level sentiment classification. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. pp. 606–615. Association for Computational Linguistics, Austin, Texas (Nov 2016), https://www.aclweb.org/anthology/D16-1058

[43] Weischedel, R., Hovy, E., Marcus, M., Palmer, M., Belvin, R., Pradhan, S., Ramshaw, L., Xue, N.: OntoNotes: A Large Training Corpus for Enhanced Processing (01 2011)

[44] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Brew, J.: Huggingface's transformers: State-of-the-art natural language processing. ArXiv abs/1910.03771 (2019)

[45] Xu, H., Liu, B., Shu, L., Yu, P.: 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). pp. 2324–2335. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019), https://www.aclweb.org/anthology/N19-1242

[46] Xu, H., Liu, B., Shu, L., Yu, P.S.: Double embeddings and CNN-based sequence labeling for aspect extraction. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). pp. 592–598. Association for Computational Linguistics, Melbourne, Australia (Jul 2018), https://www.aclweb.org/anthology/P18-2094

[47] Xue, W., Li, T.: Aspect based sentiment analysis with gated convolutional networks. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 2514–2523. Association for Computational Linguistics, Melbourne, Australia (Jul 2018), https://www.aclweb.org/anthology/P18-1234

[48] Yang, Z., Dai, Z., Yang, Y., Carbonell, J.G., Salakhutdinov, R., Le, Q.V.: Xlnet: Generalized autoregressive pretraining for language understanding. CoRR abs/1906.08237 (2019), http://arxiv.org/abs/1906.08237

[49] Zhang, Z., Zou, Y., Gan, C.: Textual sentiment analysis via three different attention convolutional neural networks and cross-modality consistent regression. Neurocomputing 275 (10 2017)

[50] Zhuang, L., Jing, F., Zhu, X.Y.: Movie review mining and summarization (01 2006)