BERT-based Multi-task Learning For Aspect-based Sentiment Analysis

by

Yesha Bhagat

A thesis submitted in partial fulfillment
Of the requirements for the degree of
Master of Science (MSc) in Computational Sciences

The Office of Graduate Studies
Laurentian University
Sudbury, Ontario, Canada

THESIS DEFENCE COMMITTEE/COMITÉ DE SOUTENANCE DE THÈSE

Laurentian Université/Université Laurentienne

Faculty of Graduate Studies/Faculté des études supérieures

Title of Thesis Titre de la thèse	BERT-based Multi-task Learning For Aspect-based Sentiment Analysis
Name of Candidate Nom du candidat	Bhagat, Yesha
Degree Diplôme	Master of Science
Department/Program Département/Programme	Computational Sciences
Date of Defence Date de la soutenance	January 20, 2022

APPROVED/APPROUVÉ

Thesis Examiners/Examinateurs de thèse:

Dr. Kalpdrum Passi
(Supervisor/Directeur(trice) de thèse)

Dr. Ratvinder Grewal
(Committee member/Membre du comité)

Dr. Ramesh Subramanian
(Committee member/Membre du comité)

Approved for the Office of Graduate Studies
Approuvé pour le Bureau des études supérieures
Tammy Eger, PhD
Vice-President, Research (Office of Graduate Studies)
Vice-rectrice à la recherche (Bureau des études supérieures)

Dr. Indranath Chaterjee
(External Examiner/Examinateur externe)

ACCESSIBILITY CLAUSE AND PERMISSION TO USE

Abstract

The Aspect Based Sentiment Analysis (ABSA) systems aims to extract the aspect terms (e.g., pizza, staff member), Opinion terms (e.g., good, delicious), and their polarities (e.g., Positive, Negative, and Neutral), which can help the customers and companies to identify product weaknesses. By solving these product weaknesses, companies can enhance customer satisfaction, increase sales, and boost revenues. There are several approaches to perform the ABSA tasks, such as classification, clustering, and association rule mining. In this research we have used a neural network-based classification approach. The most prominent neural network-based methods to perform ABSA tasks include BERT-based approaches, such as BERT-PT and BAT. These approaches build separate models to complete each ABSA subtasks, such as aspect term extraction (e.g., pizza, staff member) and aspect sentiment classification. Furthermore, both approaches use different training algorithms, such as Post-Training and Adversarial Training. Moreover, they do not consider the subtask of Opinion Term Extraction.

This thesis proposes a new system for ABSA, called BERT-ABSA, which uses Multi-Task Learning (MTL) approach and differentiates from these previous approaches by solving all three tasks such as aspect terms, opinion terms extraction, and aspect term related sentiment detection simultaneously by taking advantage of similarities between tasks and enhancing the model's accuracy as well as reduce the training time. To evaluate our model's performance, we have used the SemEval-14 task 4 restaurant datasets. Our model outperforms previous models in several ABOM tasks, and the experimental results support its validity.

Keywords: Aspect-Based Opinion Mining, Product Aspect and Opinion Extraction, Multi-Task Learning, BERT.

Acknowledgements

I can honestly say that working on these ideas was one of the most rewarding experiences of my life. It was a pleasure working on my research project with the guidance of my advisor, Dr. Kalpdrum Passi. He provided me the opportunity to select my research domain, and his continuous encouragement and guidance helped me to enhance my research work. The deep insight he brought to understanding research problems was extremely helpful to me, and I am extremely grateful for his encouragement. My sincere thanks go out to you for investing your valuable time in reading all the updates on my work. Working and talking with him was always a good experience. I would have been unable to accomplish this without his support.

Also, I would like to express appreciation to my thesis committee Internal Reader and External Reader, for their insightful comments that inspired me to complete this thesis, as well as for agreeing to serve as my thesis committee despite their busy schedules. It's a great honor to become a part of the Laurentian University. Additionally, I would like to give a special thanks to the other LU professors who guided me throughout the MSc program.

Finally, I would like to thank all my friends and roommates for their ongoing support, encouragement, and assistance. I am extremely grateful for the extended support they've shown me in the face of difficult circumstances.

Yesha Bhagat

Table of contents

List of Tables

List of Figures

Chapter 1
Introduction

Sentiment analysis is the use of Natural Language processing (NLP) and other data analysis techniques to analyze and derive the results from the raw text. Sentiment analysis is also known as Opinion Mining. Sentiment analysis has become extremely popular in recent years due to which enormous number of people have given their reviews about specific products or places on several platforms such as E-commerce web applications (e.g., Amazon) and social network platforms (e.g., Twitter). The goal of sentiment analysis is to extract people's opinions or sentiments (e.g., positive or negative) and subjectivity (subjective statements are those statements which contain opinion terms) from the texts (Do, Prasad, Maag, & Alsadoon, 2019). Also, user reviews can be written in different languages due to which we can perform sentiment analysis on multiple languages, thereby it is known as multilingual sentiment analysis. A large number of user's opinions are available about a particular place (e.g., hotel, restaurant), electronic products (e.g., laptop, phone, sound system) on different applications (e.g., TripAdvisor, Amazon). It is exceedingly difficult for a normal user to review/read all these available opinions and decide whether to visit a place or not, buy a product or not. This is where Sentiment analysis plays a savior role.

Sentiment Analysis can be performed at three levels: Document-level, Sentence level, and Aspect level. At the Document level, we assume that the whole user review expresses a sentiment towards a specific feature (Pontiki et al., 2016). However, it is not true in every condition due to which researchers look more closely at the sentiment analysis system by reviewing sentiment polarity for each sentence (L. Zhang & Liu, 2014). At the Sentence Level, the subjectivity of a sentence and opinion polarity can be checked for a complete sentence (Xue &Li, 2018). For example, given user review, “The pizza was good, but staff members were horrible to us”. The sentence level sentiment analysis extracts the neutral opinion polarity for the sentence because the opinion related to staff members is negative and the opinion related to the pizza is positive. Yet, it is not enough to precisely determine the user’s opinions by document level and sentence level opinion mining.

The Aspect based sentiment analysis (ABSA) focuses on the aspect term and term related opinion polarity (positive, negative, and neutral). In other words, instead of classifying the general opinion of the text as positive or negative, aspect-based analysis allows one to associate strong opinions with specific features of the product or service (Ma, Peng & Cambria,2018). The findings are more detailed, interesting, and reliable, as the aspect-based approach looks more closely at the details behind the text (Sharma, Nigam & Jain, 2014). For example, “The pizza was delicious, but the staff members were horrible to us." The sentence-level opinion mining system detects neutral opinion polarity for the above sentence, while Aspect based opinion mining system detects opinion polarity for each aspect term (pizza = positive, staff members = negative).

The Aspect-Based Opinion Mining (ABOM) task consists of three subtasks such as:

- Aspect term Extraction (ATE)
- Opinion term extraction (OTE) and
- Aspect term related sentiment polarity

1) Aspect Term Extraction: In this task, the aspect terms (e.g., features of the product) on which opinions have been expressed are identified. For example, a sentence is given: “The

pizza was delicious, but the staff members were horrible to us." The aspect terms for the above sentence are pizza and staff members, which will be extracted by the model to perform further tasks (Zainuddin, Selamat, & Ibrahim, 2018).

2) Opinion Term Extraction: In this task, the opinion terms (delicious, good, and horrible) expressed towards the aspect terms is identified. (Pontiki et al., 2014). For example, a sentence is given, "The pizza was delicious, but the staff members were horrible to us." The opinion term for the aspect term "pizza" is delicious and opinion term for the aspect term "staff members" is "horrible". This task is also known as Opinion Term Extraction. (Z. Li, Wei, Zhang, Zhang, & Li, 2019).

3) Aspect Term related Sentiment Polarity Detection: In this task, we will identify the sentiment polarity (Positive, Negative, and Neutral) expressed towards the aspect terms (Pontiki et al., 2014). For example, a sentence is given, "The pizza was delicious, but the staff members were horrible to us." The sentiment polarity for the aspect term "pizza" is Positive because the sentiment term for the pizza is "delicious", and sentiment polarity for the aspect term "staff members" is Negative because the sentiment term for the staff members is "horrible". This task is known as Aspect based Sentiment Analysis (ABSA) (Z. Li et al, 2019)

1.1 Benefits of Aspect Based Opinion Mining (ABOM)

Companies get the benefit from ABOM because it can identify the issues associated with each feature of the product based on the customer reviews. The ABOM automates processes such as customer support, allows them to sort and analyze customer data automatically, and provides powerful insights whenever needed. There is a large amount of feedback given by the customers than ever before. If a customer interacts with the company, either through a survey or comments, they often provide rich insights into what they are doing well and what they are doing wrong. Nevertheless, it can be challenging to wade through all that information by hand. As a result, aspect-based opinion mining handles the heavy lifting (Pascual, 2019). With aspect-based sentiment analysis (also known as ABSA), businesses can pinpoint aspects of a product or service that customers are complaining about and fix them in real time. Problems could be posed by customers as, “Is there any issue in a phone?”, “Is there a major bug in some new software?”, “Are customers getting angry about one particular service or product feature?”. Feature-based sentiment analysis can help to immediately identify these kinds of situations and take action to rectify them. (Pascual, 2019).

1.2 Problem Definition

The Aspect Based Sentiment Analysis (ABSA) mainly aims to identify the actual concern of the users on specific features of the products. Given set of reviews (e.g., pizza is great) about a product as input to the proposed approach, to identify the aspect terms and opinion term present in a review at a given time. Also, predict the opinion polarity related to each aspect term.

ATE, OTE, and ABSA: The aim of the tasks is to identify the phrases related to the product features (e.g., pizza and Food), opinion term related to each feature (e.g., great, and Delicious) and opinion polarity related each feature (e.g., positive, negative, or neutral) from the reviews.

1.2.1 Motivation

When spoken about social presence, web-based media is generally housed under the publicity office and is treated in almost a similar way to cliché board promotion. The major difference between them is that social media is way more interactive than billboards. There exactly lies the difference of power, in social media and billboard advertising. While it is difficult to engage in a conversation with a board hanging atop high towers, social platforms provide you with the aptitude to have direct communication with people in real time. Therefore, providing them an opportunity to explore what you bring to the table and bestowing yourself with the scope of new and returning engagers, irrespective of what your business structure looks like. With this ability, not only do you have a direct gateway to reach people, but so do they, to reach you; thus, imparting a 2-way reactive environment. This is exactly where people generally look upon, to provide or extract opinions, reviews and make a decision.

Restaurants and fine dining are a booming industry. In the 21st century of social media and public access, it is very vital that your social presence be impactful and informative. More than the aesthetically pleasing environment you create, both in your food business as well as your online existence, word of mouth plays a major role. Broadly, people trust a good word or a well-placed review online much more than just “trying - out” randomly, and why not? The world is now driven by feedback, be it customer reviews, social media mentions or simple survey results. Even if approaching an entirely new place or cuisine, majority of people will admit to checking the places’ online presence and the reviews on it to get a glimpse of what to expect. Therefore, to provide for even further concrete results and outputs in this industry, we approach a method that can constructively provide a breakdown of the subjective aspects and can further learn from the same to provide answers from its' learned database.

1.3 Objectives

The goal of this study is to improve upon earlier techniques to Aspect Based Opinion Mining by incorporating user feedback. By applying post-training to the BERT model, the BERT Post Training (BERT-PT) approach extracts the feature terms and each phrase's associated sentiment polarity, and the author builds the Review Reading Comprehension (RRC) model, which is a question-answering model in which the viewer can enquire, and the model will find the answer from the user's reviews. Furthermore, by using adversarial training on BERT models, the BERT Adversarial Training (BAT) technique finds the aspect words and each term's associated sentiment polarity. Several methods, such as BERT-LSTM and BERT-Attention, sought to employ different pooling algorithms in different ways to improve the precision of the BERT models. Our suggested method (BERT-ABSA) is based on Multi-Task Learning and three distinct knowledge pooling techniques. As our model (BERT-ABSA) takes advantage of learning the ATE, OTE and ABSA tasks together with LSTM and GRU pooling strategy, enhancing the learning of the model is more efficient.

1.4 Contributions

1. Our approach (BERT-ABSA) extracts aspect terms, opinion terms and aspect terms related opinion polarity simultaneously by applying Multitask Learning which can identify similarities and differences between the multiple tasks to improve the performance.

2. Also, our approach required less amount of time to train on data (e.g., half number of the epochs) as compared to previous BERT based approaches (BERT-PT (Xu, 2019), BAT (Karimi, 2021), and BERT-LSTM/Attention (Song, 2020)).
3. Our BERT-based approach achieves better results on all the tasks compared to the previous State-of-the-art models in terms of Macro F1-Score.

1.5 Thesis Organization

The rest of this thesis is organized as follows:

Chapter 2 - Related Work: Discusses the approaches using Multi-Task Learning to perform the Aspect Based opinion Mining, the existing approaches of BERT to perform Aspect Based Opinion Mining and comparisons between other studies in the same.

Chapter 3 - Methods: Discusses the techniques and approaches to data mining, Neural Networks, and different types of the same along with the proposed BERT-based Multi-Task Learning (BERT-MTL) for Aspect Based Opinion Mining (ABOM). This section also includes an example application of the proposed technique, a complete walk through of the same along with the comparison on previous BERT based system.

Chapter 4 - Experimental Evaluation: Discusses the experimental implementation for the proposed approach (BERT-MTL), the evaluation metrics and hyperparameters along with the required tools and technologies. Also, Dataset related information is described in this section. The results and its related discussion are also a part of this segment. Also provides the comparative analysis with the previous state-of-the-art approaches.

Chapter 5 – Conclusion and Future Work: Discusses the conclusion and reflects upon the feasible future work down the line.

References: Reflect on the existing research work and technology that has been used for the understanding and guidance

Chapter 2
Literature Review

The hours of relying upon a great product, service or administration are behind us, which is the explanation on why securing a wide scope of information is significant. Of course, quantitative analysis surely provides you with an overall glimpse of your image execution, however, the subjective contribution as text can give you information about how individuals actually "feel" towards your service. Sifting through all the available data, considered surveys, reviews, ratings etc. can be impossibly tedious for specific areas of focus. And doing so physically isn't feasible, while the subtleties of the opinion on the brand could be hard to determine. The answer to address this issue -Sentiment Analysis.

In Layman's terms, *sentiment analysis is the process of determining the opinion, judgement, or emotion behind natural language.* Any opinion one might have expressed, say, a comment on social media, an online survey answered, a product or service review on dedicated or mass platforms, etc. are all summarized by combing the same via sentiment analysis to provide an eagle's eye.

Purpose: Checking general opinion, doing statistical research, brand monitoring and dissecting client experiences; pertaining to the sheer scale of available data. Because of the widespread usage of social media and the large number of product reviews left by consumers, social networks and social media applications have grown in popularity over the last decade. To get an accurate picture of the same, of how many people like or dislike the services, is sentiment analysis used. Using an intelligent system of tools as Sentiment analysis, not only are we prone to rather accurate results, but we also avoid human errors, risk of bias and variations of perceptions. With these tools, one can be assured that the results fetched are coherent, all-inclusive and sans ambiguity, with so much as a click of a button. The scope of sentiment analysis is past tracking only the inspiration and cynicism in the text, but also; when made the most use of, with appropriate methods and classifiers; can provide the synonyms, polarities, and frequencies too.

Researchers attempted to extract user comments and determine each brand and its associated features reviews in order to determine the problems and concerns raised by users. We must do data cleaning and preprocessing operations on it after the data is collected to complete the analysis because user reviews that are made on social media contain unwanted words and noise. Clustering, classification, and association rule mining are some of the methods for identifying each aspect-related analysis from user evaluations. Most widely, Neural Network based classification approaches is used. In this chapter, several state-of-the-art models implemented to solve the problem of Aspect Based Opinion Mining and based on neural networks are discussed.

2.1 Multi-Task Learning (MTL) based approaches to perform Aspect Based Opinion Mining (ABOM)

The challenge of sparse data, when each task has a limited number of labelled data, was one of MTL's early motivations. There is limited labelled data for each task to effectively train a classified when there is a data sparsity problem, but with MTL, the labelled data is pooled from all the tasks during feature extraction to build a more accurate classifier (Misra, Shrivastava, Gupta, & Hebert, 2016). The MTL technique has the potential to lower the cost of the human labelling while also allowing current data to be reused. Deep MTL models have been shown to outperform single task models in studies (Y. Zhang & Yang, 2021). Furthermore, as more data is collected, MTL may train more resilient, universal, and powerful models, making it easier to share information, increase performance and reduce overfitting for each task (Y. Zhang & Yeung, 2012). Here, we will discuss some of the previous state-of-the-art approaches using MTL to perform the ABOM tasks.

2.1.1 Interactive MTL Network (IMN) (He et al., 2019)

The authors present an IMN network that can simultaneously recognise aspect phrases (such as pizza and service), opinion terms (such as good and tasty), and term-related sentiment polarity (such as positive, negative, or neutral) in user evaluations. They've also created a message-passing mechanism that would convey data from task-specific levels to shared layers shared by all tasks, allowing them to interact. The authors (He et al., 2019) perform above mention tasks using the sequence labeling in which every word has an associated label. As we can see in Table 2.1, the dataset for the IMN approach is given.

Problem Identified: Consider the Table 2.1 as input to the IMN approach, which uses user reviews to identify aspect terms, opinion terms, and opinion polarity from each review simultaneously.

Input: To train the model, user reviews and each task-related target value will be given to the neural network, and during testing, only user reviews will be given as input.

Step 1: To begin, the sentence will be translated into tokens, and each token-related word embedding vector will be generated using the Word2Vec technique.

Table 2.1 Dataset for IMN in ATE and OTE and Fine-Grained ABOM

User's Reviews	Target in ATE and OTE	Target in Fine-grained ABOM
The food was Delicious	[0, BA, 0, BP]	[0, POS, 0, 0]
The staff members were bad	[0, BA, IA, IA, 0, BP]	[0, NEG, NEG, NEG, 0, 0]
The service was excellent	[0, BA, 0, 0]	[0, POS, 0, 0]

The Convolutional Neural Networks will then be provided each vector associated to the token (CNN).

Step 2: Each token's vectors will be processed using CNN. CNN shall be performed convolutional and pooled operation on each token vector.

Step 3: As a result of the convolution operation in CNN, a feature map is created, which will be used as input to the pooling function. The greatest value from the feature map will be extracted as a result of a certain range of windows in the pooling procedure. For example, we go through a 2*2 matrix and input values will be (3, 7, 12, 10) for pooling operation. It will identify the maximum value from the input (3, 7, 12, 10) and placed it as output in a matrix such as 12.

Step 4: The output of the pooling operation will be given as input to the two different CNN and FFNN blocks which will convert the vector into class probabilities.

Step 5: The predicted values from the CNN and FFNN block with actual target values were given to the Cross Entropy Loss function.

$$Loss = -\sum t_i \cdot log(p_i)$$

Step 6: All the losses related to ATE, OTE, and ABOM tasks were summed to generate the final loss which will be given to the backpropagation algorithm to reduce it and change the weights of the model.

2.1.2 Semi-Supervised MTL framework for Aspect Based opinion Mining (ABOM) (SEML) (N. Li et al., 2020)

The SEML is a semi-supervised Multi-Task Learning framework that uses a deep learning technique to find Aspect terms and their polarities of opinion. The researchers of the SEML model consider activities like Aspect term extraction and Fine-grained Aspect based Opinion Mining to be sequence labelling challenges. The goal of this study is to employ a semi-supervised strategy to do Multi-Task Training in which not all user reviews have matching objective values. Furthermore, the authors extend the GRU network by incorporating attention into each GRU cell to determine sentence context. The user reviews are given as input to the SEML approach, which uses the user reviews to identify aspect terms, and opinion polarity from each reviews simultaneously.

Step 1: As we can see in the Figure 2.1, the Word2Vec method will create each token-related word embedding vector from the user reviews. If a word, such as liquid, water, drink, and many more, is used in a different context, the Word2Vec method creates a comparable vector for it. Following that, character characteristics generated by character level Convolutional Neural Networks will be concatenated with each vector associated to the token (CNN).

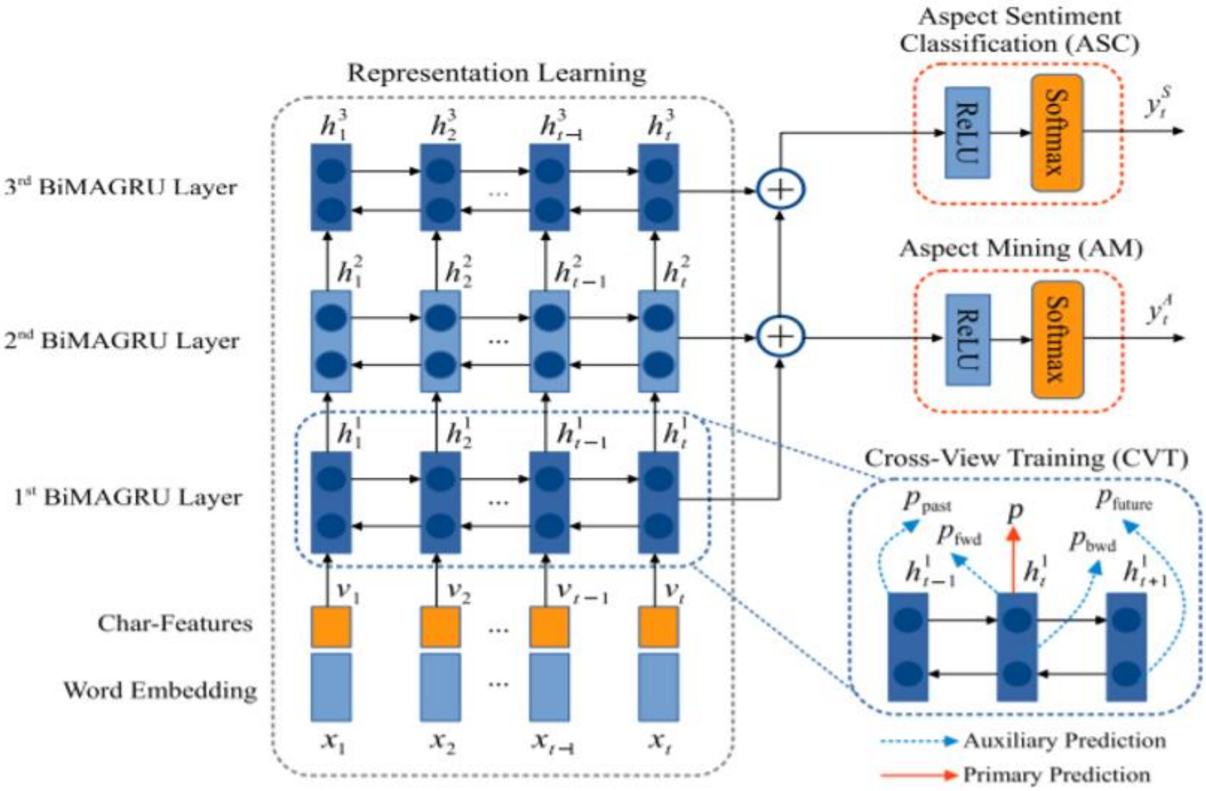


Figure 2.1: SEML for ATE and ABSA (N. Li et al, 2020)

Step 2: The vectors of each token will be processed through the Bi-directional Moving-window Attentive GRU (BiMAGRU). In BiMAGRU, the authors attached the attention gate in the GRU network with reset and update gate. The output of the Update and reset gate will form a new candidate memory in each GRU cell that new memory will be given to the attention gate to encode past nearby significance information.

Step 3: The authors build three layers of the BiMAGRU network, and the first layer performs the Multi-Task Cross-view Learning. The Second Layer performs the Aspect extraction task, and the third layer performs the opinion polarity detection task.

Step 4: The output of the second and third layers of the BiMAGRU network will be given to the softmax layer, which will be worked as a classification layer to perform ABOM tasks. It generates the prediction probability for each task.

Step 5: The prediction value from the softmax layer will be given to the loss function with the actual target values.

Step 6: After that, the calculated loss will be given to the Back-propagation algorithm to changes the weights of the model. The above steps will be repeated several times until the maximum epoch condition reach.

2.2 BERT based approaches to perform ABOM

2.2.1 BERT Adversarial Training (BAT) (Karimi et al., 2020)

The authors used an adversarial training technique to extract aspect phrases and terms linked to opinion polarity from user reviews in the BAT model. The authors claimed to be the first to use the BERT approach for adversarial training in ABOM problems. Adversarial examples are constructed using perturbation equations that are passed to the BERT Encoder function to execute adversarial training. Each token is included in a user review to create the hostile instances.

According to Figure 2.2, BAT ran the input through all the BERT model's layers, then calculated LOSS1 on which the perturbation was made, and then included in the embedding vector to create adversarial samples.

Steps of the BAT for ATE

1. Dataset contains sentences or reviews, and each sentence has target values. For example, "The pizza is good" and the target value for a sentence is [0, 1, 0, 0, 0] which denotes term "pizza" to determine the aspect term from the sentence.

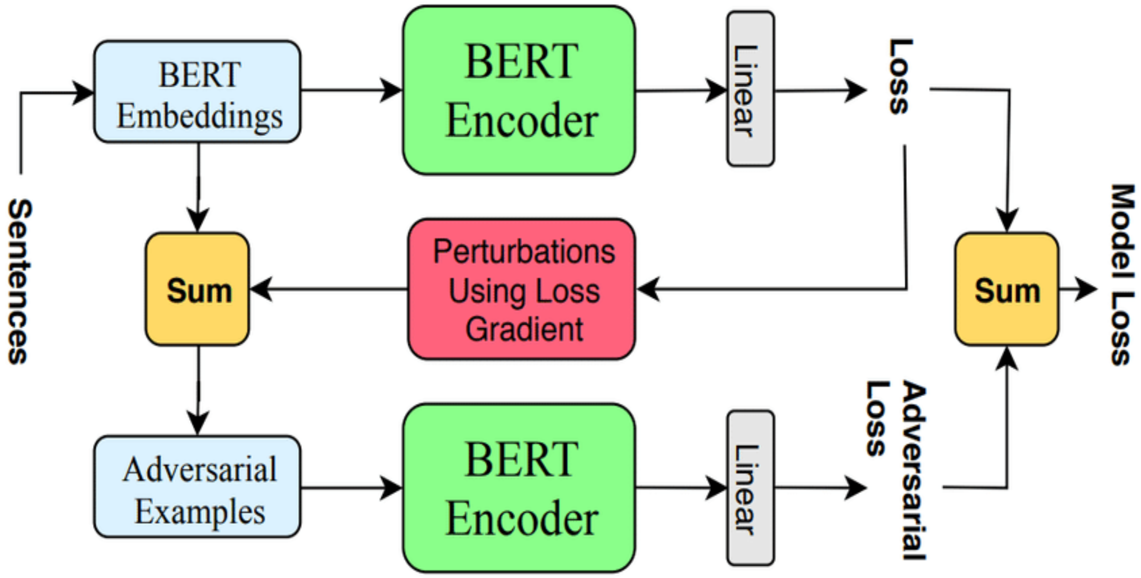


Figure 2.2: BAT for ATE and Fine-grained ABOM (Karimi et al., 2020)

2. First, the sentences are given to the BERT Tokenize Function to generate the tokens, Input Id, Attention Mask, token type ids. The above-mentioned Input Id, Attention Mask, and token type ids will be passed to the BERT Embedding Function, which will turn each lexicon into its own embedding vector. Similar words in a corpus are closer to one another, and similarity is

determined by context. For each token in the input, the embedding represents the word as a higher-dimensional vector, such as a 768-dimensional vector.

3. The embedding vector for each token is given into the BERT Encoder Function, which generates a 768-dimension vector for each sentence token. The BERT output is fed into the FFNN, which performs sequence labelling to extract aspect terms and converts the vectors into probabilities for each class. Every word has a label in sequence labelling; for example, the token pizza has an actual target of [1,0,0], where one at the first place signifies the start of the aspect term, and the model projected value for word pizza is [0.6, 0.1, 0.2].
4. The class probabilities generated by the FFNN and their respective actual target from the dataset will be given to the loss function where a loss will be calculated.
5. For sequence labeling task Cross-Entropy Loss function is used. The Cross-Entropy Loss can be calculated by below equation:

$$Loss = -\sum t_i \cdot log(p_i)$$

6. After the calculation of loss for ATE task, the loss will be given to the perturbation equation to generate the adversarial example.
7. After that, adversarial examples are processed through the BERT encoder function and FFNN same as above then adversarial loss will be calculated using below equation:

$$ADV.Loss = -\,log\,P(y|x + a\,dv \cdot example)$$

8. After that, adversarial loss and loss will be summed to generate the final loss which will be reduce by applying the back-propagation algorithm.

The above steps are repeated to perform the Fine-grained ABOM in BAT.

2.2.2 BERT-LSTM /Attention (Sun et al., 2019a)

To improve efficiency and prevent information loss during the processing of user reviews, the authors created two different knowledge pooling algorithms and applied them to each BERT Encoder layer. As pooling solutions, the authors (Sun et al., 2019a) chose a Uni-directional LSTM Network and a dot product attention network, both of which are capable of processing sequential data.

According to the Figure 2.3, the outcome of each encoder layer is sent to the LSTM pooling layer, and the fully connected layer, also known as FFNN, with softmax activation function, serves as the classification layer. The user review and aspect term are denoted by sentences 1 and 2 in the illustration, respectively.

1. Dataset contains sentences or reviews with aspect, and each (sentence, aspect) pair have targets. For example, (“The pizza is good.”, pizza) pair will be input, and the target value for a pair is “positive” to determine the opinion polarity from the sentence and aspect.
2. First, the (sentence, aspect) pair is given to the BERT Tokenize Function (discussed in above section 3.1.1) to generate the tokens, Input Id, Attention Mask, and to-ken type ids. For example, if task is Fine-grained then: tokens = [‘[CLS]’, ‘the’, ‘pizza’, ‘is’, ‘good’, ‘.’, ‘[SEP]’, ‘pizza’, ‘[SEP]’]

Input Id = [101, 1996, 10,733, 2003, 2204, 1012, 102, 10,733,102]

Attention Mask = [1, 1, 1, 1, 1, 1, 1, 1, 1]

token type ids = [0, 0, 0, 0, 0, 0, 0, 1, 1]

3. The above generated Input Id, Attention Mask, and token type ids will be given in BERT Embedding Function to convert each token into respective vectors called as embedding vector.

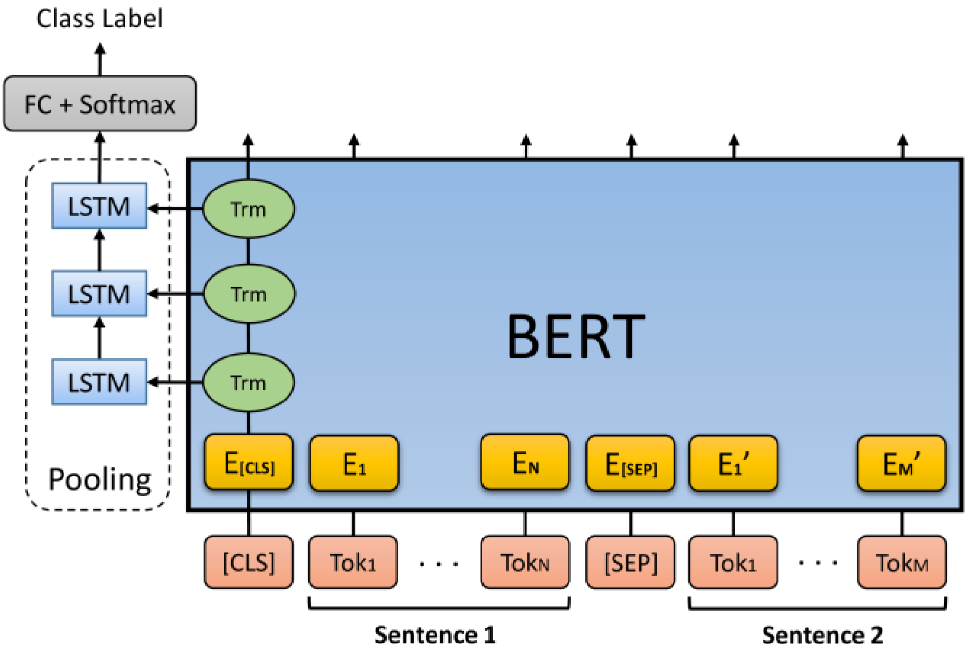


Figure 2.3: BERT-LSTM for Fine-grained ABOM (Sun et al. 2019)

4. The embedding vector related to each token fed to the BERT Encoder Function (discussed in section 3.4.1) which will generate a 768-dimension vector for each token of the sentence. In encoder, embedding vector processed through the Attention, FFNN, and Normalization layer.

5. Each token related a 768-dimension vector will be given to the pooling strategy. The pooling strategy is another neural network attached to the BERT Encoder function to identify the context and generate the final result. The authors used LSTM and Dot product Attention as pooling layer. As we can see in table, the output from pooling strategy will be given.
6. The output of the pooling Strategy given to the FFNN (discussed in section 3.2.1) to perform the classification task, such as multi-class classification to detect aspect term related opinion polarity, where it converts the vectors into each class probabilities.
7. The class probabilities generated by each FFNN and their respective actual target from the dataset will be given to the Cross Entropy Loss function where a loss will be calculated. In Multi-Class classification, the review has a label of [1,0,0] where one at first position denotes the opinion polarity of aspect (pizza) is positive, and the predicted value by the model is [0.6, 0.1, 0.2].

   The Cross Entropy Loss can be calculated by the below equation:

   $$Loss = -\sum t_i \cdot log(p_i)$$

   Where t_i denotes target value and P_i denotes the predicted value. For example, target value for sentence is [1; 0; 0] and predicted value for that sentence [0:6; 0:1; 0:2]. These two arrays will be given as input to the cross-entropy loss.
8. After that, to reduce the loss of the task, the backpropagation algorithm was used to change the weights of the approach.

2.3 Other studies in Aspect based Opinion Mining (ABOM)

The researchers have done an extensive amount of research to mine certain kinds of information (e.g., aspects, opinion polarity) from reviews to build a real-time application (Miao, Li, Wang, &

Tan, 2020). Some of the past state-of-the-art models in the Aspect-Based Opinion Mining (ABOM) systems are given below.

The sentence or User Review: “The pizza was delicious, but the staff members were horrible to us.” The sentence will be given to the BERT tokenize function and output of the tokenize will be processed through the BERT encoder and different pooling layers. At the end, classification layer shall be attached to the model to convert vectors into each class related probabilities.

MGAN: The Multi Granularity Alignment Network (MGAN) can distinguish between opinion polarity related to aspect categories and opinion polarity related to aspect terms. The author created the Coarse2Fine Attention module to accomplish both tasks, which can transfer aspect knowledge to coarse-grained and fine-grained networks. The authors also tweaked the SemEval 2014 dataset in order to assess the model’s performance. The MGAN model, for example, can accept a sentence with an aspect term or aspect category as input (e.g., text and pizza) and forecast the category or term that hasn’t been given as input, as well as predict the opinion polarity connected to aspect term (pizza has a positive opinion) and category (food has positive opinion polarity). To solve the tasks, the authors combined three distinct attention modules with Bi-directional LSTM and Word2Vec (a word embedding module that converts words into vectors).

BERT-Post Training: To complete the ABSA challenge, the authors created three distinct BERT-based models. They utilise BERT to conduct Named Entity Recognition to extract aspect words from user reviews in the first model. For instance, using the above-mentioned text as input to the model (ATE), forecast aspect words such as pizza and staff members. For the aspect terms, the second model uses a classification task to predict opinion polarity. For instance, input the above-given text and extracted aspect term to the model to determine opinion polarity linked to the aspect words included in the sentence, such as pizza has good opinion and staff personnel has a negative opinion. The final model (RRC) is unusual in that it is based on question responding, in which the user asks a question, and the BERT model predicts the answer based on the user evaluations. For example, the above text with user queries (which is the opinion word?) is fed into the RRC model to detect the presence of an opinion term in a sentence like tasty and awful. They also suggested a Post Training algorithm for all three BERT-based methods to be trained.

DomBERT: The paper's major objective is to present a BERT model for ABSA tasks that can discover specific domain (e.g., laptop, restaurant) relevant sentences from a vast pool of domains using random sentences. Domain-oriented BERT (DomBERT) also performs an ABSA task on data from a given domain. For instance, phrases like 1) The sky is clear and (Domain: Astronomy). 2) The concept is simple and straightforward (Domain: Concepts). 3) the water is very clear (Domain: Liquids). 4) The screen is crisp and clear (Domain: Laptop). To execute aspect extraction and Fine-Grained ABSA tasks, the DomBERT extracts the sentence relevant to the laptop or restaurant domain.

Snippext: The Fine-grained ABOM and ATE tasks are performed using a BERT-based semi-supervised method. The snippext model is also utilised in real-time hotel management systems to determine user opinion. The authors created two semi-supervised algorithms, MixDA and MixMatch, that employ four distinct processes to produce related phrases from user reviews. To create new user reviews from current user reviews, utilise the Replace, Swap, Delete, and Insert procedures. The BERT model and FFNN are provided all freshly produced and current user reviews to complete the ABOM tasks.

BERT for E2E ABSA: The BERT for End-to-End Aspect Based Sentiment Analysis model perform ATE tasks and ABSA task using the sequence labeling as shown in below Figure 2.4. The authors (X. Li et al., 2019) used BERT based pretrained model and use different neural network on top of the BERT such as Linear, GRU, CRF, and many others and compare the performance of the approach with state of the are approaches. In below Figure 2.4, the architecture of the BERT model has been displayed and in the end of the model authors used E2E ABSA layers to check which layer perform best. In the end, the prediction of the model displayed as y. In the evaluation BERT with the Self Attention model works better than other approaches on restaurant dataset. performed using a BERT-based semi-supervised method. The snippext model is also utilised in real-time hotel management systems to determine user opinion. The authors created two semi-supervised algorithms, MixDA and MixMatch, that employ four distinct processes to produce related phrases from user reviews. To create new user reviews from current user reviews, utilise the Replace, Swap, Delete, and Insert procedures. The BERT model and FFNN are provided all freshly produced and current user reviews to complete the ABOM tasks.

However, using Neural Machine Translation and Memory Networks, various methods to ABOM problems exist. Various types of learning, such as Unified learning to extract aspect term and term related opinion polarity, Multi-Task Learning to extract opinion term and aspect term, and Multi-view Learning, were also used on neural networks to find the aspect term, opinion term, and term related opinion polarity all at the same time. In next sections, we have compared different surveys about Aspect based sentiment analysis, Multi Task Learning, and why BERT embeddings are more useful than Word2vec model which will indicate the importance of the task we have solved and identify how Multi-Task Learning is useful.

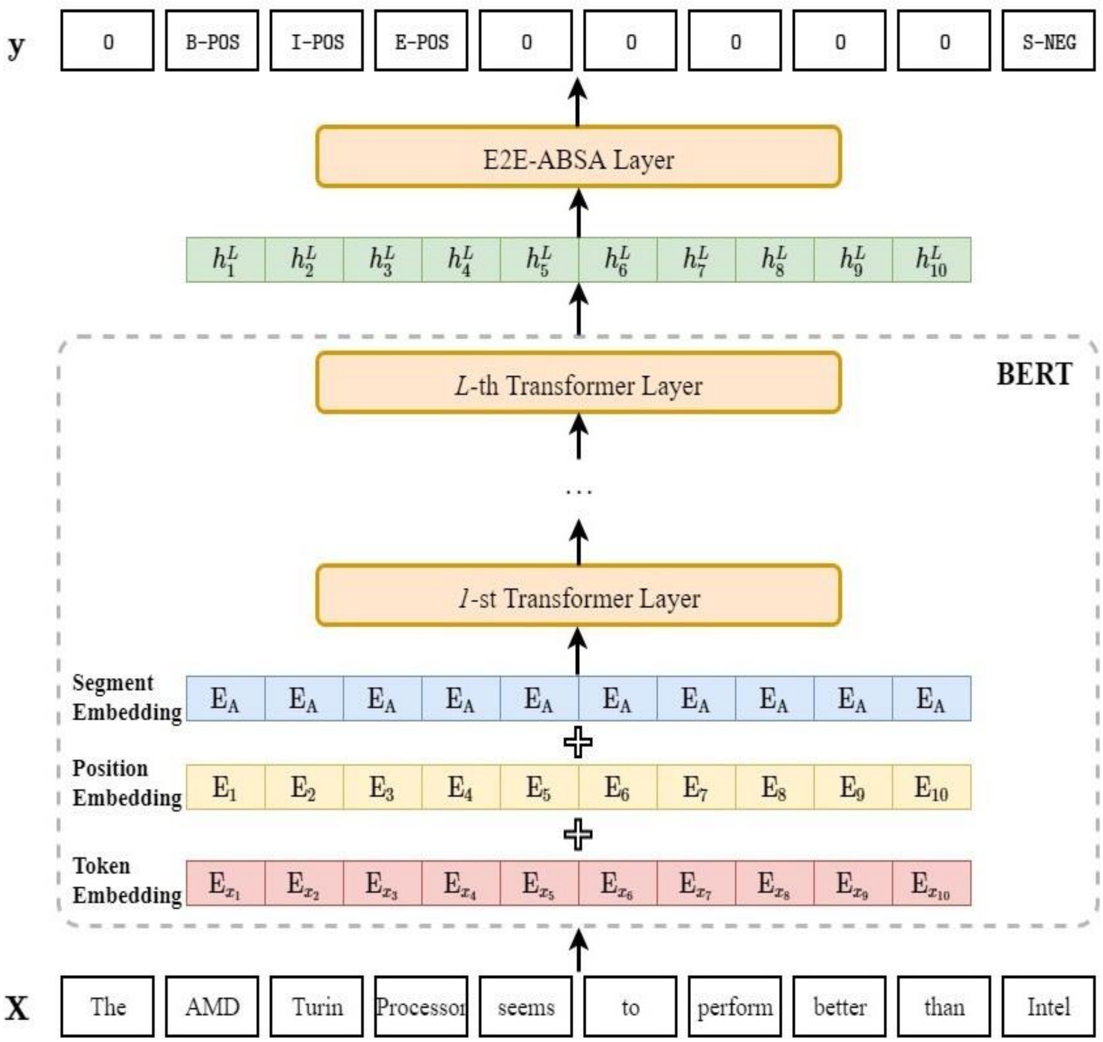

y
0
B-POS
I-POS
E-POS
0
0
0
0
0
S-NEG
E2E-ABSA Layer
h_1^L h_2^L h_3^L h_4^L h_5^L h_6^L h_7^L h_8^L h_9^L h_{10}^L
BERT
L-th Transformer Layer
...
1-st Transformer Layer
Segment Embedding
E_A E_A E_A E_A E_A E_A E_A E_A E_A E_A
Position Embedding
E_1 E_2 E_3 E_4 E_5 E_6 E_7 E_8 E_9 E_{10}
Token Embedding
E_{x_1} E_{x_2} E_{x_3} E_{x_4} E_{x_5} E_{x_6} E_{x_7} E_{x_8} E_{x_9} $E_{x_{10}}$
X
The
AMD
Turin
Processor
seems
to
perform
better
than
Intel

Figure 2.4: BERT E2E for ABSA (X. Li et al, 2019)

2.3.1 Comparison of surveys to perform ABOM and MTL

The Table 2.2 describes the Survey that has been carried out on Aspect based opinion mining (ABOM) with the studies on the same in recent times along with their core objectives.

Table 2.2: Survey on the ABOM

Approaches	Year	Objective
Issues and challenges of Aspect based Analysis: A Comprehensive Survey (Nazir Rao, Wu &Sun, 2020)	2020	The authors' discussed issues and challenges to perform the aspect and opinion term extraction, and aspect term classification. Also, highlight the factors responsible for the sentimental analysis dynamically and process the future research direction by critically analyzing the present solutions.
Deep learning for Aspect=Level Sentiment Classification Survey, Vision & Challenges (J. Zhou et al., 2019)	2019	The authors' discussed about recent deep learning approaches which are effective to perform ABOM tasks and provide the comparisons and summaries for the corresponding algorithms in each standard dataset.
Deep Learning for Aspect Based sentiment Analysis: A comparative Review (Do et al., 2019)	2019	The Authors' explained the deep learning approaches and procedure to perform ABOM tasks in detail with certain neural network and embedding algorithms. The authors claimed that the survey is designed for the students and researcher to identify the deep learning approach mechanism in ABOM.

Similarly, the below Table 2.3 depicts the Surveys on Multi-Task Learning (MTL) on recent studies with their core objectives. The purpose of this survey is to encase what ABOM and MTL can do in conjunction if their consecutive results are such.

Table 2.3: Survey on the MTL

Approaches	Year	Objective
A survey on Multi-Task Learning (Y. Zhang & Yang, 2021)	2021	The authors' discussed the issues and challenges to perform the different types of Multi-Task Learning. Also, they give a survey for MTL from the perspective of algorithmic modeling, applications, and theoretical analysis.

Multi-Task Learning for Dense Prediction Tasks: A Survey (Vanderhende et al.,2021)	2021	The authors examine various optimization methods to tackle the joint learning of multiple tasks and summarize the qualitative elements of these works and explore their commonalities and the differences, Finally, they conduct a thorough experimental assessment across a number of dense prediction benchmarks to assess the advantages and disadvantages of various methodologies, including both architectural and optimization-based methodologies.

2.3.2 Comparisons between Word2Vec and BERT Embedding

The Table 2.4 demonstrates the differences between Word2Vec Embedding and BERT Embedding. The comparison aids to suffice evidence that for the task at hand, BERT is a rather suitable alternative.

Table 2.4: Word2Vec vs BERT embedding

Parameters	Word2Vec embedding	Bert embedding
Context	The Word2Vec is a context independent approach which means it generates vector representation for each word in a sense that if word uses in different context still the representation for the same is same. For example, given sentence, “I love Apple pie but hate Apple MacBook”, the word apple has same representation, whether used in context of food or company in word2vec algorithm (HUILGOL, Aug 2020)	The BERT embedding is a context-dependent approach which means it generates vector representation for each word in a sense that if a word is used in different context, then it’s representation is different. For example, “I love Apple pie but hate Apple MacBook”, the word apple has different representation as it is used in context of food and company in BERT algorithm (Gupta, Nov 2020)
Word Ordering	The word position is not taken into account in word2vec embeddings (HUILGOL, Aug 2020)	Before computing the embedding, the BERT method basically accepts the location(index) of each word in the sentence as input (McCormick, May 2019)
Embedding generation	Word2Vec has pretrained embeddings that may be used right away. The embeddings are given as 1-to-1 mapping between words	However, BERT creates contextual embeddings, the model’s input is a sentence rather than a single word. This is because,

	and vectors (key-value pairs). The model itself isn’t required; all that’s required are the embeddings that the model created. The model takes a single word as input and returns a vector representation of the word as output (HUILGOL, Aug 2020)	before constructing a word vector, the BERT model has to know the context or surro8nding words. To produce embeddings depending on our input and context, we need to have the trained model with us. (Gupta, Nov 2020)
Out-of-Vocabulary (OOV)	In Word2Vec, embeddings are learned at a “word” level. As an example, if your Word2Vec model is trained on a corpus of 1 million unique words, then the model will generate 1-million-word embeddings - one vector per word. Although such representations might work for words encountered within a given vocabulary space, they cannot be used to generate vectors for the words which are OOV. Word2Vec doesn’t support out-of-vocabulary (OOV) words, thus representing a major drawback of the algorithm (Gupta, Nov 2020)	The BERT algorithm, on the other hand, learns representations at the level of “sub words” (also called WordPieces). Subwords can be though of as a sweet spot between character-level embeddings and word-level embeddings. So, even if a BERT model is trained on a corpus containing, say, 1 million unique words, it will only have a vocabulary space of, say, 50k words. Models of this type have become very popular because they can produce vector representations of any arbitrary word and are not limited to the vocabulary space. The vocabulary is essentially endless! BERT supports Out-of-Vocabulary (OOV) words. (Alghanmi, Espinosa Anke &Schockaert, 2020)

Chapter 3
Methods

3.1 Data Mining

A data mining technique uses large data sets to look for patterns, anomalies, and correlations that can predict outcomes. The authors (Chen, Han, & Yu, 1996) identify the Data mining as a knowledge discovery in databases, means a process of non - trivial extraction of implicit, previously unknown, and potentially useful information (such as knowledge rules, constraints, regularities) from data in databases. There are three major approaches to Data mining, including clustering, classification, and association rule mining, which will be discussed in the following sections.

3.1.1 Association Rule Mining

A machine learning technique called association rule learning is used to discover interesting relationships between variables in large databases. An analysis of patterns, correlations, and associations observed from datasets stored in various types of databases, such as relational databases, transactional databases, and other types of repositories, this process known as Association Rule Mining. A rule is defined as an implication of the form: $A \Rightarrow B$ and A, B is Subset from the Item-set (I). One of the techniques of association rule mining is the apriori

(Agrawal & Srikant, 1994) algorithm. In the apriori algorithm, support and confidence are counted by the below equation:

Support(s): The support of an itemset A ⊆ I is the fraction of transactions in (T) that contain both A and B. You can calculate an Itemset's support count by looking at how many transactions in the database contain the item (Rana & Cheah, 2017).

$$\text{Support(itemset)} = \frac{number\ of\ tuples\ in\ the\ itemset}{total\ number\ of\ tuples\ in\ the\ database}$$

The confidence of the rule A→B is the conditional probability that a transaction in T contains B, given that it also contains A(Rana & Cheah, 2017)

$$Confidence(A \to B) = \frac{Support\ (A \cup B)}{Support(A)}$$

Problem: Consider the given transactions, lets say T = T1;T2;T3;T4;T5 given in Table 3.1, some items are bought in all these transactions, where candidate set (C1)=I1, I2, I3, I4 using association rule mining (Apriori algorithm), we can find the set of frequent patterns from large itemset (Li) iteratively by computing the support of each itemset in the candidate set Ci.

Solution:

Step 1: Get frequent item (L1) from candidate set (C1).

The principal step in the apriori process is to find a frequent item by counting the occurrence of each item. The items that don't satisfy the minimum support count are pruned and produced frequent items (L1). In our case, frequent item (L1) = I1:3, I2:5, I3:1, I4:3. In pruning, we delete item set which have less count than the support. Here, I3 has 1 count and minimum support is 2 due to which delete the I3 from the further process.

Table 3.1 Dataset for Apriori

Transaction	List of items
T1	I1, I2
T2	I1, I2, I4
T3	I1, I2
T4	I2, I3, I4
T5	I2, I4

Step 2: Generate candidate set (C2) from the frequent item (L1) by Apriori join (L1 App-join L1).

W can generate a candidate set (C2) by L1 App-join L1. Frequent item (L1) can be joined only with an item that comes after it infrequent item (L1). Which will give candidate set (C2) = I1I2, I1I4, I2I4.

Step 3: Get frequent item (L2) from candidate set (C2)

Frequent item (L2) is obtained by the same procedure as in step 1. We can count the occurrence of each item in candidate set (C2), and infrequent items are removed to create frequent itemset (L2) = I1I2: 3, I1I4: 1, I2I4: 3.

Step 4: Repeat above steps until we reached the final prune frequent Item set.

3.1.2 Clustering

Clustering is a process of the arranging the items in groups or clusters, so they have similar characteristics and are more similar to one another. Clustering is an unsupervised data mining strategy that does not require manual labeling or any kind of target values. K-means (Forgy, 1965), DBSCAN (Ester, Kriegel, Sander, Xu, et al., 1996), Agglomerative and many other algorithms are called as the clustering algorithms. Data points are assigned to non-overlapping clusters subgroups) according to the k-means algorithm with each having a unique cluster and each unique cluster have data points which share some properties with other data points in the same cluster, but they do not share any of their characteristic with other data points which are in different clusters (Korovkinas, Danenas, & Garsva, 2019). It aims that data points in the same clusters have similar characteristics and share similar properties while the distance between clusters is kept far as possible. It distributes data points to clusters in such a way that the sum of the squared distances between them and the cluster's centroid (arithmetic mean of all the data points in that cluster) is as small as possible (Korovkinas et al., 2019).

Problem Identified: Consider the data points present in n dimensional space with the coordinates of x-axis and y-axis as a dataset that can be plots on a particular graph. Using the K-means algorithm for clustering, we aim to find the possible clusters.

Solution: The K-means clustering algorithm consists of five major steps:

Step 1: Firstly, we assume the two data points as the separate centroids and based on that, we need to calculate the distance between the initial centroid points with other data points to identify the similarities. The formula is given below.

$$Distance = \sqrt{(X1 - X2)^2 + (Y1 - Y2)^2}$$

where X_1; Y_1 denotes the centroid of the cluster and X_2; Y_2 represents the data points to calculate the distance against the centroids.

Step 2: Next, we need to group the data points which are closer to centroids. The closer data points are considered as they share some properties with each other.

Step 3: We calculate the mean values of the clusters created, and the new centroid values will these mean values and centroid move along with the graph.

Step 4: Again, the values of Euclidean distance are calculated from the new centroids, and the above steps will be repeated until the max iteration condition meets.

3.1.3 Classification

We observe which category the group of input or observation belongs to in classification. For instance, the remark "The pizza is wonderful" can be divided into three categories: positive, negative, and neutral. Several algorithms are used to perform the classification tasks such as Support Vector Machine (SVM) (Cortes & Vapnik, 1995), Decision Tree (Maimon & Rokach, 2014), Naive Byes, and Neural Networks and algorithm which performs classification tasks are called as a classifier. I explained the Naive Bayes technique and how it works to perform the classification to help you understand it.

The Naive Bayes algorithm was based on a basic assumption: the existence of one characteristic in a class is unrelated to the presence of any other feature. For example, if a fruit is red, round, and roughly 3 inches in diameter, it is termed an apple (Parkhe & Biswas, 2014). Even if these characteristics are reliant on one another or on the presence of other characteristics, they all add to the likelihood that this fruit is an apple, which is why it is called a “Naïve”. The formula to compute the probability in Naive Bayes algorithm is given below:

$$P(c|x) = \frac{P(x|c) * P(c)}{P(x)}$$

Where, P (c | x) denotes the posterior probability of class (c, target) given predictor (x, attributes). P(x) represents the prior probability of predictor, and P(c) is the prior probability of class. P (x | c) is the likelihood which is the probability of predictor given class.

Problem Identified: Consider the Table 3.2 (Ray, September 2017), our aim is to identify the chances in which weather condition players will play.

Let’s understand the above formula and computation of algorithm by example, below I have a training data set of weather, and corresponding target variable ‘Play’ (suggesting possibilities of playing) (Ray, September 2017). Now, we need to classify whether players will play or not based on weather condition. Let’s follow the below steps to perform it.

Step 1: Convert the data into a frequency table by counting the play values associated with each weather class (e.g., Overcast, Rainy, and Sunny) (e.g., Yes or No) (Ray, September 2017).The Table 3.3 depicts the same.

Step 2: Create a likelihood table by calculating probabilities such as overcast probability = 0.29 and playing probability = 0.64 (Ray, September 2017). The Table 3.4 demonstrates the same.

Table 3.2: Dataset for Naïve Bayes

Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	Yes
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

Table 3.3 Frequency Table

Weather	No	Yes
Sunny	2	3
Overcast		4
Rainy	3	2
Grand Total	5	9

Table 3.4: Likelihood Table

Weather	No	Yes	
Sunny	2	3	5/14 = 0.36
Overcast	0	4	4/14 = 0.29
Rainy	3	2	5/14 = 0.36
Total	5	9	
	5/14 = 0.36	9/14 = 0.64	

Step 3: Calculate the posterior probability for each class using a Naive Bayesian equation. The outcome of prediction is the class with the highest posterior probability (Ray, September 2017).

Problem: Players will play if the weather is sunny. Is this statement correct?

We can solve it using above discussed method of the posterior probability.

$$P(Yes\ |Sunny) = \frac{P(Sunny \mid Yes) * P(Yes)}{P(Sunny)}$$

Here we have P(Sunny | Yes) = 3/9 = 0.33, P(Sunny) = 5/14 = 0.36, P(Yes) = 9/14 = 0.64

Now, P(Yes | Sunny) = 0.33*0.64/0.36 = 0.60, which has higher probability. The Naïve Bayes technique predicts the likelihood of distinct classes on multiple attributes in a similar way. This approach is most commonly used for text classification and problems with numerous classes (Ray, September 2017)

3.2 Neural Networks

We’ll talk about Neural Networks and how they train and test on data in this part. We’ve also gone over the techniques and how the computing works in neural network training. All neural network-based approaches start with a huge dataset to train their model. Specific hyperparameters, such as the Number of Epochs, must be initialised before the neural networks training starts (e.g., how many times the model will run over the dataset to identify patterns). The weights among neurons are changed throughout network training, and after reaching the maximum number of epochs, the training is stopped, and the weights between neurons are finalised. At testing time, these finalised weights are employed as classification rules. There are different type of neural networks such as FFNN (LeCun, Bengio, & Hinton, 2015), LSTM (Hochreiter & Schmidhuber, 1997), GRU(Cho et al., 2014), CNN(Sercu, Puhrsch, Kingsbury, & LeCun, 2016), Attention(Vaswani et al., 2017), Memory network(Weston, Chopra, & Bordes, 2014), Neural Turing Machine(Graves, Wayne, & Danihelka, 2014), Transformers (Clark, Khandelwal, Levy, & Manning, 2019) and many others.

3.2.1 Feed-forward Neural Network (FNN)

Each input is connected to the neurons of the very first layer of the network in FFNN, while the output is generated by the network’s last layer. Hidden layers are the layers between the first (Input) and the last (Output). Figure 3.1 demonstrates the architecture of the Feed-Forward Neural Network. This type of Neural Networks have each neuron inside one layer connected to every neuron in the next layer by weights, allowing all input to be processed forward due to which this network is called as FFNN. To understand, Table 3.5 shows an example of Input for the Feed Forward Neural Network.

1. Convert the sentence (S) into list of the tokens (Food, is, great) and generate each token related embedding vector (I_{food}, I_{is}, I_{great}) and each vector is given as input to the feed forward input layer.

Table 3.5 Input for the FFNN

User reviews	Opinion Polarity
Food is great	Positive (1)
Service is bad	Negative (0)
Pizza and pasta were delicious	Positive (1)

2. Each embedding vector connected to the hidden neuron (mathematical function) via weights (arbitrarily selected).
3. Each neuron in input layer will be denoted as (I_{food}, I_{is}, I_{great}). Similarly, in hidden and output layer, neurons are represented as (H_1, H_2, O_1, O_2).
4. Each neuron performs below calculation with each connected (w_{n_j})input vector (I_n).

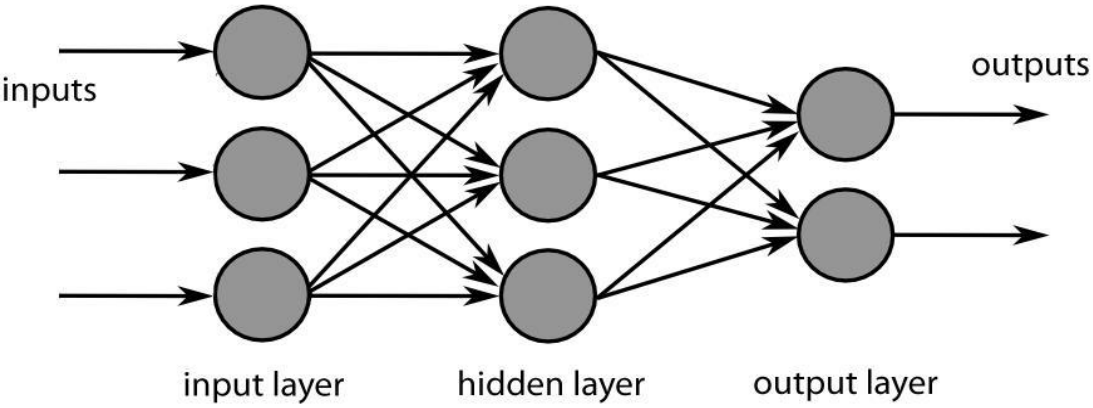


Figure 3.1: Feed forward Neural Network

$$y_j = \sum w_{n_j} \cdot I_n + b_j$$

$$Out = sigmoid\left(y_j\right) = \frac{1}{1+\exp^{\left(-y_j\right)}}$$

5. The preceding example's target value is associated with class O1, but the model predicts class O2 as the output because it reaches a greater value than O1. This indicates that the network misclassified the example; the network will then change the weights using the backpropagation technique and a learning rate eta = 0.1.
6. If the target categorisation was associated with O1, this means that the target output for O_1 was 1, and the target output for O_2 was 0. Hence, we can calculate the error values for the output units O_1 and O_2 as follows:

$$\delta o1 = O1(E)[1 - O1(E)][T1(E) - O1(E)]$$

7. To calculate the error values for the hidden units H1, multiply the error term of O1 by the weight from H1 to O1, then add this to the multiplication of the error term of O2 and the weight between H1 and O2. To turn this into the error value for H1, we calculate by below formula: (H1(E)*(1-H1)),
8. $\Delta = \eta * \delta H * In$, which is used to calculate the difference that the network wants to make in the weights. As we can see in Table 1.6, the calculation of error for each input neuron to hidden neuron will be given. All the values are arbitrary selected.
9. $New_{weights}$ = $Old_{weights}$ + Δ, using the $New_{weights}$ the above process is repeated till the maximum number of epoch limit reaches.

10. At the end, the finalized weights are used as classification rule to perform the classification. Table 3.6 shows the Summary of what the output shall look like (with arbitrary values), considered the input as in Table 3.5.

Table 3.6 FFNN Summary

Input	Hidden unit	η	δH	In	$\Delta - \eta * \delta H * In$	Old weight	New weight
I1	H1	0.1	-0.0000705	10	-0.0000705	0.2	0.1999295
I1	H2	0.1	-0.00259	10	-0.00259	0.7	0.69741
I2	H1	0.1	-0.0000705	30	-0.0002115	-0.1	-0.1002115
I2	H2	0.1	-0.00259	30	-0.00777	-1.2	-1.20777
I3	H1	0.1	-0.0000705	20	-0.000141	0.4	039999
I3	H2	0.1	-0.00259	20	-0.00518	1.2	1.19482

3.2.2 Long-Short Term Memory

The LSTM structure is depicted in Figure 3.2, which includes the input and output of each LSTM cell, as well as the hidden operations executed by each LSTM cell. Input Gate, Forget Gate, and Output Gate are the three logical operations performed by each LSTM cell. Each cell in an LSTM is connected to the next cell and shares its output, which is used as input in the following cell. Basically, the structure of the LSTM network is a chain where each cell output is connected to the next cell input.

According to the Figure 3.2, each LSTM cell accepts three inputs: current timestamp (x_t), prior cell output(h_{t-1}), and previous cell memory (C_{t-1}). To begin, all three input processes must pass through the Forget Gate. The output from the previous cell (h_{t-1}), and the current timestamp input (x_t), will be concatenated and supplied as inputs in the forget gate. Concatenated input will be multiplied by weight and processed through the sigmoid (σ) activation function at the Forget Gate (f_t). If the forget gate value is set to 0, there is no need to process any additional data; otherwise, the forget gate value is set to 1, indicating that the data is important to process.

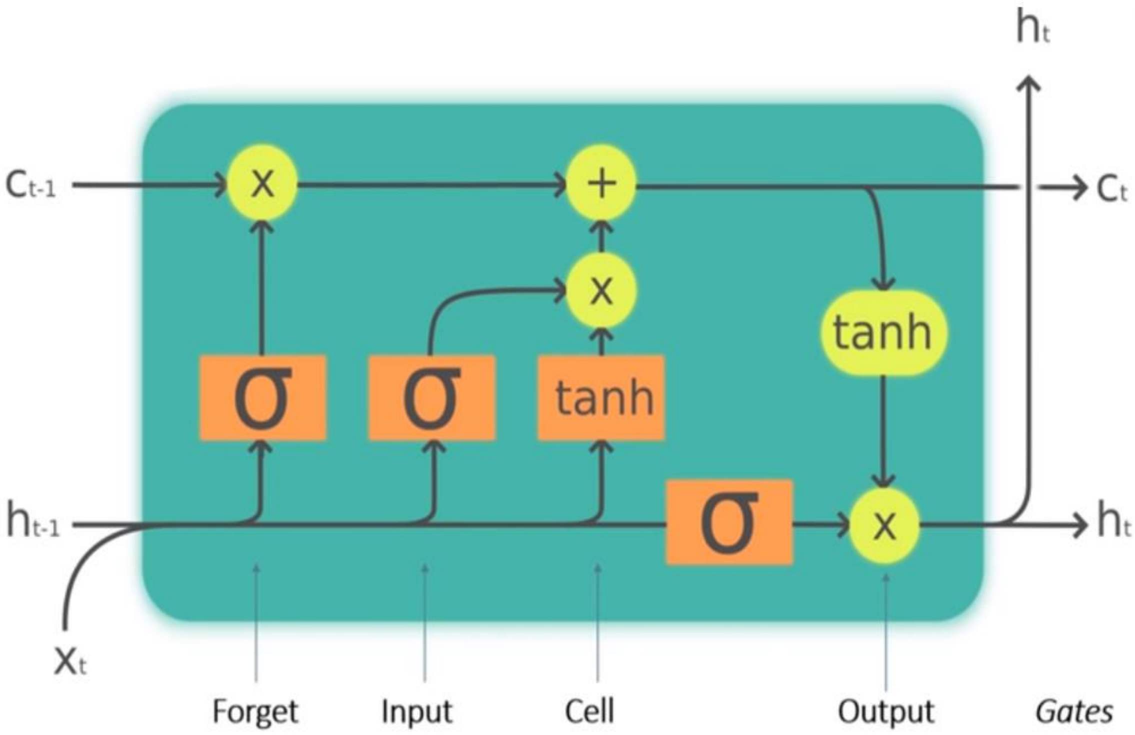


Figure 3.2 LSTM (Yuan, Li and Wang, 2019)

$$f_t = \sigma(\omega_f[h_{t-1}, x_t] + b_f)$$

The concatenated input ($[h_{t-1}, x_t]$) will be supplied to the tanh and sigmoid activation functions to process, and the output from both will execute element-wise multiplication (·) to generate the new memory in the input gate (i_t). The sum of the input gate's output and the forget gate's output

(f_t) will be used as a new memory, and the sum with the forget gate indicates how much old memory should be reflected on new memory (c_t).

$$i_t = \sigma(\omega_i[h_{t-1}, x_t] + b_i)$$

$$c_{t_1} = \tanh(\omega_{\mathrm{C}}[h_{t-1}, x_t] + b_c)$$

$$C_t = f_t \cdot c_{t-1} + i_t \cdot c_{t^1}$$

New memory and concatenated input will be handled in the output gate (O_t) using the tanh and sigmoid activation functions, respectively. Finally, the output from both the activation functions is multiplied element by element to produce the final result.

$$Ot = \sigma(\omega_0[h_{t-1}x_t] + b_0)$$

$$h_t = O_t \cdot \tanh(M_t)$$

Here, w_i, w_f, w_c, w_o are the weights and b_i , b_f, b_c, b_o are the bias and both are learnable parameters of the network.

3.2.3 Gated Recurrent Unit

The input was processed by the LSTM Network through three gates in a time-consuming and tricky process. The authors of (Cho et al., 2014) built the Gated Recurrent Unit, a more effective and less complicated network (GRU). The data, on the other hand, is passed through two gates in GRU: the Reset Gate and the Update Gate. As shown in Figure 3.3 (Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola, June 2021), the Reset gate functioned similarly to LSTM forget gates in terms of determining how much information needs to be processed, whereas the Update gate functioned similarly to LSTM input and output gates in terms of determining how much previous information needs to be considered in order to generate new information and output. The computation formula for Reset and Update Gate in GRU is given below: The output from the previous GRU cell (h_{t-1}) and the current timestamp input (x_t) will be concatenated and supplied to the gate as inputs to reset and update it in the GRU network. The concatenated input will be processes during the sigmoid (σ) activation at both gates, as shown in Figure 3.3

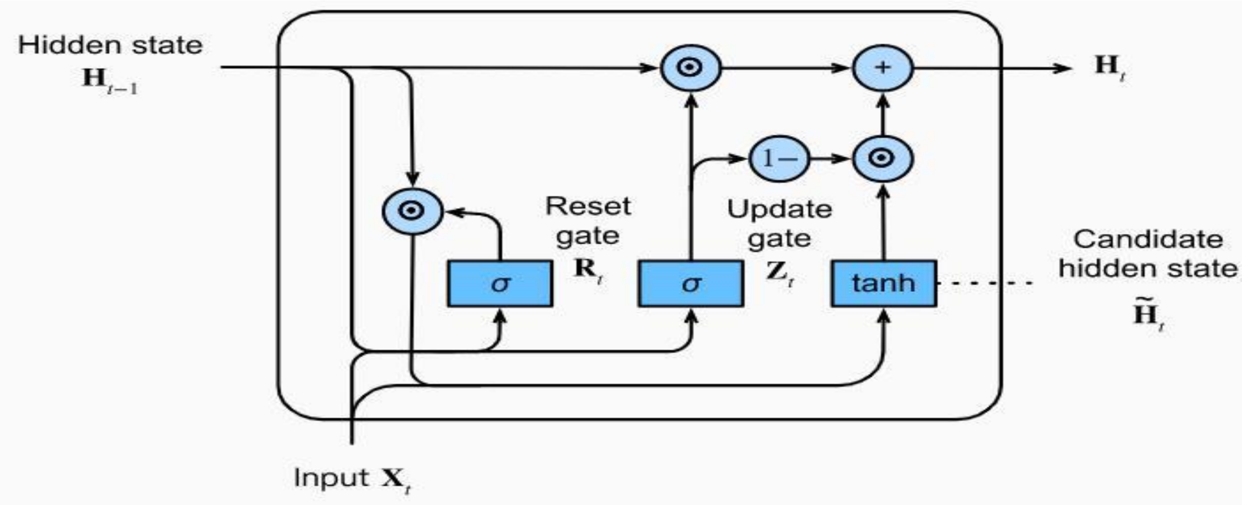


Figure 3.3: GRU (Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola, June 2021)

$$r_t = \sigma\left(\omega r\left[h_{t-1,} Final_t\right] + b_r\right)$$

$$z_t = \sigma\, th[\, h_{t-1}, Final_t] + b_z$$

It will do element-wise multiplication (·) with reset gate output and previous cell output after establishing the reset and update gate to determine how much information is required. The data will then be passed via the tanh activation function, which will conduct element-wise multiplication with the update gate’s output.

$$n_t = \tanh(\omega_n[h_{t-1} \cdot r,\ Final_t] + b_n$$

$$h_t = h_{t-1} \cdot z_t + (1 - z_t) \cdot n_t$$

Here, rt and zt denotes the reset and update gate, respectively. Here w_r, w_z, $w_n$are the weights and b_r, b_z and b_n are the bias and both learnable parameters of the network.

3.3 Proposed Approach

It is very essential to detect user's sentiment towards the product to identify which features are more liked by the user and how can we enhance the performance of any product or service which are provide by the companies. Also, some companies such as DELL, Rogers, and many other Telecommunications company uses the surveys and feedback forms to identify how satisfying their services are. To determine it the Deep Learning models can be used. There are several approaches such as Attention network, Memory network, LSTM and GRU network, and pre-trained model-based approaches are available to perform the ABSA tasks. The previous BERT-based approaches require different models to perform all the four sub-tasks of ABOM from user reviews. To resolve that problem, we suggested a novel BERT-based model (BERT-ABSA) for Aspect based Sentiment Analysis, in which, we build BERT-based models. The model performs the BERT-based Multi-Task Learning approach to extract the aspect terms, opinion terms, and opinion polarities from user reviews. Simply put, multi-task learning aims to preform various task learning to become familiar with numerous errands simultaneously, while amplifying the performance on one or all undertakings in general. The use of Multi-Task Learning showcases the advantage on deep learning models as well as by learning several tasks together we can reduce the train time and computation.

3.4 BERT for ATE, OTE and ABSA

In this section, we have introduced our approach related methods such as Multi-task Learning, Sequence Labeling, and BERT model.

3.4.1 Bidirectional Encoder Representations from Transformers (BERT)

BERT stands for Bidirectional Encoder Representations from Transformers (Devlin, Chang, Lee, & Toutanova, 2018). We’re utilising the pre-trained BERT-BASE model, which has 12 encoder layers and includes Attention, Layer Norm, and Feed Forward Neural Network in each encoder layer (FFNN). One Encoder layer is given in the Figure 3.4 (Alammar, 2018). BERT-BASE model contains a total of 12 like that on top of each other. BERT is a pre-trained model, unlike other neural network-based techniques, and can be re-tuned with just one additional output layer due to which it can give a head start to our approach. Pre-training works on the principle of training the model with diverse datasets and using their weights as an initial weight in the model. The tokenize function is also used by BERT to produce tokens and token-related ids from the input sentences. The BERT Encoder function takes these token-related ids as input and creates each token-related 768-dimension vector (Sun, Huang, & Qiu, 2019b).

Problem Identified: Consider the Table 3.7 as input to the BERT which have the user reviews and aim is to encode the information and understanding the operation.

Table 3.7: Input to the BERT

User Reviews	Opinion Polarity
The pizza is great	Positive (1)
Service is bad	Negative (0)
Restaurant is nicely decorated	Positive (1)

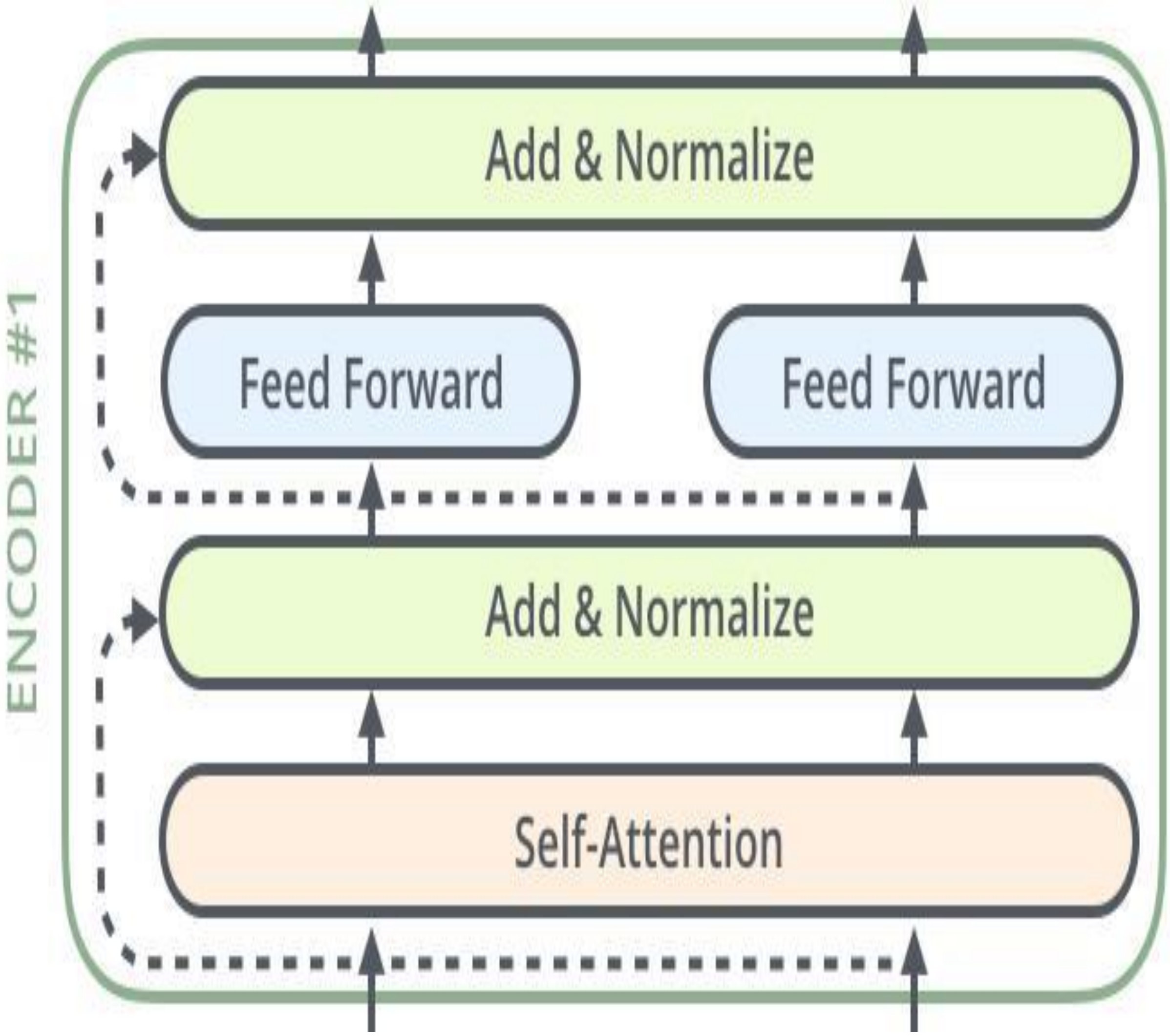


Figure 3.4 : BERT Encoder Layer (Alammar, 2018)

BERT Tokenize function:

1. For each sentence (s) in a dataset Do:
2. Add two unique tokens in each sentence, such as [CLS] and [SEP]. The token [CLS] is used to indicate the starting of sequences and classification, while [SEP] is used to separate the sequence from the subsequent. The BERT tokenizer also requires the max_len parameter, which is used to maintain the same sequence length for all the sentences in the dataset. For example, in our case max_len is 7 (Z. Li et al., 2019).
3. To generate Tokens, Input Id, Attention Mask, token type ids from a sentence, BERT Tokenizer function convert each sentence into list tokens.
4. Input Id are token indices (numerical representations of tokens) and Attention Mask is used to identify the tokens and padding where tokens are represented as 1 and padding denotes 0.
5. Token_type_ids are used to detect the sequence of the sentence. The Input Id, Attention Mask, token type ids for sentence S1 are denoted in below Table 3.8. In Table 3.8, zero in last two columns denote the padding.

Table 3.8: BERT sentence Representation

Tokenized sentence for sentence S1	CLS	Food	Is	Great	[SEP]	0	0
Input Id for sentence S1	101	2833	2003	2307	102	0	0
Attention Mask for sentence S1	1	1	1	1	1	0	0
Token type ids for sentence S1	0	0	0	0	0	0	0

Embedding vector(x) for sentence S1	x_{CLS}	x_{Food}	x_{is}	x_{great}	x_{SEP}	0	0
Query vector (Q) for sentence S1	$Q_{CLS(8)}$	$Q_{Food(8)}$	$Q_{is(8)}$	$Q_{great(8)}$	$Q_{SEP(8)}$	0	0
Key vector (K) for sentence S1	$K_{CLS(8)}$	$K_{Food(8)}$	$K_{is(8)}$	$K_{great(8)}$	$K_{SEP(8)}$	0	0
Value vector (V) for sentence S1	V_{CLS}	V_{Food}	V_{is}	V_{great}	V_{SEP}	0	0

BERT Encoder function

Input: The tokenize text and Input_id, Attention_Mask, and Token_Type _id are given as input to the Encoder function.

Output: Generate each token related encoded 768-dimension vector.

1. In encoder layer, every token embedding vector of 768-dimensions creates a 12 different triplet of 64-dimension vectors, called the key (K), query (Q), and value (V) vectors. As BERT uses Multi-head Attention and number of heads are 12 due to which from each token embedding vector, 12 different triplets of query, Key, and value vectors are generated.
2. In attention layer, every token related attention score will be calculated by below equation:

$$Attention\ Score(Z) = Softmax\left(\frac{Q \cdot k^T}{\sqrt{d_k}}\right)V$$

Where Q, K and V are respectively query, key and value vector and d_k denotes the dimension of the key vector. The softmax function is an activation function whose equation is given below

$$Softmax(Ai) = \frac{\exp(Ai)}{\Sigma \exp(Ai)}$$

3. Above step is repeated for all the 12 attention heads and output of 12 attention heads concatenate to generate the final output of Attention Layer which will be 768-dimensions vector.
4. After that, the final output from attention layer (Z) will be given to Add and Norm layer where embedding vector (x) are added into the attention output (Z) and given to LayerNorm function (Ba, Kiros, & Hinton, 2016).

$$Normalizedz = LayerNorm(x + Z)$$

5. In the FFNN, output (Normalized$_Z$) from the LayerNorm function is given as input and output of the feed forward neural network will be calculated using below equation

$$y = W.NormalizedZ + b$$

$$Out = sigmoid(y) = \frac{1}{1+\exp^{(-y)}}$$

Where W and b denotes weight and bias, respectively.

6. The output of previous Norm Layer and feed forward neural network layer will be concatenate and given as input to the Add and Norm layer and final encoder output will be generated.

$$Final_{output} = LayerNorm(\ Normalized_z + Out)$$

Above steps are repeated up to 12 time as BERT-BASE model contains 12 Encoder layer and generate 768-dimension vector as output for each token in a sentence.

3.4.2 Multi-Task Learning (MTL)

Multi-task learning has proven to be effective in a variety of machine learning applications, ranging from natural language processing and speech recognition to computer vision and regenerative medicine. MTL has been referred to by a variety of titles, including joint learning, learning to learn, and learning with auxiliary task. In general, if you find yourself optimising more than one loss function, you’re engaging in multi-task learning (in contrast to single-task learning). In certain situations, it can be beneficial to think about what you’re trying to do in terms of MTL and take conclusions from it.

In the context of Deep Learning, multi-task learning is typically done with either hard or soft parameter sharing of hidden layers. So, hard parameter sharing (*Figure 3.6*) is done by sharing the hidden layers across all jobs while keeping a few task-specific output layers separate. The hard parameter sharing reduce the chances of the overfitting. In soft parameter sharing (*Figure 3.5),* each task has its own model with its own parameters. The distance between the parameters of the model is then regularized in order to encourage the parameters to be similar. The example model for soft parameter and hard parameter sharing has been displayed in Figures 3.5 and 3.6 respectively. In our approach we have applied the Hard Parameter sharing in which BERT model is shared by the three tasks and at the end each task specific layer has been attached.

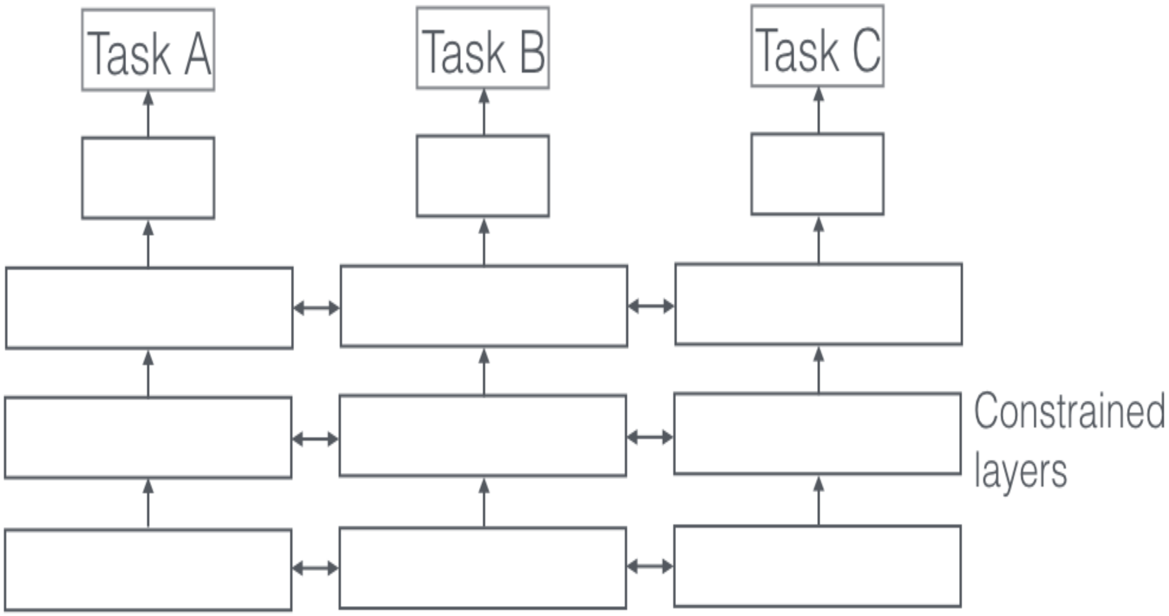


Figure 3.5: Soft Parameter Sharing

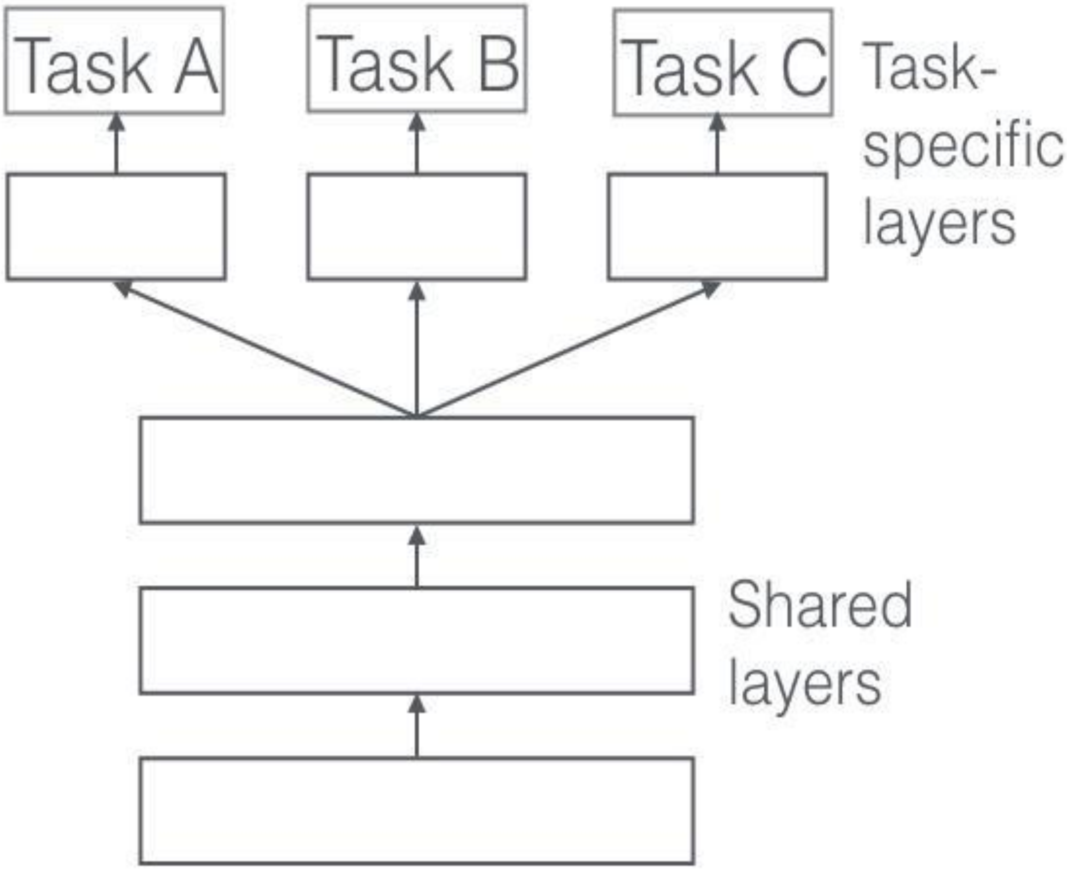


Figure 3.6: Hard parameter Sharing

To extract all the Aspect Terms and Opinion Terms and ABSA present in the user reviews, we will use Multi-Task Learning (MTL) approach using BERT (Devlin et al., 2018). In MTL, a variety of tasks are learnt together in a single network, each task having its own output (Caruana, 1997). MTL captures the similarities between tasks and improves model generalization capacity in certain situations by learning semantically similar tasks in parallel using a shared representation (He et al., 2019). The MTL network has a common input, and layers of the BERT (Devlin et al.,2018) model are shared between the two tasks, such as Aspect Term Extraction (ATE), Opinion Term Extraction (OTE) and Aspect-Based Sentiment Analysis.

According to the previous research, the use of MTL reduces the chances of overfitting the model, which is a bigger problem in most neural network approaches (Ruder, 2017). Overfitting means the model learns the noise in data as a concept of training data due to which it affects the accuracy. Our proposed model's approach is shown in algorithm 1; By giving user review as an input to the model, it can predict the Aspect term, Opinion term, and ABSA simultaneously by applying Multi-Task Learning (MTL). In the BERT-MTL approach, we calculate the final loss function as sum of both the task-related loss function such as $Loss = Loss_{OTE} + Loss_{ATE} + Loss_{ABSA}$, where all three losses are calculated by CrossEntropyLoss() (Xue, Zhou, Li, & Wang, 2017) function.

3.4.3 Sequence Labelling Approach

In Sequence Labeling, each word in the sentence has a label in the BIO format. We displayed words in a sentence with its predicted label by BERT-MTL, where B in BIO stands for the Beginning of aspect terms, I stands for Inside (continue) of aspect terms, and O stands for Outside of aspect terms. Some of the aspect terms are phrase level (two or more words for e.g., Cheese Garlic Bread), due to which we need the Inside label to identify all the words that can be considered as aspect terms. For the given sentence \Garlic Bread is good.", the output of sequence labeling task for each word of input will be Garlic = B, Bread = I, is = O, and good = O. The input sentence with n words which will be given to the BERT model is represented as:

$$Input1 = [CLS], w1, w2, w3, \dots, wn, [SEP];$$

$$Final = BERT(Input1);$$

$$Y = FFNN\ (W1 \cdot Final + b1);$$

where w1, w2, w3, ……, wn are words present in a sentence and input will be prepared by adding unique tokens with words. The weight W1ϵ R^{3*d1} and bias b1ϵR^{3} (3 is a total number of classes (BIO) and d1 is a hidden dimension of BERT). In the end, we have used the FFNN layer to perform classification for each word based on class (BIO) probability.

3.5 Proposed Algorithm

Algorithm 1: Proposed approach for ATE, OTE and ABSA

```
Input: Training sentence (s) from dataset
Output: Aspect Ter, Opinion term and Aspect Term related polarities
Loop until the terminal condition is met. Maximum Training Epochs:
    $Sentences_{batch} \longleftarrow sample(Sentences; b);$            //sample a minibatch of size b
    $Input_{Id}, Attention_{Mask}, token_{type_{ids}} = BERT\ Tokenizer(Sentences);$
    $Final = BERT\ Encoder\left(Input_{Id}, Attention_{Mask}, token_{type_{ids}}\right);$
    $prediction1\ (ATE) = Feed\ Forward\ NN(Final);$                    //Classification layer 1
    $prediction2\ (OTE) = Feed\ Forward\ NN(Final);$                    //Classification layer 2
    $prediction3\ (ABSA) = Feed\ Forward\ NN(Final);$                   //Classification layer 3
    $Loss1 = CrossEntropyLoss\ (prediction1, target1);$                       //Loss for ATE
    $Loss2 = CrossEntropyLoss\ (prediction2, target2);$                       //Loss for OTE
    $Loss3 = CrossEntropyLoss\ (prediction3, target3);$                       //Loss for ABSA
    $Loss = Loss1 + Loss2 + Loss3;$                                           // Sum for the Loss
    $Backpropogation\ algorithm\ is\ used\ to\ change\ the\ weights\ of\ the\ approach$
End
```

3.6 Steps of Proposed Approach

1. Dataset contains sentences or reviews, and each sentence has target. For example, “The pizza is good.” and the target value for a sentence is “pizza”, “good”, and “positive” to determine the aspect term, opinion term, and related opinion polarity, respectively, from the sentences.
2. First, the sentences are given to the BERT Tokenize Function to generate the tokens, Input_Id, Attention_Mask, token_type ids.

   a) The two unique tokens are added to identify the starting and ending of the sentence, such as [CLS] and [SEP]. Each word will be represented as a token from the input sentence using the WordPiece algorithm. If the input sentence has fewer tokens than the max length, then the [PAD] token will be added up to the max length reaches. For example, from the above sentence, tokens are generated as below, and the max length will be 10,

      Tokens = [‘[CLS]’, ‘the’, ‘pizza’, ‘is’, ‘good’, ‘.’, ‘[SEP]’, ‘[PAD]’, ‘[PAD]’, ‘[PAD]’]

   b) The Input ID is unique number associated for each token or we can assume as index number in list of words, where tokens ‘[CLS]’, ‘[SEP]’, and ‘[PAD]’ is represented by 101, 102, and 0 every time during tokenization.

      Input_Id = [101, 1996, 10,733, 2003, 2204, 1012, 102, 0, 0, 0]

   c) The Attention Mask is used to identify which are the tokens and which are the padding in the input. where tokens from the sentence will be represented as 1 while padding tokens are represented as 0.

      Attention_Mask = [1, 1, 1, 1, 1, 1, 1, 0, 0, 0]

d) The token_type_ids are used to represent the subsequent sentences if they are present in input but here, we do not have any sub-sequence then all the values will be zero. token_type_ids = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0] (Patel, 2021).

3. The above generated Input_id, attention_mask, and token_type ids will be given in BERT Embedding Function to convert each token into respective vectors called as embedding vector which represents the word into higher dimension vector as shown in below Table 3.9

Table 3.9: Embedding representations for each token

Tokens	Embedding Vector
[CLS]	[0.25, -0.78, …,0.93]
The	[-0.57, 0.45, …,0.67]
Pizza	[0.67,0. 23, …, 0.79]
Is	[-0.17,0.57, …, 0.09]
	[-0.15, 0.27, …, 0.33]
Good	[-0.4, 0.19, …,0.21]
[SEP]	[-0.17, 0.07, …, 0.39]

4. The embedding vector related to each token fed to the BERT Encoder Function which will generate a 768-dimension vector for each token of the sentence.

(a) Firstly, each token related embedding vector divide into Query (Q), Key (K), and Value (V) vector. For example, each token processed through the below equation:

$Q \cdot k^T = Q_{Pizza} * k_{CLS},\ Q_{Pizza} * k_{the},\ Q_{Pizza} * k_{pizza},\ ,\ Q_{Pizza} * k_{is},\ ,\ Q_{Pizza} * k_{good},\ ,\ Q_{Pizza} * k_{SEP}$

$Softmax\left(\frac{Q \cdot k^T}{\sqrt{d_k}}\right)$ = 0.006, 0.1007, 0.2592, 0.03507, 0.7045, 0.0012, 0.0006

Where each numeric value denotes each token related importance to token pizza.

(b)The above numeric value will be multiplied with the Value vector (V) to generate the attention layer output (Z_{pizza}).

Zpizza =0.0006 * V_{CLS} +0.1 *V_{the} + 0.26 * V_{pizza} + 0.035 * V_{is} + 0.70 *V_{good} + 0.001 *V. + 0.0006 * V_{SEP}

(c)The Output (Z_{pizza}) will be processed through FFNN and Normalization layer and generate 768-dimension vector for each token. This process will be repeated for each token present in the input (Patel, 2021).

5. The output of the BERT model is given to the three different FFNN to perform different tasks, such as sequence labeling to extract aspect term, opinion term, and related opinion polarity, where it converts the vectors into each class probabilities. In sequence labeling, every word has a label; for example, the token "pizza" has the actual target of [1,0,0] where 1 at the first position denotes the beginning of the aspect term, and the predicted value the model for the word "pizza" is [0.6, 0.1, 0.2].
6. Similar method is applied for opinion term extraction and aspect term related opinion detection.
7. The class probabilities generated by each FFNN and their respective actual target from the dataset will be given to the loss function where a loss will be calculated.
8. For sequence labeling task Cross-Entropy Loss function is used. The Cross-Entropy Loss can be calculated by the equation below:

$$Loss = -\sum t_i \cdot log(p_i)$$

Where t_i denotes target value and p_i denotes the predicted value, for example, target value for word pizza is [1; 0; 0] and predicted value for that sentence [0:6; 0:1; 0:2]. These two arrays will be given as input to the cross-entropy loss and calculation of loss:

$$Loss_{ATE} = - (1* \log(0.6) + 0*\log(0.1) + 0*\log(0.2)) = 0.22185$$

9. After the calculation of loss for three different task using sequence labeling, all the loss will be summed to generate the final loss function.

$$\text{Loss} = Loss_{ATE} + Loss_{OTE} + Loss_{ABSA}$$

10. After that, to reduce the loss of all three tasks, the backpropagation algorithm was used to change the weights of the approach.

3.6.1 Comparison Between different approaches

In the end, we have performed BERT based Multitask learning model which extracts the aspect and opinion terms with each aspect term related sentiment polarity. To compare and validate our contribution, we have performed one comparison based on previous state of the art models which indicate our contribution.

Table 3.10: Comparison between models

Approaches	BERT	MTL	ATE	OTE	ABSA
MGAN	NO	NO	NO	NO	YES
BERT-LSTM	YES	NO	NO	NO	YES
BERT-ATTENTION	YES	NO	NO	NO	YES
BERT -PT	YES	NO	YES	NO	YES
BAT	YES	NO	YES	NO	YES
SEML	NO	YES	YES	NO	YES
IMN	NO	YES	YES	YES	YES
PROPSOSED APPROACH	YES	YES	YES	YES	YES

As we compare our approach with other state of the art models in terms of Model and different learning techniques used by the approaches, and various Tasks performed by the approaches. As we can see in above Table 3.10, The MGAN approach did not used BERT or MTL approach and just performs the aspect-based sentiment analysis tasks while our proposed approach performs MTL and used BERT pre-trained model to perform several ABSA tasks. The BERT-LSTM and BERT-Attention approaches perform BERT based aspect-based sentiment analysis task using different pooling strategy while on the other hand, we have used the BERT with Multitask learning to perform ATE, OTE, and ABSA tasks. Similarly, BAT and BERT-PT models uses the different training approaches such as Adversarial Training using BERT model and Post Training using BERT respectively to perform the ATE and ABSA tasks. However, our approach uses Multitask learning and BERT model to preform ATE, OTE and ABSA tasks. Furthermore, the SEML and IMN approaches uses different type of multi-task learning compared to our, for example, IMN using hard parameter sharing in Multitask Learning with message passing channel while in our approach, we have used the BERT model with Multitask learning on pure hard parameter sharing without any changes still we achieve better results. To conclude, we have used the different approach compared to previous models and achieves better or comparable results on all the three ABSA tasks.

Chapter 4
Experimental Evaluation

4.1 Experiments

In this section, we discuss the experiment and its results in detail. We tested our model in Google Colab - GPU [1]. The code was implemented in NumPy 1.19.5 (Harris et al., 2020), PyTorch 1.7.1 (Paszke et al., 2017), and Hugging Face transformers 4.3.2 (Wolf et al., 2020) environment.

4.1.1 Dataset Selection and Information

We run tests on the SemEval-2014 task 4 citearticle dataset, which comprises restaurant customer reviews. There are two les in the dataset: training data and testing data. Each le contains user feedback as well as the goal values for each of the four activities (labels). Two evaluation metrics, Accuracy and Macro F1- Score, are used to evaluate the suggested approach's performance.

The statistics related to the dataset are represented in Table 4.1 for each task. In Table 4.1, each number in the Train and Test row represents the number of user reviews present in the dataset for every task. We have removed the sentences from the dataset, which leads to the conflict opinion polarity because the number of user reviews is small with conflict opinion polarity. The Fine-Grained ABOM (training and testing combined) dataset contains 2892 positive, 1001 negative, and 829 neutral sentences.

Table 4.1: Statistics of the Dataset

Dataset	ATE	Fine grained ABOM	OTE
Train	3,044	3,044	3,044
Test	800	800	800

4.1.2 Evaluation Metrics

To evaluate our model's performance, we will consider three evaluation strategies: Precision, Recall, and F1-score. In evaluation, we are using the Macro F1-score because it is used to deal with the problem of unbalanced class and Macro F1-score is calculated as average F1-score of each class (Karimi et al., 2020). The formulas for the evaluation methods are given below:

True Positives: It means when the model predicted YES and the actual output was also YES (Powers, 2020).

True Negatives: It means when the model predicted NO, and the actual output was NO (Powers, 2020).

False Positives: It means when the model predicted YES, and the actual output was NO (Powers, 2020).

False Negatives: It means when the model predicted NO, and the actual output was YES (Powers, 2020).

Precision: It measures the correctly identified positive cases from all the predicted positive cases. It is important when the costs of False Positives are high (Goutte & Gaussier, 2005).

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

Recall: It measures the correctly identified positive cases from all the actual positive cases. It is important when the cost of False Negatives is high (Goutte & Gaussier, 2005).

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

F1-Score: It is the harmonic mean of Precision and Recall (Goutte & Gaussier, 2005).

$$F1\ Score = \frac{2 * Precision * Recall}{Precision + Recall}$$

4.1.3 Hyperparameters

The selection of hyper-parameters is essential during the evaluation of model performance. We have used the BERT-BASE uncased model to conduct experiments on the dataset, where uncased denotes that the model is not case sensitive, which means it will not make any difference between words such as ENGLISH and English. We have executed our models several times to figure out what number of epochs and how much dropout probability yields the highest results for our approach. After executing the models on different parameters, we found that the dropout probability and epochs for the proposed approach should be 0.3 and 4, respectively. Also, we have used the Adam optimizer for better learning, and the learning rate is set to be 2e - 5 in our approach.

4.2 Results and Discussion

In this section, we compare our results with several state-of-the-art models on the SemEval-2014 task 4 restaurant dataset. Also, not all the models include the opinion term and aspect term and aspect term related sentiment polarity due to which comparison is based on every task. The results for Aspect Term Extraction (ATE), Opinion Term Extraction (OTE) are displayed in Table 4.2, while the results for the Aspect Based Sentiment Analysis(ABSA) are displayed in Table 4.3. The higher value of precision, Recall, and Macro - F1 denotes the better model. As we can see in Table 4.2, our proposed approach achieves good results on Opinion Term Extraction (OTE) and Aspect Term Extraction (ATE) tasks in terms of Macro-F1 and Precision. Also, BERT-ABSA worked better than previous approach in Aspect based Sentiment Analysis as displayed in Table 4.3. Our approach outperforms most previous BERT-based models such as BERT-PT, BAT, DomBERT, and BERT-LSTM/Attention in ATE and ABSA tasks.

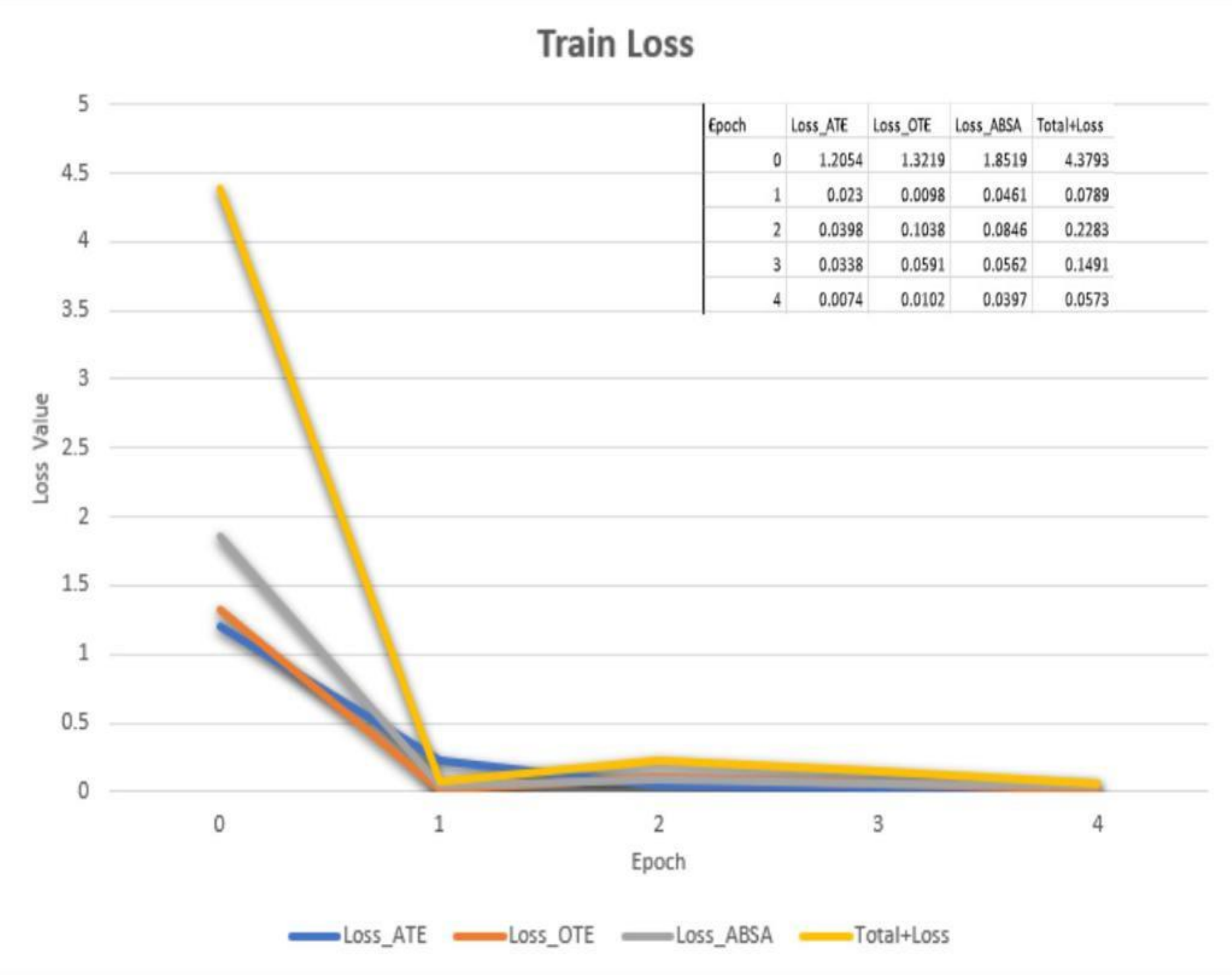
Table 4.2: Result of OTE and ATE

Model	OTE (Macro F1)	ATE (Macro F1)
MTNA (Xue et al., 2017)	-	84.01
RNSCN (Wang, Pan, Dahlmeier & Xiao, 2016)	81.67	82.12
JERE-MHS (Bekoulis, Deleu, Demeester & Develder,2018)	77.44	79.79
Spanmlt(Zhao, Huang, Zhang, Lu & Xue,2020)	93.98	87.40
BERT-PT (Xu et al., 2019)	-	77.97
IMN (He et al., 2019)	85.61	84.01
DomBERT (Xu et al., 2019)	-	77.21
Our Proposed Approach	85.06(1.00)	87.33(1.00)

Table 4.3: Fine Grained ABOM

Model	ABSA (Macro F1)
MGAN (Z. Li et al., 2019)	71.48
BERT-PT (Xu et al., 2019)	76.96
BERT-LSTM (Sun et al., 2019a)	72.52
BERT-Attention (Sun et al., 2019a)	73.38
BAT (Karimi et al., 2020)	73.7
DomBERT (Xu et al., 2019)	75.00
DomBERT (Xu et al., 2019)	73.15(1.00)

Also, all these BERT-based approaches trained their models up to 10 epochs while our models train up to only 5 epochs, due to which we achieve a better result with less computation time. In terms of the ATE and OTE tasks, we achieve the state-of-the-art result in very less training time (epochs) while in ABSA tasks, we achieve comparable results with other BERT-based approaches. As displayed in Figure 4.1, the feasibility of the decreasing loss function as we progress in the number of epochs are shown. As we reach the 5th epoch, the generated output by the model is almost equal to the targeted output and the loss function as close to zero as possible. Thus, providing us with the optimal time efficiency for implementing the model.



Epoch	Loss_ATE	Loss_OTE	Loss_ABSA	Total+Loss
0	1.2054	1.3219	1.8519	4.3793
1	0.023	0.0098	0.0461	0.0789
2	0.0398	0.1038	0.0846	0.2283
3	0.0338	0.0591	0.0562	0.1491
4	0.0074	0.0102	0.0397	0.0573

Figure 4.1: Training Loss

Chapter 5
Conclusions and Future work

5.1 Conclusions

In this thesis, we propose a novel BERT-based approach to perform subtasks of Aspect-Based Sentiment Analysis. The various approaches in recent years are developed to perform the ABSA tasks, and most of the approaches are focused on Aspect term extraction and Fine-grained ABSA. Also, the previous researchers built a separate model to perform each subtask of ABSA, which requires more training time and achieves less accuracy. To solve that problem, we include a Multi-Task learning model to extract the Aspect Terms and Opinion Terms simultaneously from the user reviews. Furthermore, we also perform Fine-grained ABSA using the BERT model. We have evaluated our model on the SemEval2014 restaurant benchmark dataset. We achieved better results during the evaluation of the model than the previous approaches on all the subtasks of ABSA such as Aspect Terms, Opinion Term detection, and Fine-grained ABSA. To conclude, our approach indicates that the use of Multitask, and BERT model enhances the performance as well as it requires less training time and we do not need to build separate model to perform each task in ABSA. Therefore, we achieve higher accuracy, in less training time and on the same platform.

5.2 Future Work

Thus, some possible future works are:

1. The proposed approach can be evaluated on multiple datasets to generalize and find the efficiency of the approach.
2. Additionally, one of the limitations of our approach is that it is unable to extract the aspect-opinion pair (e.g., (food, good)) from the reviews.
3. We can also include Adversarial and Post training on our approach to evaluate the model's performance.
4. There is a possibility to enhance the model's performance using soft parameter sharing in multitask learning approach.
5. By including aspect category task, we can enhance the performance of model on the ATE task and more user reviews can be considered as input, sometime user reviews do not contain any aspect terms, but they only have context towards specific features which can be detected in aspect category task.
6. Inclusion of a fourth term, "conflict" could also be a potential scope. For example: when a speaker says "The service was bad, the pizza was just as good", it provides a confusing/ conflicting terminology, as to what the speaker is intending to say, if the term "pizza" was "good" or "as bad as the service".

References

- Agrawal, R. S., & Srikant, R. (1994). R. fast algorithms for mining association rules. In Proceedings of the 20th international conference on very large data bases, vldb (pp. 487{499).

- Alammar, J. (2018, June). The illustrated transformer. Retrieved from https://jalammar.github.io/illustrated-transformer/

- Alghanmi, I., Espinosa Anke, L., & Schockaert, S. (2020, November). Combining BERT with static word embeddings for categorizing social media. In Proceedings of the sixth workshop on noisy user-generated text (w-nut 2020) (pp. 28{33). Online: As-sociation for Computational Linguistics. Retrieved from https://www.aclweb.org/ anthology/2020.wnut-1.5 doi: 10.18653/v1/2020.wnut-1.5

- Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola. (June 2021). Gated recurrent units (gru). Dive into Deep Learning. Retrieved from https://d2l.ai/ chapter recurrent-modern/gru.html

- Ba, J. L., Kiros, J. R., & Hinton, G. E. (2016). Layer normalization.

- Caruana, R. (1997). Multitask learning. Machine learning, 28 (1), 41{75. And

Chen, M.-S., Han, J., & Yu, P. S. (1996). Data mining: an overview from a database perspective. IEEE Transactions on Knowledge and data Engineering, 8 (6), 866{883. Cho, K., Van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H.,Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 .

- Clark, K., Khandelwal, U., Levy, O., & Manning, C. D. (2019). What does bert look at? an analysis of bert's attention. In Proceedings of the 2019 acl workshop blackboxnlp: Analyzing and interpreting neural networks for nlp (pp. 276{286).

- Cortes, C., & Vapnik, V.(1995). Support-vector networks. Machine learning, 20 (3), 273{297.

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

- Do, H. H., Prasad, P., Maag, A., & Alsadoon, A. (2019). Deep learning for aspect-based sentiment analysis: a comparative review. Expert Systems with Applications, 118 , 272{299.

- Ester, M., Kriegel, H.-P., Sander, J., Xu, X., et al. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In Kdd (Vol. 96, pp. 226{231).

- Forgy, E. W. (1965). Cluster analysis of multivariate data: efficiency versus interpretability of classifications. biometrics, 21 , 768{769.

- Goutte, C., & Gaussier, E. (2005). A probabilistic interpretation of precision, recall and f-score, with implication for evaluation. In European conference on information retrieval (pp. 345{359).

- Graves, A., Wayne, G., & Danihelka, I. (2014). Neural turing machines.

- Gupta, L. (Nov 2020). Differences between word2vec and bert. Medium. Retrieved from https://medium.com/swlh/differences-between-word2vec-and -bert-c08a3326b5d1

- Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., . . . Oliphant, T. E. (2020, September). Array programming with NumPy. Nature, 585 (7825), 357{362. Retrieved from https://doi.org/10.1038/s41586-020-2649 -2 doi: 10.1038/s41586-020-2649-2

- He, R., Lee, W. S., Ng, H. T., & Dahlmeier, D. (2019). An interactive multi-task learn-ing network for end-to-end aspect-based sentiment analysis. In Proceedings of the 57th annual meeting of the association for computational linguistics. Association for Computational Linguistics.

- Hochreiter, S., & Schmidhuber, J. (1997). Lstm can solve hard long time lag problems. Advances in neural information processing systems, 473{479.

- Hu, M., Zhao, S., Zhang, L., Cai, K., Su, Z., Cheng, R., & Shen, X. (2018). Can: Constrained attention networks for multi-aspect sentiment analysis. arXiv preprint arXiv:1812.10735

- HUILGOL, P. (AUG 2020). Top 4 sentence embedding techniques using python! Analytics Vidhya. Retrieved from https://www.analyticsvidhya.com/blog/2020/08/top-4 -sentence-embedding-techniques-using-python/

- Karimi, A., Rossi, L., Prati, A., & Full, K. (2020). Adversarial training for aspect-based sentiment analysis with BERT. CoRR, abs/2001.11316 . Retrieved from https:// arxiv.org/abs/2001.11316

- Kiritchenko, S., Zhu, X., Cherry, C., & Mohammad, S. M. (2014). Nrc-canada-2014:Detecting aspects and sentiment in customer reviews. SemEval 2014 , 437.

-

- Korovkinas, K., Danenas, P., & Garsva, G. (2019). Svm and k-means hybrid method for textual data sentiment analysis. Baltic Journal of Modern Computing, 7 (1), 47{60.

- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. nature, 521 (7553), 436{444. Li, N., Chow, C.-Y., & Zhang, J.-D. (2020). Seml: A semi-supervised multi-task learning framework for aspect-based sentiment analysis. IEEE Access, 8 , 189287{189297.

- Li, X., Bing, L., Zhang, W., & Lam, W. (2019). Exploiting BERT for end-to-end aspect-based sentiment analysis. CoRR, abs/1910.00883 . Retrieved from http://arxiv.org/abs/1910.00883

- Li, Z., Wei, Y., Zhang, Y., Zhang, X., & Li, X. (2019). Exploiting coarse-to- ne task transfer for aspect-level sentiment classification. In Proceedings of the aaai conference on artificial intelligence (Vol. 33, pp. 4253{4260).

- Ma, Y., Peng, H., & Cambria, E. (2018). Targeted aspect-based sentiment analysis via embedding common sense knowledge into an attentive lstm. In Thirty-second aaai conference on artificial intelligence.

- Maimon, O. Z., & Rokach, L. (2014). Data mining with decision trees: theory and applications (Vol. 81). World scientific.

- McCormick, C. (May 2019). Bert word embeddings tutorial. Retrieved from https:// mccormickml.com/2019/05/14/BERT-word-embeddings-tutorial/

- Miao, Z., Li, Y., Wang, X., & Tan, W.-C. (2020). Snippext: Semi-supervised opinion mining with augmented data. In Proceedings of the web conference 2020 (p. 617{628). New York, NY, USA: Association for Computing Machinery. Retrieved from https:// doi.org/10.1145/3366423.3380144 doi: 10.1145/3366423.3380144

- Misra, I., Shrivastava, A., Gupta, A., & Hebert, M. (2016, June). Cross-stitch networks for multi-task learning. In Proceedings of the ieee conference on computer vision and Pattern recognition(cvpr)

- Movahedi, S., Ghadery, E., Faili, H., & Shakery, A. (2019). Aspect category detection via topic-attention network. arXiv preprint arXiv:1901.01183 .

- Nazir, A., Rao, Y., Wu, L., & Sun, L. (2020). Issues and challenges of aspect-based sentiment analysis: A comprehensive survey. IEEE Transactions on A effective Computing.

- Parkhe, V., & Biswas, B. (2014). Aspect based sentiment analysis of movie reviews: finding the polarity directing aspects. In 2014 international conference on soft computing and machine intelligence (pp. 28{32).

- Pascual, F. (2019). A comprehensive guide to aspect-based sentiment analysis. Monkey Learn. Retrieved from https://monkeylearn.com/blog/aspect-based-sentiment -analysis/

- Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., . . . Lerer, A. (2017). Automatic differentiation in pytorch. nips-w. In Proceedings of the 31st conference on neural information processing systems (nips 2017), long beach, ca, usa (pp. 4{9).

- Patel, M. (2021). Neural network-based multi-task learning for product opinion mining (Unpublished doctoral dissertation). University of Windsor (Canada).

- Pontiki, M., Galanis, D., Papageorgiou, H., Androutsopoulos, I., Manandhar, S., Al-Smadi, M., . . . others (2016). Semeval-2016 task 5: Aspect based sentiment analysis. In International workshop on semantic evaluation (pp. 19{30).

- Pontiki, M., Galanis, D., Pavlopoulos, J., Papageorgiou, H., Androutsopoulos, I., & Manandhar, S. (2014, 01). Semeval-2014 task 4: Aspect based sentiment analysis. Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014), 27-35. doi: 10.3115/v1/S14-2004

- Powers, D. M. (2020). Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. arXiv preprint arXiv:2010.16061 .

- Rana, T. A., & Cheah, Y.-N. (2017). Improving aspect extraction using aspect frequency and semantic similarity-based approach for aspect-based sentiment analysis. In Inter-national conference on computing and information technology (pp. 317{326).

- Ray, S. (September 2017). 6 easy steps to learn naive bayes algorithm with codes in python and r. Analytics Vidhya.

- Ruder, S. (2017). An overview of multi-task learning in deep neural networks. arXiv preprint arXiv:1706.05098 .

- Sercu, T., Puhrsch, C., Kingsbury, B., & LeCun, Y. (2016). Very deep multilingual convolutional neural networks for lvcsr. In 2016 ieee international conference on acoustics, speech and signal processing (icassp) (pp. 4955{4959).

- Sharma, R., Nigam, S., & Jain, R. (2014). Mining of product reviews at aspect level. arXiv preprint arXiv:1406.3714 .

- Sun, C., Huang, L., & Qiu, X. (2019a). Utilizing bert for aspect-based sentiment analysis via constructing auxiliary sentence. arXiv preprint arXiv:1903.09588 .

- Sun, C., Huang, L., & Qiu, X. (2019b). Utilizing bert for aspect-based sentiment analysis via constructing auxiliary sentence.

- Vandenhende, S., Georgoulis, S., Van Gansbeke, W., Proesmans, M., Dai, D., & Van Gool, L. (2021). Multi-task learning for dense prediction tasks: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence.

- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., . . . Polo-sukhin, I. (2017). Attention is all you need. arXiv preprint arXiv:1706.03762

- Weston, J., Chopra, S., & Bordes, A.(2014).Memory networks. arXiv preprint arXiv:1410.3916 .

- Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., . . . Rush, A. M. (2020, October). Transformers: State-of-the-art natural language processing. In Proceedings of the 2020 conference on empirical methods in natural language processing: System demonstrations (pp. 38{45). Online: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/2020.emnlp-demos.6

- Xu, H., Liu, B., Shu, L., & Yu, P. S. (2019). Bert post-training for review reading comprehension and aspect-based sentiment analysis. arXiv preprint arXiv:1904.02232 .

- Xue, W., & Li, T. (2018). Aspect based sentiment analysis with gated convolutional networks. arXiv preprint arXiv:1805.07043 .

- Xue, W., Zhou, W., Li, T., & Wang, Q. (2017). Mtna: A neural multi-task model for aspect category classification and aspect term extraction on restaurant reviews. IJCNLP 2017 , 151.

- Yuan, X., Li, L., & Wang, Y. (2019, 02). Nonlinear dynamic soft sensor modeling with supervised long short-term memory network. IEEE Transactions on Industrial Informatics, PP, 1-1. doi: 10.1109/TII.2019.2902129

- Zainuddin, N., Selamat, A., & Ibrahim, R. (2018). Hybrid sentiment classification on twitter aspect-based sentiment analysis. Applied Intelligence, 48 (5), 1218{1232.

- Zhang, L., & Liu, B. (2014). Aspect and entity extraction for opinion mining. In Data mining and knowledge discovery for big data (pp. 1{40). Springer.

- Zhang, Y., & Yang, Q. (2021). A survey on multi-task learning. IEEE Transactions on Knowledge and Data Engineering.

- Zhang, Y., & Yeung, D.-Y. (2012). A convex formulation for learning task relationships in multi-task learning.

- Zhou, J., Huang, J. X., Chen, Q., Hu, Q. V., Wang, T., & He, L. (2019). Deep learning for aspect-level sentiment classification: Survey, vision, and challenges. IEEE access, 7 , 78454{78483.

- Zhou, X., Wan, X., & Xiao, J. (2015). Representation learning for aspect category detection in online reviews. In Proceedings of the aaai conference on articial intelligence (Vol. 29).