UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft
Faculty for Computer Science, Electrical Engineering and Mathematics
Department of Computer Science
Research Group Computational Social Science

Master's Thesis

Submitted to the Computational Social Science Research Group
in Partial Fulfilment of the Requirements for the Degree of

Master of Science

Cross-domain Aspect-based Sentiment Analysis with Multimodal Sources

by
Pavan Kumar Sheshanarayana

First Reviewer:
Jun.-Prof. Dr. Henning Wachsmuth
Department of Computer Science
Paderborn University

Second Reviewer:
Dr. Theodor Lettmann
Department of Computer Science
Paderborn University

Thesis Supervisor:
Wei-Fan Chen

Paderborn, August 10, 2022

Declaration

I hereby declare that the thesis I am submitting is entirely my own original work except where otherwise indicated.

Paderborn, 10/082022

Place, Date

Signature

Abstract.

In this work, we examined the effectiveness of Semi-supervised learning for the E2E-ABSA problem. With two Semi-supervised methods, we also performed cross-domain analysis across five domains. Finally, we also performed E2E-ABSA experiments with erroneous transcriptions and two other settings to evaluate the outcome. As part of the first research task, the pre-trained BERT model with additional layers is used for the downstream task of E2E-ABSA sequence labeling. The model is then incorporated as part of two Semi-supervised methods, namely the Self-Training and the Tri-Training. With Self-Training, one of the models achieved a Macro-F1 increment of 28.67% over the baseline, and with Tri-Training, one of the models improved on its baseline by a Macro-F1 percentage of 22.53%. Altogether, on the Self-Training and Tri-Training experiments with four models, one model from Self-Training and two different models from Tri-Training outscored its baseline. For the second research task of cross-domain analysis, we evaluated the Semi-supervised models on their domain adaptation possibilities. On the individual evaluation of sentiment and aspect, we found the domain adaptation in terms of sentiment to be reasonable on four domains. However, the evaluation of Aspect terms conducted on all domains combined was inadequate and indicated scope for improvement. On the final research task of evaluating E2E-ABSA under erroneous transcriptions from an external Automatic Speech Recognition system (ASR), we found that even amidst a notable error rate, the E2E-ABSA determination remained unaffected. This observation remained true even while evaluating the sentiment on instances containing erroneous and spontaneous speech instances. The experiments with these instances outscored even the ones involving the gold standard data. We establish that transcriptions obtained from an ASR are capable enough for deducing aspects and sentiment, despite errors. We also observed that in the runs involving one-fifth of the original unlabeled data and faulty transcriptions, the Macro-F1 scores improved over the experiments that included complete unlabeled data. The significance of the moderate addition of unlabeled samples for semi-supervised works might also be an outcome of this work. In conclusion, this work expands the other research on Semi-supervised methods for E2E-ABSA by signifying their constructive results, leading to improved performances and generalization into other domains.

Contents

1 Introduction **1**
1.1 Motivation . . . 1
1.2 Problem Statement . . . 2
1.3 Outline . . . 4

2 Literature **5**
2.1 Natural Language Processing . . . 5
2.2 Text Classification . . . 5
2.3 Semi Supervised Learning . . . 6
2.3.1 Self-Training . . . 6
2.3.2 Tri-Training . . . 7
2.4 Language Models . . . 9
2.4.1 Recurrent Neural Networks . . . 9
2.4.2 Long Short-Term Memory Networks . . . 9
2.4.3 Gated Recurrent Units . . . 10
2.4.4 The Transformer . . . 10
2.4.5 BERT . . . 12
2.5 Aspect-based Sentiment Analysis . . . 13
2.6 Semi-Supervised Approaches for ABSA . . . 14

3 Datasets & Evaluation Techniques **17**
3.1 E2E-ABSA Datasets . . . 17
3.2 CMU-MOSEI . . . 18
3.3 Evaluation Techniques . . . 20
3.3.1 Evaluation on Sentiment . . . 20
3.3.2 Evaluation on Aspect . . . 21

4 Approaches **25**
4.1 Baseline for E2E-ABSA . . . 25
4.1.1 Linear layer . . . 26
4.1.2 Gated Recurrent Unit . . . 26
4.1.3 Self-Attention Networks . . . 27
4.1.4 Transformer Layer . . . 27
4.2 E2E-ABSA with Semi-Supervised Approaches . . . 28
4.2.1 E2E ABSA with Self-Training . . . 28
4.2.2 E2E ABSA with Tri-Training . . . 28
4.3 ABSA across Domains using Semi-Supervised Approaches . . . 29

4.3.1 Domain Classification using Topic Modeling 30
4.3.2 E2E ABSA with Self-Training 30
4.3.3 E2E ABSA with Tri-Training 31
4.4 E2E-ABSA across Erroneous Transcriptions 31
4.4.1 Preparing Transcriptions from AWS Transcribe 33
4.4.2 Semi-Supervised Modeling for E2E-ABSA 34

5 Experiments, Results and Analysis 35
5.1 Experiments for Baseline Model 35
5.2 Experiments with Semi-Supervised Approaches 36
5.2.1 E2E-ABSA with Self-Training 36
5.2.2 E2E-ABSA with Tri-Training 38
5.3 E2E-ABSA across Domains 44
5.3.1 Domain Segregation using Topic Modeling 44
5.3.2 E2E-ABSA across Domains 45
5.4 E2E-ABSA across Erroneous Transcriptions 54
5.4.1 E2E-ABSA across AWS Transcriptions and Gold Standard 55
5.4.2 Evaluating Sentiment with Erroneous Transcriptions 57
5.4.3 Evaluating Sentiment on Spontaneous instances 58

6 Conclusion 61
6.1 Future works 62

Bibliography 63

1 Introduction

1.1 Motivation

Sentiment is a long-term disposition evoked when a person encounters a specific topic, person, or entity (Deonna and Teroni, 2012). Understanding these inclinations has been a topic of interest for research and subsequently has also been a driving force for many modern applications. Sentiment analysis hence aims to uncover the underlying sentiment that the reviewer holds against an entity. Earlier sentiment analysis works were predominantly carried out over text data (Soleymani et al., 2017). Textual data in terms of movie reviews, restaurant reviews, etc., were largely available over the web which made these analyses possible in a mainstream manner.

Recent advances in the field of deep learning have enabled great progress in the sentiment analysis field. According to the survey by Soleymani et al. (2017), in terms of twitter-based sentiment analysis competitions, deep learning-based approaches outscored their counterparts in SemEval challenges (Pontiki et al., 2016; Tang et al., 2014). Contextual info which is now known to be crucial for these analyses has been greatly aided with the introduction of Long Short-Term Memory networks (popularly known as LSTMs) (Schmidhuber et al., 1997). The introduction of attention (Vaswani et al., 2017) based Sequence models has also been known to be helpful.

Lately, with people's opinions being shared online through forms other than text, the focus has shifted towards audio-video sources. This, along with the limitations of text-only analysis of sentiment (such as the presence of colloquialisms) and concurrent advances made in deep learning, has today enabled multiple research to venture into sources other than language alone for more info. This has in turn enabled many state-of-the-art approaches to derive sentiment better. Hence, in this work, it is also our intention to look for sources of different modalities to derive the sentiment.

In addition to the above, we also delve into two other perspectives of sentiment analysis for this thesis. The first is a sub-field of sentiment analysis known widely as the Aspect Based Sentiment Analysis (ABSA). Here, Aspect Mining, sometimes abbreviated as AM, and Aspect Sentiment Classification, abbreviated as ASC are the two goals. These are also the point of interest, either solely or together to many research works.

The majority of works in the research area of ABSA are set in the supervised setting, for either of its sub-tasks. But there have been a few approaches in unsupervised backdrop even though most of these works are focused on the sole task of AM (Jo and Oh, 2011; He et al., 2017; Zhao et al., 2010). To the best of our information, in the supervised setting, there are no works for ABSA when there are multiple domains involved. A well-known countermeasure for such cases, in general, is known to be the incorporation of a semi-supervised approach. Hence this will be the other perspective that will be the focus of this study.

1.2 Problem Statement

The field of sentiment analysis is extremely broad, with studies focusing on many domains for various applications and use cases. The sub-field of sentiment analysis, ABSA, comes with three existing research problems. The original ABSA, aiming to predict sentiment against a given aspect entity, the Aspect Oriented Opinion Words Extraction and the E2E-ABSA (Li et al., 2019c). In the E2E-ABSA challenge, the goal is to accomplish both the sub-tasks of ABSA together. Over the years, there have been many challenges or competitions in several domains to undergo each of these tasks, either solely or together. The SemEval challenges (Pontiki et al., 2016, 2015, 2014) whose publishers have their datasets released can be said to be the most widely used datasets here. The datasets in question are published over three consecutive years (Task-4 from 2014, Task-12 from 2015, and Task-5 from 2016) and comprise reviews for the restaurant, laptops, etc. Here, sentences of varied lengths are annotated for sentiment and words that best describe the aspect to which the said sentiment is associated with. The goal is to solve the detection of both the aspect and sentiment together as a joint task rather than two separate tasks.

At the time of writing this, the research works involving a Semi-supervised approach to the ABSA problem are majorly on the AM task. Few works in the domain use prior domain knowledge to aid an unsupervised topic model. This, however, requires domain-specific insights. Xu et al. (2018) used pre-training to learn domain-specific word embeddings from unlabeled reviews which are then used in a supervised learning setting. For the ASC task, a few works (Lau et al., 2014; Hussain and Cambria, 2018; Cambria et al., 2015, 2016) use commonsense knowledge networks which can also be a case of a semi-supervised approach. Miao et al. (2020) used data augmentation to perform Semi-supervised opinion mining.

To the best of our information, the work, SEML (Li et al., 2020b) is the only Semi-supervised approach to propose an end-to-end deep learning Semi-supervised framework that does both the AM and ASC tasks. The work, however, does not provide cross-domain dataset analysis as the dataset considered for both the labeled and unlabeled sets were from the same domain.

Hence with the Semi-supervised setting for the thesis, we also specifically focus on the following things. We tackle the task of an E2E-ABSA problem. For this, we leverage the SemEval dataset re-prepared by Li et al. (2019b) as part of our labeled dataset. The unlabeled data counterpart for this sub-goal would be obtained from the CMU-MOSEI (Zadeh et al., 2018), with individual video captions being used for training. The CMU-MOSEI consists of videos focused on multiple topics such as movie reviews, short speeches on politics, investments, etc. These different subject matters act as different domains against which sentiment prediction would be made. It is to be noted that, though we derive the data for the above task from sources of different modalities, we do not model different modalities for the inference of sentiment. This is to say that only the sources would be from more than one modality even though we finally base all our tasks, as part of the thesis, solely on textual data eventually captured.

Hence, with the above setting defined, we devise the following research tasks for this work.

E2E-ABSA with Semi-supervised approaches As part of this task, first, we create a few baseline models capable of tackling the E2E-ABSA. We then incorporate these models with two Semi-supervised methods to record their E2E-ABSA performances.

For the baseline, we create a computational model based on the *bert-base-uncased* with a few additional layers. With BERT capable of encoding representations of words in a sentence, we then pass on these representations to the additional layers capable of labeling every word for aspect and sentiment. For the latter, we make use of the work from Li et al. (2019c). The baseline models would be trained in a supervised manner from the SemEval datasets belonging to Restaurant and Laptop domains.

Next, we make use of these models and employ two Semi-supervised methods. Specifically, we make use of the Self-Training and the Tri-Training methods with unlabeled data from CMU-MOSEI to evaluate E2E-ABSA performances.

Hence, the outcome of this task would be the results of the baseline implementations compared against the two mentioned Semi-supervised methods.

Cross-domain analysis of E2E-ABSA This task would be an extension of the previous task. The CMU-MOSEI dataset used as the unlabeled data counterpart contains data belonging to various domains. Since the distinction between them are not readily available, we first devise a method using Topic modeling to classify them into various domains.

We then leverage the models from two Semi-supervised methods to predict aspects and sentiment to these domains that we then evaluate.

Hence, the outcome of this task is to segregate domains and predict aspects and sentiments for them. The predictions for each domain are then evaluated on the sentiment separately. The aspect evaluation, however, for the lack of sufficient labels is done on all domains combined.

E2E-ABSA across Erroneous transcriptions As part of the final research task, we investigate E2E-ABSA with faulty transcriptions. For this, we first obtain the transcriptions for the videos belonging to CMU-MOSEI from an external Automatic Speech Recognition system (ASR). With the models from the Semi-supervised methods, we obtain E2E-ABSA predictions with the transcriptions from the ASR and the gold standard as unlabeled data, in separate runs.

As part of the other adjacent experiments in this task, we evaluate for sentiment in the additional setting of just the erroneous transcriptions and the instances of spontaneous speeches involving incomplete transcriptions, repetitions, hesitation, etc.

Hence the outcome of this task is the direct comparison of E2E-ABSA results with runs from the ASR transcriptions and gold standard. Additionally, the results for the evaluation on sentiment under the setting of just the erroneous transcriptions and spontaneous speeches would be made.

1.3 Outline

With the research tasks for the work defined, we now give the outline of this entire work. In Chapter 2, we present the overview of relevant literature referenced by our work. We also present concepts that are important to this work. In Chapter 3, we explain in detail the datasets used as part of this research and present the evaluation methods for evaluating one of our research tasks. In Chapter 4, we present the approaches used for various research tasks of this work. We explain the baseline models in section 4.1. In section 4.2, we explain how we use the two Semi-supervised methods for our work. In section 4.3, we present the approach to the second research task of this work, the cross-domain analysis of E2E-ABSA. We present the approach to the final research task of our work, E2E-ABSA across Erroneous transcriptions, in section 4.4.

In Chapter 5, we give out additional experiment settings wherever utilized and follow them up with the results and analysis. Finally, we conclude our work in Chapter 6 and present possible future research that can extend this work.

2 Literature

2.1 Natural Language Processing

Natural Language Processing or NLP, a branch of Computer science and subsequently Artificial Intelligence, deals with equipping machines with the ability to read, understand and interpret meaning from Human discourse and, in many cases, reproduce it in some form. NLP essentially involves textual data and finds its application in many use cases such as Language Translation, Question Answering, Smart assistants, Sentiment Analysis, and so on. Classical NLP approaches included rule-based, lookup-based methods to address a problem, while the modern NLP uses statistical methods such as Machine Learning to infer the patterns and regularities in the text to generate a probabilistic structure to accomplish the task in hand.

NLP is also used in the context of various unsupervised and Reinforcement learning. However, in our case, the supervised approach is of interest. In supervised NLP problems, the models are trained on annotated datasets that facilitate learning. Text classification, a classical supervised NLP problem uses labels or annotations to assign a certain category to the text or part of the text. Sentiment Analysis, a major text classification problem and one of the key elements of this work, assigns sentiment to text.

In the next section, we briefly discuss the text classification problem, especially, in the context of Sentiment Analysis.

2.2 Text Classification

Earlier Text classification methods used shallow learning models such as K-NN-based classification, Support Vector machines (SVM), Decision Trees, etc. These classifiers were mainly general purpose in nature and hence were not task-specific (Gasparetto et al., 2022). Some of the earlier works in the field of Sentiment Analysis employed these classifiers. To name a few, Kang et al. (2012) presented a Naive-Bayes classifier that classified sentiments on a restaurant dataset. Li and Li (2013) used Support Vector machines (SVM) for opinion classification on Twitter platform data. Some works used Decision Tree classifiers with algorithms such as CART, ID3, C5.0, C4.5 (Revathy and Lawrance, 2017; Hssina et al., 2014; Singh and Gupta, 2014; Patel and Prajapati, 2018) for opinion mining.

However, with the introduction of word embeddings such as Word2Vec and GloVe, them paired with simple classifiers achieved better results (Gasparetto et al., 2022). Gasparetto et al.

(2022) summarize them in an informal explanation that this is akin to how a person would likely be able to label a piece of text if they understood what it meant.

Further, Deep-learning-based models such as the Recurrent neural networks (RNNs), Long short-term memory networks (LSTMs), captured better representations of text than the above counterparts and soon achieved state-of-the-art results (Kowsari et al., 2019). These models are also called sequence models since they view text as a sequence of words/characters. They are also employed in our thesis for the E2E-ABSA task and are explained in further sections.

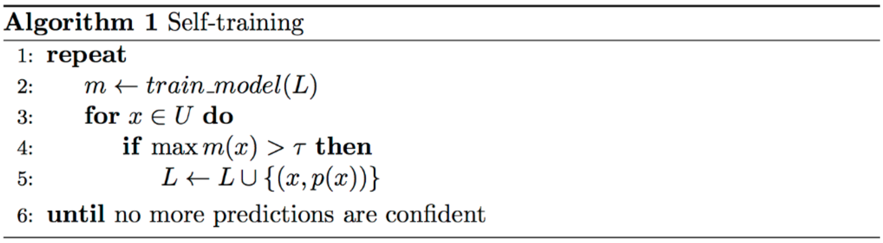
2.3 Semi Supervised Learning

Supervised learning requires labeled data. But in circumstances where their availability is scarce or there are more unlabeled data than the labeled counterparts, the supervised learning process becomes difficult. In such cases, Semi-supervised learning becomes a viable option (Zhu and Goldberg, 2009). It also becomes a feasible option in real-world applications since labeling is more expensive than the collection of data due to the former being labor-consuming.

While there are numerous Semi-supervised approaches proposed over the years, we will present the ones that are relevant to this work.

2.3.1 Self-Training

Self-Training (Yarowsky, 1995; McClosky et al., 2006; Reichart and Rappoport, 2007) is one of the earliest introduced Semi-supervised approaches that leverage the model's prediction of the unlabeled samples for further learning.



```
Algorithm 1 Self-training
repeat
    $m \leftarrow train\_model(L)$
    for $x \in U$ do
        if $\max m(x) > \tau$ then
            $L \leftarrow L \cup \{(x, p(x))\}$
until no more predictions are confident
```

Figure 2.1: Self-Training Algorithm [1]

In Self-Training, a model m is initially trained on a labeled training set L. After this, the model provides predictions $m(x)$ in the form of a probability distribution over C classes for all examples x from an unlabeled data set U. If the probability assigned to the most likely class is higher than a pre-considered threshold τ, then x is added to the labeled examples with the label $p(x) = argmax(m(x))$. The corresponding sample then becomes a pseudo-labeled sample for the next iteration of training. This is then carried out at each iteration. The process could generally be either repeated for a fixed number of iterations or until the predictions on unlabeled examples are no more confident. This instantiation is the most widely used and shown in Figure2.1[1].

Van Asch and Daelemans (2016) believe that Self-Training remains a controversial approach to Semi-supervised learning particularly when its effectiveness is taken into consideration. Sagae (2010) opines that Self-Training is at its most effective when the training data and the test data, which in this case, the unlabeled samples used at each iteration, are sufficiently dissimilar.

Hence Self-Training can be seen as a form of learning for domain adaptation wherein the training and test data are from different domains, though the notion of the domain itself remains debatable (Van Asch and Daelemans, 2016).

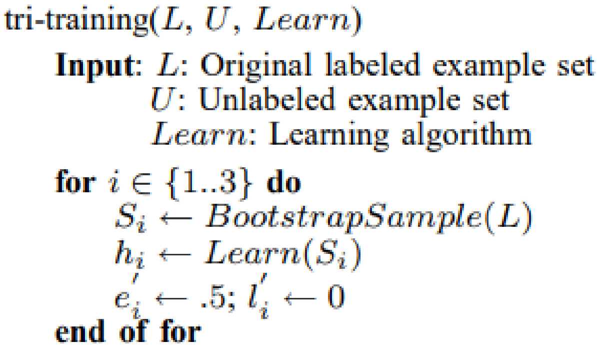
[1] from: `https://ruder.io/semi-supervised/index.html`

In this work, we also consider the application of Self-Training to be a domain adaptation task. Here, essentially we evaluate the Self-Training method across multiple domains thereby providing cross-domain analysis. Further details on this, will be explained in further chapters.

2.3.2 Tri-Training

Tri-Training, first introduced by Zhou and Li (2005) is essentially a form of Multi-view Training. In Multi-view Training, the agenda is that different models trained on different views of data help each other in improving their performances. This is a collaborative approach that aims to ideally, complement the learning made by the other models. The views may come from training these models with different features, by the architecture of the models themselves, or by the data on which they are trained.

The algorithm for the Tri-Training is presented in the figures 2.2 and 2.3. There are two phases to the implementation. In the first phase, the three classifiers are initially trained by Bootstrap sampling of the labeled training data, as shown in Figure 2.2.



```
tri-training($L$, $U$, $Learn$)
    Input: $L$: Original labeled example set
           $U$: Unlabeled example set
           $Learn$: Learning algorithm
    for $i \in \{1..3\}$ do
        $S_i \leftarrow BootstrapSample(L)$
        $h_i \leftarrow Learn(S_i)$
        $e_i^{'} \leftarrow .5; l_i^{'} \leftarrow 0$
    end of for
```

Figure 2.2: Tri-Training Algorithm Initialization phase, from Zhou and Li (2005)

Then in the second phase, the update phase, shown in Figure 2.3, each of the models are checked for whether they are eligible for re-training with a portion of the unlabeled sample added for them at that iteration. The criteria that any model h is eligible for this is based on the error it generates at the previous iteration. If the error is lesser than that of other previous ones, they are to continue training. For this purpose, when a model is trained at an iteration, it is immediately checked for an error measurement at the beginning of the next iteration. The iteration at which this does not hold, the model stops re-training.

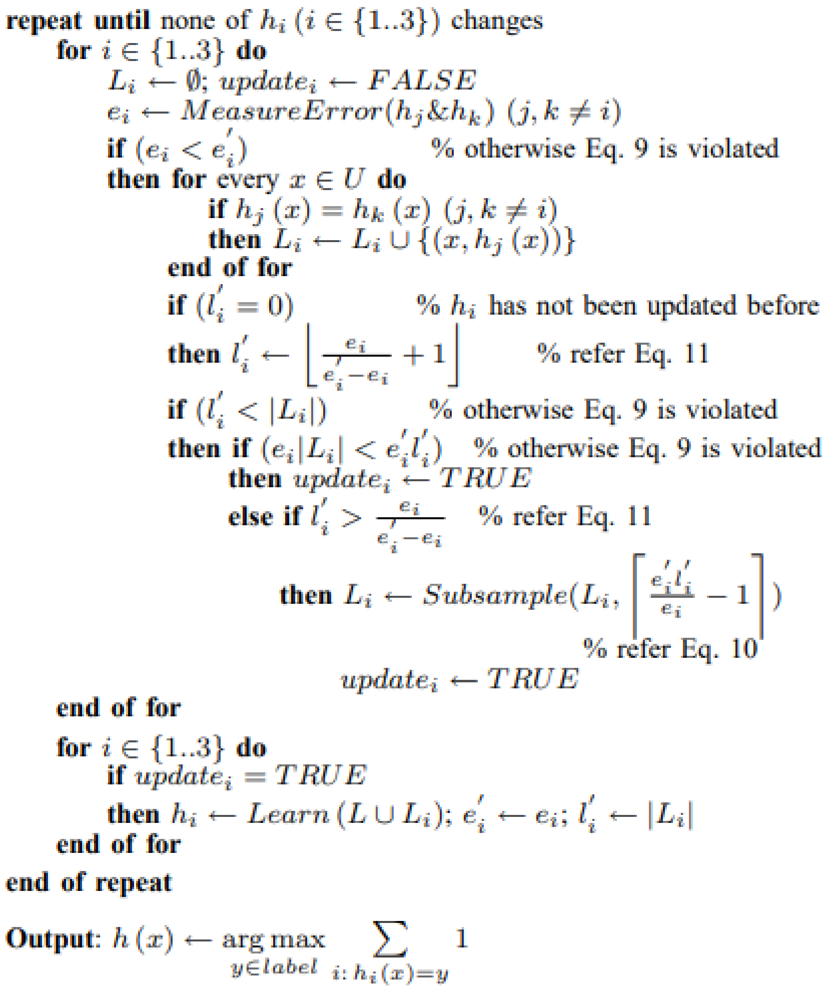
Further, there is also a criterion to check whether an unlabeled sample could be added for training to a classifier at any iteration. This is shown in the following relation:

$$L[t] * e[t] < L[t-1] * e[t-1] \tag{2.1}$$

where L denotes the number of samples for training and e denotes the classification error at iteration t and $t-1$. The equation suggests that the combination of the number of unlabeled samples $L[t]$ and error $e[t]$ at any iteration t should always be lesser than its previous iteration.

The main aim to limit the samples added for training is to compensate for the classification noise rate. This is based on the idea that the increase in classification noise rate can be compensated if the amount of newly labeled samples added is just sufficient.

Ruder and Plank (2018), in their work of POS tagging under Domain shift, found that Tri-Training outperformed even the then other recent state-of-the-art approaches. Hence, we try to investigate how well the Tri-Training approach (i) tackles the E2E-ABSA problem when



```
repeat until none of $h_i$ ($i \in \{1..3\}$) changes
    for $i \in \{1..3\}$ do
        $L_i \leftarrow \emptyset$; $update_i \leftarrow FALSE$
        $e_i \leftarrow MeasureError(h_j \& h_k)$ $(j, k \neq i)$
        if $(e_i < e_i^{'})$                % otherwise Eq. 9 is violated
        then for every $x \in U$ do
                if $h_j(x) = h_k(x)$ $(j, k \neq i)$
                then $L_i \leftarrow L_i \cup \{(x, h_j(x))\}$
            end of for
            if $(l_i^{'} = 0)$              % $h_i$ has not been updated before
            then $l_i^{'} \leftarrow \left\lfloor \frac{e_i}{e_i^{'} - e_i} + 1 \right\rfloor$       % refer Eq. 11
            if $(l_i^{'} < |L_i|)$            % otherwise Eq. 9 is violated
            then if $(e_i |L_i| < e_i^{'} l_i^{'})$   % otherwise Eq. 9 is violated
                then $update_i \leftarrow TRUE$
                else if $l_i^{'} > \frac{e_i}{e_i^{'} - e_i}$     % refer Eq. 11
                    then $L_i \leftarrow Subsample(L_i, \left\lceil \frac{e_i^{'} l_i^{'}}{e_i} - 1 \right\rceil)$
                                                % refer Eq. 10
                         $update_i \leftarrow TRUE$
    end of for
    for $i \in \{1..3\}$ do
        if $update_i = TRUE$
        then $h_i \leftarrow Learn(L \cup L_i)$; $e_i^{'} \leftarrow e_i$; $l_i^{'} \leftarrow |L_i|$
    end of for
end of repeat
Output: $h(x) \leftarrow \arg\max_{y \in label} \sum_{i:\, h_i(x) = y} 1$
```

Figure 2.3: Tri-Training Algorithm Update phase, from Zhou and Li (2005)

compared to that of supervised setting and (ii) tackles E2E-ABSA for cross domain settings. Further details are provided in the next chapters.

2.4 Language Models

Language modeling is the task of predicting the next word or sequence of words with an already existing sequence available. While the goal is to assign a likelihood to a sentence, it can also be considered a task where the next word is predicted based on the sequence at hand. These models are built on an assumption that the likelihood of a word depends entirely on the n words that precede it. Hence these are also sometimes called $n - gram$ models.

While the language models were initially successful, they had two main problems[2]. First is the inability to capture context info[2]. Context can hugely influence the choice of the next word and language models cannot capture them. The next main problem is the scale and scarcity[2]. As the size i.e., n increases, the number of possible permutations rises steeply, even though most permutations never occur in the text. Non-occurring n-grams then create a sparsity problem. The granularity of the probability distribution can be very low; as a result, most of the words have the same probability.

Neural Network-based language models combat the sparsity problem by embedding layers, creating an arbitrary-sized vector of each word. These embeddings then capture the needed semantic, hierarchical, and context info[2].

2.4.1 Recurrent Neural Networks

The Recurrent Neural networks or RNNs were able to solve the sparsity problem prevalent in the language models. RNNs are also known to be part of sequence models since they capture information as a sequence of units. In a textual setting, these units could be words or characters, depending on the problem.

Some earlier works (Socher et al., 2013; Poria et al., 2016; Dragoni and Petrucci, 2017) used RNNs for the sentiment classification tasks. But it quickly came to be known that RNN, though capable of carrying contextual info, was ultimately limited when the task required capturing contexts at a longer length. It also suffers from the vanishing-gradient problem which meant that the gradients became smaller when they reached earlier layers during backpropagation. These gradients that facilitate learning would then be inconsequential if they became too low.

2.4.2 Long Short-Term Memory Networks

Long short-term memory networks or LSTMs, introduced by Hochreiter and Schmidhuber (1997) were very effective in retaining long-term dependencies using the mechanism of gates. The architecture of LSTM is shown in Figure2.4.

Each cell shown in the figure corresponds to one unit in the LSTM and each unit comprises cells and states. The cells are memory blocks that can be transferred to another LSTM cell by the use of gates. These gates include the forget gate, input gate, and output gate.

The forget gate is responsible for removing info from the cell state. It takes in as input, the h_{t-1} (info from the previous hidden state) and x_t (current input). A sigmoid function used on top of both input vectors decides whether the cell state should retain the value (if 1) or not (if 0).

[2] `https://informationmatters.org/2022/05/the-power-and-the-pitfalls-of-large-language-models-a-fireside-chat-with-ricardo-baeza-yates/`

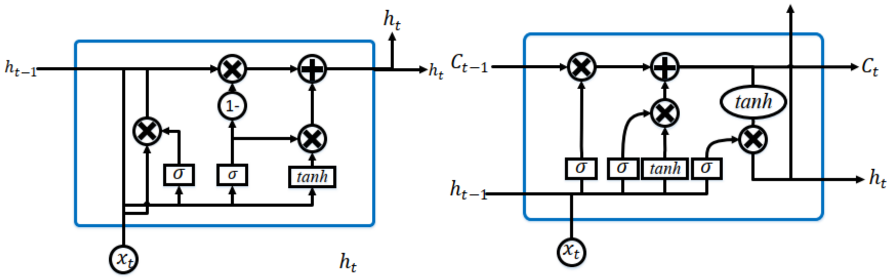


Figure 2.4: Architecture GRU vs LSTM, from Kowsari et al. (2019)

The input gate is responsible for adding info to the cell state. The gate takes in the input h_{t-1} and x_t.

The output gate is then responsible to pass on values to the next LSTM cell, which it deems can be valuable.

The other variant of LSTM called the Bidirectional-LSTMs (Bi-LSTMs) uses information from both the previous and the next time step. Both LSTMs and Bi-LSTMs have been a part of various sentiment analysis works (Vateekul and Koomsubha, 2016; Uysal and Murphey, 2017; Rao et al., 2018) and have managed to be generally better than RNNs.

2.4.3 Gated Recurrent Units

Gated Recurrent Units or GRUs (Cho et al., 2014) are a variation of RNNs and are similar to LSTMs in their architecture, shown in Figure 2.4.

As opposed to the LSTMs, GRUs employ two gates, namely the update gate and the reset gate. There is also only the hidden state and no cell state.

The update gate functionally is similar to that of the LSTMs, while the reset gate is responsible for deciding whether the info from the previous hidden state is vital to be passed on or not. This can be seen in the calculation of the current hidden state which is the element-wise product of the reset gate and the previous hidden state.

2.4.4 The Transformer

With the paper "Attention is all you need", Vaswani et al. (2017) introduced the Transformer architecture. The architecture consisted of Encoder-Decoder stack. Here, the encoder maps an input sequence of symbol representations (x_1, ..., x_n) to a sequence of continuous representations z = (z_1, ..., z_n). Given z, the decoder then generates an output sequence (y_1, ..., y_m) of symbols one element at a time (Vaswani et al., 2017)

Encoder The encoder stack comprises two sub-layers. The Multihead self-attention and a fully connected feed-forward network. A special encoding step performed before the first layer of the encoder ensures that the embeddings for the same word appearing at a different position in the sentence will have a different representation. This step is called positional encoding (Gasparetto et al., 2022). Also, here, the input embeddings are three different weight vectors to generate different representations known as Q (query), K (key), and V (value), given below:

$$Q = X.W_q, K = X.W_k, V = X.W_v \ (from\ Vaswaniet\ al.\ (2017)) \tag{2.2}$$

where W_q, W_k, W_v $\in$R$^{dim \times d_k}$ are learnable parameters and X $\in$R$^{N \times dim}$ are the embeddings for the input sequence and Q, V, K $\in$R$^{N \times d_k}$.

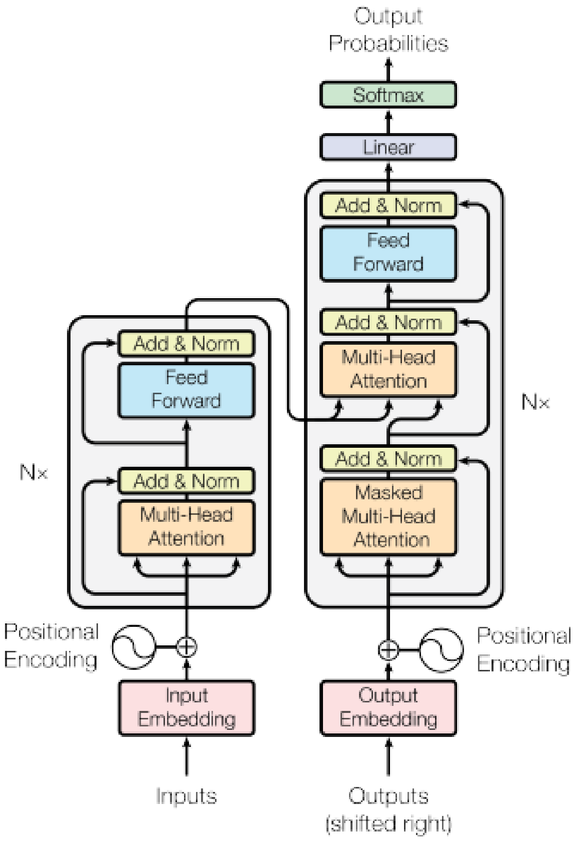


Figure 2.5: Transformer Architecture, from Vaswani et al. (2017)

The Attention which is called as Scaled Dot-Product attention in the original paper is then calculated by the following equation:

$$Attention(Q, K, V) = softmax(QK^T/\sqrt{d_k}).V \quad (from\ Vaswaniet\ al.\ (2017)) \tag{2.3}$$

While the above denotes the concept of attention, the authors also propose the idea of *'Multi-head Attention'*. The number of heads employed by the authors in their work was $h = 8$. This is essentially projecting the queries, keys, and values h times with each time, a different learned linear projection being made. These are all essentially done in parallel and hence there is no additional time overhead involved. With different heads, for any input item, its context to the other parts of the input are learned and then all these are concatenated at the end to provide a rich representation for any input item.

$$MultiHead(Q, K, V) = Concat(head_1, ..., head_h)W^O \quad (from\ Vaswaniet\ al.\ (2017)) \tag{2.4}$$

where $head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$.

Decoder The Decoder stack is made of three layers. Here, the Attention layers are masked. That is, during training, the decoder is supplied with the real target sequence, But during Inference, the generation of the next word depends on the previous sequence. While predicting the token at position i, the masked M_H self-attention layer ensures that only the self-attention scores between words at position [1, i,−1] are used. M is set to -inf for masked positions, and 0 otherwise. The exponential in the softmax operation will zero the attention scores for masked tokens.(Gasparetto et al., 2022).

$$MaskedAttention(Q, K, V) = softmax(QK^T + M/\sqrt{d_k}).V \quad (from\ (Gasparettoet\ al.,\ 2022)) \tag{2.5}$$

The output of this layer is sent to the Multi head attention that also receives the output from the encoder. The output from this layer after passing through the feedforward layer is then fed to linear and finally, a Softmax layer to give out the most probable word.

There are few skip connections or residuals present in both encoders and decoders for better results (Wankhade et al., 2022).

Attention-based models have been very effective in the Sentiment analysis problem. To name a few, Yuan et al. (2018) employed Bi-LSTM with attention mechanism for their Multi-domain sentiment classification work, Basiri et al. (2021) proposed Attention-based Bidirectional CNN-RNN Deep model which gave them good results.

Attention-based models have also been used for Aspect based Sentiment analysis problem (Wankhade et al., 2022) which is one of the focuses of our work. In addition to this, we also employ the Self-Attention layer for the task of sequence labeling of Aspect and Sentiment tags for the sentences. We provide further details regarding this in the next chapters.

2.4.5 BERT

Bidirectional Encoder Representations from Transformers or BERT is a multi-layer bidirectional Transformer encoder. Since its first introduction, it has achieved great performances in the field of deep learning (Wankhade et al., 2022) and has been also used for a multitude of NLP tasks. The main advantage of BERT is that it is already pre-trained on BookCorpus (Zhu et al., 2015) consisting of 11038 unpublished books[3] containing 800M words and English Wikipedia

[3] as per: `https://huggingface.co/bert-base-uncased#:~:text=of%20this%20model.-,Training%20data,lists%2C%20tables%20and%20headers%`

consisting of 2500M words. BERT model can be trained in two ways: pre-training and fine-tuning. Fine-tuning offers a great advantage to many NLP tasks since it reduces the training overhead significantly. Due to this, BERT is employed in many works, for Sentiment Analysis with studying the impact of coronavirus in social life (Singh et al., 2021), Aspect-based Sentiment Analysis (Li et al., 2019c), Spam detection (Yaseen et al., 2021), For detection of text-based emotion (Acheampong et al., 2021). A part of our work is adapted from the Aspect-based Sentiment Analysis work mentioned above (Li et al., 2019c) which is further elaborated in the next chapters.

BERT is made of the encoder stacks of transformer and is available primarily in two variants, The BERT base and The Bert large. The BERT base consists of 12 encoders stacked upon one other containing 110 million parameters and The BERT large consists of 24 encoders containing 330 million parameters. Our work makes use of the base version for all the experiments made.

BERT is pre-trained with two objectives:

Masked LM: The model takes a text as input and randomly masks 15% of the words. The masked sentences are then processed by the model which now has to predict the masked words. This allows the model to learn the bidirectional representation of the sentence.

Next Sentence Prediction: The model concatenates two masked sentences as inputs during pretraining. These sentences could either be next to each other in the original text or not. The model then has to predict if the two sentences were following each other or not. This task is meant to allow the model to better learn sentence relationships (Wankhade et al., 2022).

The outcome of these tasks is that the model would now be able to be finetuned to the downstream tasks very easily. Devlin et al. (2018) surmise that there have been very good results obtained in classification by doing just this as the model passes representations obtained by the encoders through a single-layer, feed-forward neural network.

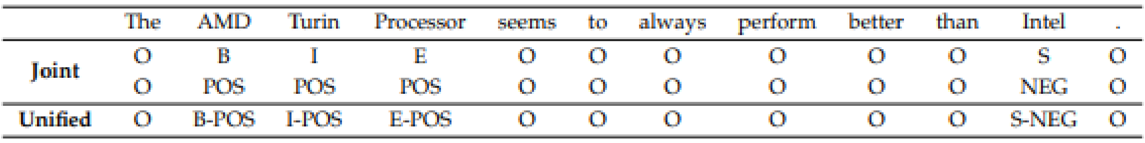
2.5 Aspect-based Sentiment Analysis

Aspect-based Sentiment analysis (ABSA) has been a sub-field of Sentiment Analysis and opinion mining for over a couple of decades (Schouten and Frasincar, 2015; Nazir et al., 2020). In the ABSA problem, the concerned target on which the sentiment is expressed shifts from an entire sentence or document to an entity or a certain aspect of an entity (Zhang et al., 2022). Zhang et al. (2022), in their survey, categorize ABSA broadly as Single ABSA tasks and Compound ABSA tasks.

In Single ABSA tasks, either aspect sentiment classification on a pre-specified aspect term (Jiang et al., 2011) or only the aspect term extraction alone is carried out (Liu et al., 2015).

In Compound ABSA tasks, the aim is to not only extract the aspect terms and the tied sentiment (or in some cases opinion terms) but also to accomplish them together. Some works try to extract the aspect term and its associated opinion terms as a pair (Zhao et al., 2020; Chen et al., 2020). For example, In the sentence, *The portrait is beautiful* , the mentioned works try to extract (*portrait, beautiful*) as an aspect and opinion pairs together. But in our work, we try to tackle the E2E-ABSA task which aims to derive the aspect term and the associated sentiment polarity together. This ties back to the Aspect Mining or AM and the Aspect Sentiment classification or ASC tasks that were mentioned in the section 1.1.

According to Zhang et al. (2022), there are three ways the task has been approached in the literature. The first couple approaches tackle the task as a sequence-labeling work where every word of the sentence is tagged to whether it belongs to the aspect of that sentence or not.



	The	AMD	Turin	Processor	seems	to	always	perform	better	than	Intel	.
Joint	O	B	I	E	O	O	O	O	O	O	S	O
	O	POS	POS	POS	O	O	O	O	O	O	NEG	O
Unified	O	B-POS	I-POS	E-POS	O	O	O	O	O	O	S-NEG	O

Figure 2.6: Demonstration of the joint and unified tagging methods for the E2E-ABSA task, from Zhang et al. (2022)

Usually, this is annotated with a particular labeling scheme (i.e, BIO or BIEOS, to account for multiple aspect words).

In the Joint method, the labels are provided in two rows as shown in Figure 2.6. The first row captures the aspect words and their positions if there are multiple aspect words. The second row denotes the sentiment bound to those words. Here, the task is also tackled in a multi-task learning framework (Liang et al., 2020; Luo et al., 2020; Chen and Qian, 2020b; He et al., 2019; Luo et al., 2019) with two subtasks aiming to extract aspect and sentiment terms and a final prediction being made from the combination of two sub-tasks (Zhang et al., 2022).

There are also a few works (Wang et al., 2018; Li et al., 2019b,c) that approach the task with a Unified (or collapsed) tagging scheme, where the tags for both the aspect and sentiment are provided together. This can also be seen in Figure 2.6 that the tags contain two parts. The first part is made of a BIEOS tagging scheme denoting whether the corresponding words are at the Beginning (B), Inside (I), or at the End of the Aspect (E). If the words do not define the aspect, they are tagged an 'O' and if there is only a single word aspect, then the word is tagged with an 'S'. The second part now contains the sentiment polarity to the said aspect and $\in \{POS, NEU, NEG\}$.

In a novel third way by Hu et al. (2019), the authors approach the task in a sequential manner, they infer that the sequence labeling approach suffers from problems such as huge search space and sentiment inconsistency. The authors propose a span-based extract-then-classify framework, where multiple opinion targets are directly extracted from the sentence under the supervision of target span boundaries, and corresponding polarities are then classified using their span representations (Hu et al., 2019). This approach can be seen as one conducted with a pipeline since the tasks are undertaken sequentially.

With us adopting a part of the work from Li et al. (2019c) to ours, our approach hence tackles the E2E-ABSA problem as a sequence labeling problem that uses a collapsed-labeling scheme format for the training.

2.6 Semi-Supervised Approaches for ABSA

One of the focuses of this work is to evaluate the E2E-ABSA performances across domains. We strive to evaluate how well an E2E-ABSA model trained on one domain can generalize to the others. This cross-domain evaluation is also known as domain adaptation in some literature. In this work, we use these terms interchangeably.

While the notion of the domain remains a vague concept (Van Asch and Daelemans, 2016), in one of the tasks, we try to divide the data into different topics using one of the prominent scientific methods known to do so. While doing so, we try to give it a sense of scientific meaning in how we divide the data. But, this is elaborated in section 5.3.1

As we tried to tackle this task, we realized the data from the other domains did not contain the annotation needed for the E2E-ABSA task. Hence, when multiple domains are involved, and in the setting of E2E-ABSA, we concluded that a Semi-supervised approach may be warranted.

Many of the works in the Semi-supervised approach to the ABSA are from the sole task of AM. Some works (Mukherjee and Liu, 2012; Chen et al., 2013; Li et al., 2019a) use topic modeling with pre-defined domain-specific seed words for the AM task. Few approaches have incorporated varied forms of data augmentation such as masked sequence-to-sequence generation(Li et al., 2020a), soft prototype generation (Chen and Qian, 2020a), progressive self-training (Wang et al., 2021) to generate more pseudo-labeled data for the AM task.

One work aiming for the ASC incorporates the commonsense knowledge into their attentive neural network (Ma et al., 2018).

To the best of our knowledge, Li et al. (2020b) is the only work that proposes an end-to-end Semi-supervised Deep learning framework that can leverage labeled and unlabeled reviews for both the AM and ASC sub-tasks. The work, however, incorporates labeled datasets from the Laptop and Restaurant domain. The unlabeled data is collected from the laptop review of the Amazon review dataset (He and McAuley, 2016) and restaurant reviews from the Yelp review dataset(yelp,2014). Hence, they do not provide cross-domain analysis as the dataset considered for both the labeled and unlabeled sets were from the same domains.

3

Datasets & Evaluation Techniques

In this chapter, we first give an account of the datasets used in the ABSA task. Then we discuss the version of this dataset we employ in our work as part of the labeled dataset. In the next section, we discuss the dataset incorporated as part of the unlabeled data counterpart of our work. We conclude this chapter by presenting techniques to evaluate the predictions made on this unlabeled data.

3.1 E2E-ABSA Datasets

Labeled datasets play a vital role in the development of ABSA methods. For the ABSA task, the annotated Laptop reviews and Restaurant reviews published as part of SemEval challenges are the most widely used datasets. These datasets, namely are from SemEval-2014 (Pontiki et al., 2014), SemEval-2015 (Pontiki et al., 2015), and SemEval-2016 (Pontiki et al., 2016). The datasets contain annotations of aspect categories, aspect terms, and sentiment polarities (although not all of them contain all these annotations),. This lends itself to many ABSA tasks such as aspect mining or aspect sentiment classification(Zhang et al., 2022).

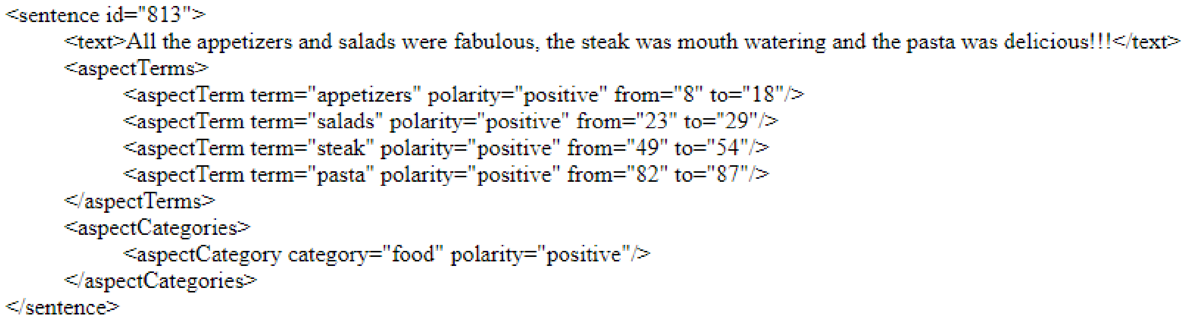
Since the release of the above datasets, there have been modifications and additions done to it by other works. For instance, the original release did not contain the opinion terms. After the work of Fan et al. (2019), these additional annotations were added by Xu et al. (2020).

For our work, we make use of the SemEval dataset repurposed by the work of Li et al. (2019b)[1]. We also conducted all our experiments on both the restaurant and laptop data samples which we respectively prefix with REST and LAPTOP in our work. The samples for the former are accumulated from the respective tasks of the years 2014 to 2016. While, for the latter, we leverage the repurposed set corresponding to Task-4 of SemEval-2014 (Pontiki et al., 2014).

Figure 3.1[2] gives a snapshot of one of the data samples from the original SemEval-2014 Task-4 challenge. The annotation is given in an XML format as can be seen. Along the text XML tag is the individual sentence sample. The aspect terms and aspect categories are given in their respective XML tags. All the aspect terms are mentioned under the aspect term tag. Aspect terms, hence are single or multi-word terms naming particular aspects of the target entity [2]. Along with it are the individual properties such as the term, polarity, and the position of the said aspect term in the sentence. Here, the possible values of the polarity field are: *positive, negative, conflict, and neutral.* There is also a mention of the aspect category with

[1] Also available at https://github.com/lixin4ever/BERT-E2E-ABSA

[2] As seen from https://alt.qcri.org/semeval2014/task4/



```
<sentence id="813">
    <text>All the appetizers and salads were fabulous, the steak was mouth watering and the pasta was delicious!!!</text>
    <aspectTerms>
        <aspectTerm term="appetizers" polarity="positive" from="8" to="18"/>
        <aspectTerm term="salads" polarity="positive" from="23" to="29"/>
        <aspectTerm term="steak" polarity="positive" from="49" to="54"/>
        <aspectTerm term="pasta" polarity="positive" from="82" to="87"/>
    </aspectTerms>
    <aspectCategories>
        <aspectCategory category="food" polarity="positive"/>
    </aspectCategories>
</sentence>
```

Figure 3.1: Illustration of annotation format for the sentences in the REST [2].

an XML tag of its own. As can be seen, for this sample, the category assigned is *food*. The possible values of the category field are fixed, and they are: *food, service, price, ambiance*, and *anecdotes/miscellaneous*. A sentence may be classified into one or more aspect categories based on its overall meaning [2].

For our work, we would be making use of the same data samples modified to suit the task of sequence labeling as shown in the table 3.1. The sequence labeling includes the info corresponding to that under the aspectTerms tag of Figure 3.1. The modified annotations are a consequence of the work by Li et al. (2019b) and we would be adopting it directly to our work. In the given table, the words are tagged with an 'OT' tagging scheme. These are also recognized by the 'IO' scheme in some works (Sang and Veenstra, 1999). As can be seen in the table, the corresponding aspect words that were tagged with positive polarity in the original release are replaced with pairs 'T' and 'POS' together. The other words are tagged an 'O'.

All	the	appetizers	and	salads	were	fabulous	,	the	steak	was	mouth	watering	and	the	pasta	was	delicious	!	!	!
O	O	T-POS	O	T-POS	O	O	O	O	T-POS	O	O	O	O	O	T-POS	O	O	O	O	O

Table 3.1: Sequence labeled counterpart for Figure 3.1, obtained from Li et al. (2019b)

There were 1799 and 2741 data samples as part of labeled training data for REST and LAPTOP, respectively. Additionally, there were 676 and 800 samples for the test data and 200 and 304 samples for the dev set respectively. All these segregations were done by the authors of the work, Li et al. (2019c) and we do not alter it in our experiments.

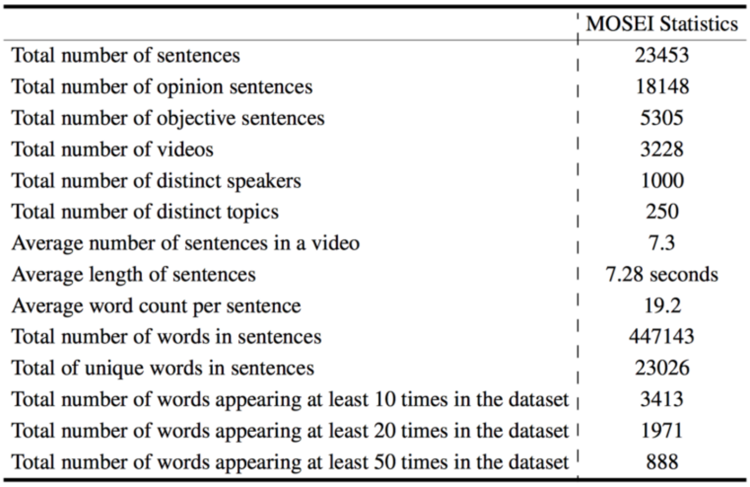
3.2 CMU-MOSEI

For the cross-domain evaluation task of our work, we required a dataset that contained data instances from domains other than the restaurant or laptop. The dataset should also have been at least rooted in the study of Sentiment Analysis so that it would contain the labels needed for the evaluation. Some of the datasets that were considered were the IEMOCAP (Busso et al., 2008), ICT-MMMO (Wöllmer et al., 2013), YouTube (Morency et al., 2011), CMU-MOSI (Zadeh et al., 2016), CMU-MOSEI [3] (Zadeh et al., 2018) and the muse-car (Stappen et al., 2021). These

[3] Available at: `https://github.com/A2Zadeh/CMU-MultimodalDataSDK`

datasets are approximately ordered according to their sizes and all of them are well known in the field of Multimodal Sentiment Analysis, another sub-field of Sentiment Analysis. While we do not venture into modalities other than text in our work, we still were interested in the above datasets since they contained at least sentiment labels. The labels were for the manually annotated transcriptions available for the videos in the respective datasets.

While the ICT-MMMO contained sentiment labels at a video level, the IEMOCAP was not a publically available dataset. The labels for the muse-car were not aligned at the sentence level and at the time of exploration, access to it required special permission from the organizers. Since the labeled set containing ABSA annotations was for the individual sentences, we deemed it necessary that choosing one of the above datasets as the unlabeled set meant that they should contain labels for at least sentiment, but at a sentence level. After carefully examining the remaining choices, we opted for CMU-MOSEI as it contained manual transcriptions of speech along with sentiment and emotion labels at a sentence level. CMU-MOSEI also was larger than CMU-MOSI and YouTube which would benefit us with greater training samples (pseudo-labeled samples).



	MOSEI Statistics
Total number of sentences	23453
Total number of opinion sentences	18148
Total number of objective sentences	5305
Total number of videos	3228
Total number of distinct speakers	1000
Total number of distinct topics	250
Average number of sentences in a video	7.3
Average length of sentences	7.28 seconds
Average word count per sentence	19.2
Total number of words in sentences	447143
Total of unique words in sentences	23026
Total number of words appearing at least 10 times in the dataset	3413
Total number of words appearing at least 20 times in the dataset	1971
Total number of words appearing at least 50 times in the dataset	888

Figure 3.2: Summary of statistics for CMU-MOSEI [4]

Figure 3.2 shows the summary of statistics for CMU-MOSEI [4]. While the figure shows there are 3228 videos to be found, we found that there were 3837 videos available when accessing the dataset in its raw form. However, the manual transcriptions were available to only 3303 videos. Figure 3.3 shows the distribution of sentiment polarities among the sentences found in CMU-MOSEI. Also, Figure 3.4 shows the word cloud for the diversity of topics found, with size denoting the number of videos for the topic.

The CMU-MOSEI comes bundled with the SDK put together by the organizers called the CMU-MultimodalDataSDK. It is a collection of various datasets along with the CMU-MOSEI, accessible through an SDK. While the data (i.e, individual videos and their transcriptions) can also be obtained in the raw form, the labels are obtainable through only the SDK provided.

Annotations for CMU-MOSEI CMU-MOSEI contains annotations for sentiment and emotion. Sentiment annotations are in [-3,3] according to the Likert scale. Effectively, this would mean as follows: [3: highly negative, 2: negative, 1: weakly negative, 0: neutral, +1: weakly positive, +2: positive, +3: highly positive] (Zadeh et al., 2018). While the dataset also contains labels for emotions, based on the Ekman emotions (Ekman et al., 1980) of {happiness, sadness, anger,

[4] As shown in: `http://multicomp.cs.cmu.edu/resources/cmu-mosei-dataset/`

fear, disgust, surprise}, emotion recognition of any form is out of scope for our work. Hence, we only look for sentiment labels provided. Evaluation and analysis carried out for different domains would be evaluated against these sentiment labels. More details on this are elaborated in the next section 3.3.1

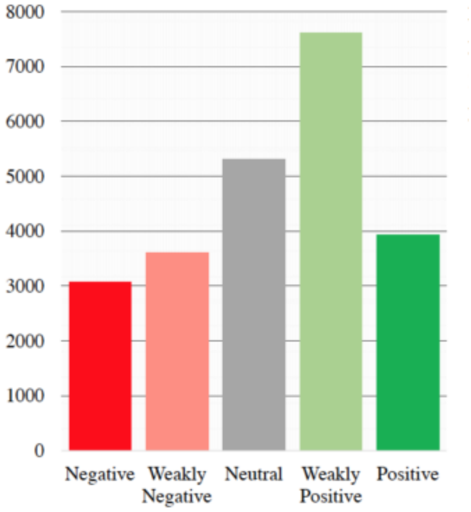


Figure 3.3: Distribution of Sentiment polarities for individual sentences [a]

[a] As shown in: `http://multicomp.cs.cmu.edu/resources/cmu-mosei-dataset/`

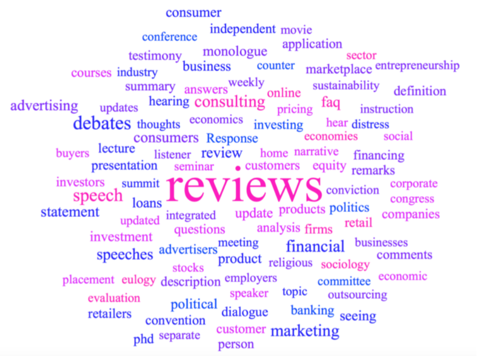


Figure 3.4: Topic Distribution for CMU-MOSEI [a]

[a] As shown in: `http://multicomp.cs.cmu.edu/resources/cmu-mosei-dataset/`

It is critical to mention that the annotations to the work of CMU-MOSEI were made under the influence of modalities other than text alone. The annotators labeled the sentence samples for sentiment and emotion after watching the individual videos which meant that there was additional info such as visual and acoustic cues which were not the settings for either the REST or LAPTOP. However, we continue to use these labels for the evaluation of sentiment in our work but deem this distinction important enough to be mentioned.

3.3 Evaluation Techniques

In this section, we discuss the techniques used for evaluating predictions on the unlabeled data. The techniques discussed here tie into the cross domain analysis task of section 4.3, whose results are provided in the section 5.3.2.

3.3.1 Evaluation on Sentiment

As explained in previous sections, To obtain the labels for individual sentences, we needed to utilize the CMU-MultimodalSDK [3], since the labels for CMU-MOSEI were not available in the raw dataset. CMU-MultimodalSDK is an SDK that allows access to features and metadata of different modalities corresponding to multiple datasets, including the CMU-MOSEI. These features and other metadata such as intervals are available in the form of computational sequences that can be operated only programmatically.

We first obtained the respective computational sequences required for our task. These were the *CMU_MOSEI_TimestampedWords* and the *CMU_MOSEI_Labels*. We then aligned them so that they can be both accessed in conjunction. Here, it is vital to state that this alignment led to a loss of many sentence samples (referred to as TimestamedWords in the SDK). More on this is explained in section 5.3.2

	Sentence	Label
	It's important for teachers to understand their legal, ethical, and professional obligations in an educational setting.	[1.0, 2.3334, 0.0, 0.0, 0.0, 0.0, 0.0]

Table 3.2: A sample sentence and its associated label from CMU-MOSEI. The first value in the array is the sentiment (valence) label and the rest values are for emotion (arousal).

The labels from the CMU-MOSEI contained sentiment and emotion values for the individual sentences, the format of which is shown in the table 3.2

As can be seen, these labels were available as an array of floating-point numbers. The first number corresponded to the sentiment and the rest corresponded to various emotions. In our case, we use the former. The sentiment labels were annotated in the range [-3,3] with -3 being highly negative to +3 being highly positive.

Since the format of these sentiment labels was different from that of REST and LAPTOP, which was in either of $\{POS, NEU, NEG\}$, we used the following logic for comparison and evaluation:

$$\begin{aligned}
&if\ \ predicted\_sentiment = NEG\, and\, label\, in\, [-1,-3]:\\
&\quad correct\_prediction = True\\
&else\ \ if\, predicted\_sentiment = POS\, and\, label\, in\, [1,3]:\\
&\quad correct\_prediction = True\\
&else\ \ if\, predicted\_sentiment = NEU\, and\, label = 0:\\
&\quad correct\_prediction = True\\
&else\ \ correct\_prediction = False
\end{aligned}\tag{3.1}$$

Hence, by evaluating the predictions of the individual sentences against the CMU-MOSEI label in a semi-automatic fashion, as shown above, we record the results. In the next section, we discuss the procedure to evaluate aspect words.

3.3.2 Evaluation on Aspect

Unlike the evaluation for sentiment, where we had the benefit of labels from CMU-MOSEI, the Aspect evaluation could only be done with the inclusion of necessary annotations, these annotations being the aspect words for individual samples of the unlabeled set. For this purpose, we asked three annotators to annotate their opinion of aspect word(s) on a few sentence samples. They were tasked with providing opinions on a randomly sampled 200 sentence-units. While every unit mostly consisted of a single sentence, some units could be broken down into several sub-sentences (separated by commas in CMU-MOSEI transcriptions). The sentences could be shorter in nature (less than 5 words) or could be very long (more than 30 words). In the case of the latter, while it is debatable to consider them as a single sentence, we believe, it could still be considered a single unit for processing, and hence, in this work, we recognize them as such. A couple of such samples are given below:

It's Walt Disney

Frank will co-lead it with Peggy Brown from the Mental Health Commission, there will be representatives from PHNs but, in particular, from throughout the mental health sector- and this will take the form of redoing the guidelines but then overviewing each of the 31 PHNs- doing intensive work on their individual commissioning to make sure, wherever possible, we are building on existing services rather than simply creating or destroying and recreating, and that, I think, is a very important initiative.

For annotation work, we followed the guidelines proposed by the works of SemEval Task-4 (Pontiki et al., 2014) and SemEval Task-12 (Pontiki et al., 2015), as close as possible. While the guidelines[5] [6] give a more detailed account of the sentences that can or cannot be considered to be of containing aspect terms, we give some of the major points here, as follows:

- Aspect terms could be single or multi-word terms.
- Each aspect term identified, has to contain one of the following polarities, based on the sentiment that is expressed in the sentence about it [6]
  - positive
  - negative
  - conflict (both positive and negative sentiment)
  - neutral (neither positive nor negative sentiment)
- Pronouns like we, they, and them cannot be aspect words. The following sentence has no aspect words:

  We all felt it was worth it.

- Sentences denoting objective information cannot be tagged for any aspect words[6]. The following sentence has no aspect terms.

  I went to this restaurant with a woman that I met recently.

- Subjectivity indicators (i.e., words/phrases expressing an opinion, evaluation, etc.) cannot be considered as aspect terms[6]. The following sentence again has no aspect terms.

  The MacBook is way too overpriced for something so simple and chaotic.

In addition to this, it is vital to mention that the original guidelines proposed were for the dataset set in either the restaurant domain or the laptop. This made the detection of aspect words relatively easier, as the target domain was known. But for the unlabeled data of our task, the domains were very diverse. To combat this to a certain degree, we also provided the

[5] As seen in `https://alt.qcri.org/semeval2015/task12/data/uploads/semeval2015_absa_restaurants_annotationguidelines.pdf`

[6] As seen in `https://alt.qcri.org/semeval2014/task4/data/uploads/semeval14_absa_annotationguidelines.pdf`

annotators with the domain info for the individual sentences. This domain info consisted of two things. First, the domain category that the sentence belonged to, and second, the possible topics from each of these domains. The procedure to obtain both of these are explained in the sections 4.3.1 and 5.3.1.

Even with the presence of these guidelines and additional metadata info, we observed that numerous instances contained neither positive nor negative sentiment but were also debatable to be considered neutral. In such cases, we generally observed disagreements in opinion.

Additionally, we also asked the annotators to provide the sentiment info along the aspect words. While this sentiment info is not used for evaluation (as it is already realized with true labels, as explained in the previous section), we use them to infer the distribution of aspect words to these sentiments (more on this in section 5.3.2).

4 Approaches

In this chapter, we first discuss the approach to the E2E-ABSA problem by detailing the model design. We also elaborate on this by explaining the variants of the model and then set them all as our baseline approaches. In the further sections, we set up the Semi-supervised approach to the E2E-ABSA baseline and explain in detail the procedure for both the Self-Training and Tri-Training methods.

In the next section, we give, in detail, the procedure to segregate the unlabeled data into topics that we then consider as domains for our work. We also lay out details for experiments on cross domain analysis by incorporating the above-mentioned Semi-supervised methods.

Finally, in the last section of the chapter, we outline the approach to evaluate the E2E-ABSA in settings of spontaneous and erroneous speeches. For this, we present, in detail, the methods to procure transcripts from an external Automatic Speech Recognizer that enables these speech instances. We conclude the chapter by mentioning the experiments conducted on these transcripts that would stand in as unlabeled data for this task.

4.1 Baseline for E2E-ABSA

With our approach to the E2E-ABSA problem set as a sequence labeling problem, we develop the model in two steps. The first step involves the usage of BERT as an embedding layer to learn the contextual representations of the individual sentences. In all our experiments, we use Huggingface's implementation of the BERT base model, the *bert-base-uncased* [1] to finetune our model. In the next step, the learned representations from the BERT are passed on to another layer for the downstream task of tagging individual words based on a tagging scheme. The downstream task is carried out by a layer called the ABSA layer and is adapted from the work of Li et al. (2019c). There are six various implementations for this layer from the original work namely, the linear layer, Gated Recurrent Unit (GRU), Long short-term memory (LSTM), Self-Attention networks (SAN), and transformer (TFM), and Conditional Random Fields (CRF). When we conducted the experiments with these layers, we did not obtain good results with both the LSTM and CRF, hence we would be omitting them in all our experiments and proceed with the others.

[1] https://huggingface.co/bert-base-uncased

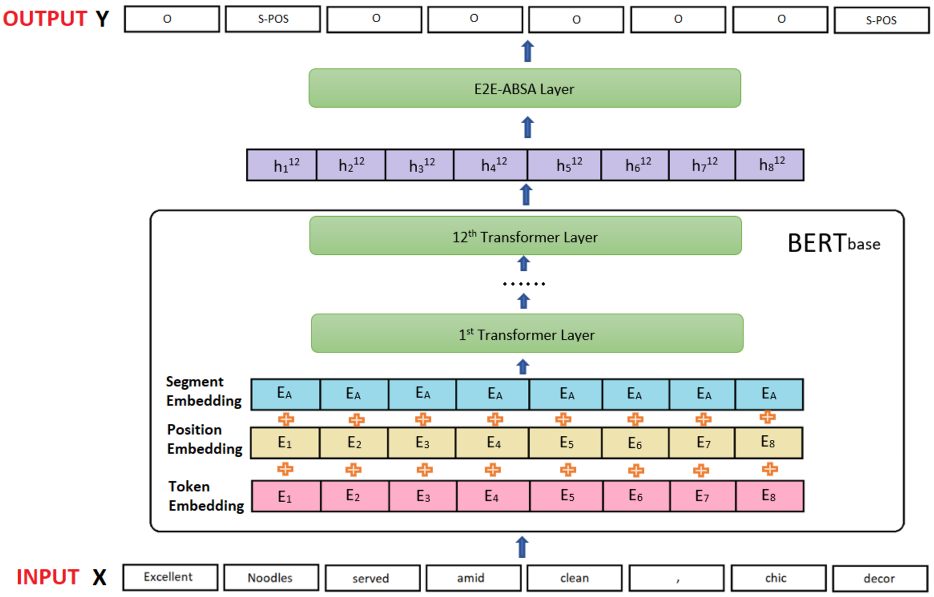


Figure 4.1: Model Architecture

Formally, if we define input sequence X to be $X = \{x_1, x_2, ..., x_t\}$ with t being the length of the sequence, Then after passing through the BERT with L layers, we get the corresponding contextualized representations as $H^L = h_1^L, h_2^L, ...h_t^L \in \mathrm{R}^{t*dim_h}$ with dim_h denoting the dimension of the representation vector (Li et al., 2019c). The individual operations at the layer performing the downstream tasks i.e., the E2E-ABSA layer can be regarded as follows:

4.1.1 Linear layer

The obtained contextualized representations can be directly passed onto a linear layer with softmax activations, with the number of labels being |S| where S = {'O',' E', 'S-POS', 'B-POS', 'I-POS', 'E-POS', 'S-NEG', 'B-NEG', 'I-NEG', 'E-NEG', 'S-NEU', 'B-NEU', 'I-NEU', 'E-NEU'}. This can be formally given as:

$$P(y_t|x_t) = softmax(W_o h_t^L + b_o) \qquad (from\ Liet\ al.\ (2019c)) \tag{4.1}$$

where $W_o \in \mathrm{R}^{dim_h \times |Y|}$ is the learnable parameters for the linear layer.

4.1.2 Gated Recurrent Unit

Gated Recurrent Unit or GRU, belonging to the family of Recurrent Neural networks are also utilized for the downstream task. The layer with GRU implementation arrives at the calculated value of h_t as follows:

$$\begin{aligned} h_t^{`} &= tanh(W_h r_t h_{t-1} + b_h) + tanh(W_h X_t + b_h) \\ u_t &= \sigma(W_u h_{t-1} + b_u) + \sigma(W_u X_t + b_u) \\ r_t &= \sigma(W_r h_{t-1} + b_r) + \sigma(W_r X_t + b_r) \\ h_t &= u_t h_t^{`} + (1 - u_t) h_t - 1 \end{aligned} \tag{4.2}$$

where σ is the sigmoid activation function and $W_h \in \mathrm{R}^{2dim_h \times dim_h}$ and $W_u\, and\, W_r \in \mathrm{R}^{dim_h \times dim_h}$ are learnable parameters. The $u_t\, and\, r_t$ are update and reset gates respectively. The h_t is then passed onto the softmax layer for predictions as follows:

$$P(y_t|h_t) = softmax(W_o h_t^L + b_o) \qquad (from\ Liet\ al.\ (2019c)) \tag{4.3}$$

4.1.3 Self-Attention Networks

The authors (Li et al., 2019c) describe Self-Attention networks or SAN as to be one containing a self-attention layer(Vaswani et al., 2017) and a residual connection(He et al., 2016), this can be formalized as follows:

$$h_t = LN(H^L + SLF\_ATT(Q, K, V)) \qquad (from\ Liet\ al.\ (2019c)) \tag{4.4}$$

where Q, K, V = $H^L W^Q, H^L W^K, H^L W^V$, and LN is layer normalization added on top of the Self-attention(Li et al., 2019c).

The h_t is then passed onto the softmax layer for predictions as shown in equation 4.3

4.1.4 Transformer Layer

Another variant that has the same architecture as that of the single transformer encoder in the BERT is used for the downstream task (Li et al., 2019c). The operations can be formalized as below:

$$\begin{aligned} h'^L &= LN(H^L + SLF\_ATT(Q, K, V)) \\ h'_t &= LN(h'^L + FFN(h'^L)) \qquad (from\ Liet\ al.\ (2019c)) \end{aligned} \tag{4.5}$$

where FFN refers to the point-wise feed-forward networks and LN is the layer normalization(Vaswani et al., 2017)

The h'_t is then passed onto the softmax layer for predictions as shown in 4.3

With the individual layers defined, the whole architecture can be summed up in Figure 4.1.

For the baseline runs, we consider the model depicted in 4.1. The model with each of the four different ABSA layers defined above is considered a different implementation. Hence, for the baseline experiments, we have four implementations altogether.

With the BERT base model containing the number of transformer layers $L = 12$ and hidden size dim_h as 768, we train the model with the REST and LAPTOP. The REST and LAPTOP contain 1799 and 2741 training samples, respectively. Here, the E2E ABSA layer would be composed of either of the layers defined above and a run is made for each of them. So, in all, there are four implementations with REST and another four with LAPTOP.

The other parameters for the model are as follows: The whole training is conducted up to a maximum of 3000 steps with batch size set to 16 for REST and 24 for LAPTOP. The learning rate is set to 2e-5. For every 100 steps of the training, the model is saved so that it can be further accessed for the evaluation against the dev set and the test set. Model selection is done on the dev set, considering Macro-F1 score as the primary metric. The experiment results would be discussed in the chapter 5.1

4.2 E2E-ABSA with Semi-Supervised Approaches

With the baseline models defined in the last section, we leverage the two Semi-supervised methods defined in section 2.3. First, we describe the E2E-ABSA with the Self-Training method. We lay out the procedure for E2E-ABSA with the Tri-Training method in the next sub-section.

4.2.1 E2E ABSA with Self-Training

With the general outline for the algorithm given in section 2.3.1, we now give further details regarding the runs made leveraging this algorithm. The details are presented considering the training set as REST, But the procedure is analogous to LAPTOP.

The models initially trained on the 1799 samples are saved at every 100 steps (i.e., checkpoints) of training and then loaded for evaluation against the dev set and the test set, as said before. With the Macro-F1 being the primary metric of evaluation, we recognize the best checkpoint when evaluated against the dev set. The checkpoint at which the highest Macro-F1 value would be obtained would then be considered as our base model for the next set of training.

As part of the unlabeled data, we compiled 19961 sentences from 2467 documents (or video transcriptions) from CMU-MOSEI. The base model chosen before is made to predict these sentences for sequence labeling of Aspect and Sentiment (i.e, E2E-ABSA tags) at the end of every iteration. Now, Suppose an 'X' quantity of samples from a 19961 set would be tagged for aspect by our model, then, these 'X' samples would then be considered as pseudo-labeled samples. Now, these pseudo-labeled samples are then added to the original labeled set and used to re-train the model for the next iteration. Other hyperparameters remain intact for the next iteration.

Effectively, for the next iteration, the number of unlabeled samples would be reduced by X. The new model, now trained on data containing both the original labeled samples and the pseudo-labeled samples (from the last iteration) is tasked with predicting the remaining unlabeled samples. This process is iteratively continued for up to either 20 iterations of the run or at an iteration where less than 30 tagged samples are produced.

This is carried out with all the four variants of ABSA layers, employed one at a time. We consider each of these four runs as an implementation. These four implementations are then repeated with LAPTOP as training data. We give the evaluated results in the section 5.2.1.

4.2.2 E2E ABSA with Tri-Training

Similar to the previous section, we present the procedure for Tri-Training implementations considering REST as the training data, But the overall procedure remains the same for LAPTOP as well.

The framework for Tri-training, as given in 2.3.2 will be made use for our work. For every implementation, there are simultaneously three models in play. The three models are initially trained on the labeled samples i.e., REST. While the original work by Zhou and Li (2005) proposes bootstrap sampling of the original labeled data , we initially got poorer results while measuring for error (explained further) using this sampled data approach. The number of labeled samples being modestly low in totality could be a possible reason for this. It is for this reason that we train all three models with the entirety of available labeled samples. In addition to this, For measuring error, the models predict E2E-ABSA tags for samples from the test set of the REST. In preparation of pseudo-labeled samples, the models also predict E2E-ABSA tags for samples from the unlabeled set before concluding the first phase of the implementation. As part of the unlabeled data, we leverage the 19961 samples of CMU-MOSEI.

For the next phase, the update phase, since the agenda is to moderate the addition of pseudo-labeled samples to the next iteration of training of a model, we calculate the error in measurement for it first. This error measurement is carried out against the test set of the original labeled data i.e., REST. Here for any model h_i $with$ $i \in \{1, .., 3\}$, the idea is to estimate the classification error rate of the hypothesis derived from the combination of the other two models, say, h_j and h_k (Zhou and Li, 2005). This is done by dividing the number of labeled examples on which both h_j and h_k make incorrect classification with the number of labeled examples on which the classification made by h_j is equal to that of h_k. Iteratively, this is done for all three models. For any model h_i if the classification error is more than 0.6 (as opposed to the proposed 0.5 in the original work), then it is not eligible for an update. An update involves re-training the model with the addition of pseudo-labeled samples, as part of the next iteration. We changed the comparison for error from 0.5 to 0.6 to facilitate the update since the error rate for all our model were just above 0.5. Moreover, we did not find any significance for the value of 0.5 in the original work and hence assume that it is open to re-interpretation.

After this, For re-training any h_i, we consider those samples that are identically predicted by h_j and h_k. While all these samples are available for re-training this model, we ultimately sub-sample only a portion of them. The idea is to moderate the inclusion of these pseudo-labeled samples so that there is no dramatic increase in classification error. This holds for any iteration. This is also summarized with the following relation :

$$L[t] * e[t] < L[t-1] * e[t-1] \tag{4.6}$$

where L denotes the number of samples for training and e denotes the classification error at iterations t and $t-1$.

The sub-sampling itself is carried out by a factor. This factor is given as $|L_t| - s$. The s, however, can be calculated by a factor:

$$\lceil \frac{e^{t-1}[L^{t-1}]}{e} - 1 \rceil \tag{4.7}$$

We do this to all the three models in operation and we keep track of pseudo-labeled samples for each of them.

We finally update the models with their respective pseudo-labeled samples added to the original ones and update the parameters such as the error and sample terms for the next iteration.

This whole process is a single implementation and is carried out up to an iteration that results in an error higher than that of the previous (for any of the three models), in which case the implementation terminates.

The same proceedings are carried out for the LAPTOP. The results for all these implementations are presented in the section 5.2.2

4.3 ABSA across Domains using Semi-Supervised Approaches

In their work of predicting the effectiveness of Self-Training, Van Asch and Daelemans (2016) state that the domain, in literature, is inherently a vague concept. They opine that, in Machine Learning, it is paramount that any domain should differ from the other by the distribution of words. One of the prevalent ways to identify domains is to measure a sense of similarity in-between the corpora thereby creating a sense of boundary. This concept of similarity is applied to many use cases such as feature selection (Della Pietra et al., 1997), training corpus creation (Chen et al., 2009), etc . Van Asch and Daelemans (2016) state that the current research focuses on semantic textual similarity (STS) (Agirre et al., 2013). One such method that uses the STS is Latent Dirichlet allocation. We detail this in the further sub-section.

4.3.1 Domain Classification using Topic Modeling

The CMU-MOSEI is distributed across various topics, as seen in Figure 3.4. While the original work did not contain metadata concerning the domain(s) that each video could belong, for our work, in order to proceed forward with evaluating E2E-ABSA across domains, these needed to be classified. Hence for this task, we used the manual transcriptions available with the CMU-MOSEI obtained in its raw form. With each of these transcriptions set as a document, we used Latent Dirichlet Allocation (LDA) (Blei et al., 2003) to perform this unsupervised classification. LDA, a topic modeling approach, can be seen formally as a multinomial distribution over the several words in the corpus and represents documents as a mixture of topics. To start with, we put together all sentences together from all the individual transcribed files which we consider as a single document. 2467 such documents were considered for the topic modeling task.

Additional preprocess for the individual documents involved removing stopwords words lesser than 3 characters. To identify and recognize successive words that made sense as a pair(than alone), we then ran the words through the Bigram phraser to leverage the existence of prevalent bigrams in the documents if any. We then remove tokens that appeared in less than 15 documents and more than 0.5 documents. We finally only keep the first 100000 most frequent tokens.

We performed the training for the LDA model on two sets of corpora created from the documents. First converts the documents into the bag of words. In the second, we make use of gensim's TF-IDF model [2] and apply it to the bag of words corpora to create the individual vectors.

For the topic modeling itself, we make use of gensim's LDAMulticore implementation [3] of LDA and run it against both the bag of words corpus and TF-IDF vectors. To aim for a better coherence value, we ran the model against the number of topics being set to between 4 and 20. Other hyperparameters such as alpha and eta were varied by a factor of 0.10 in the range of 0.01 to 1. Other values for alpha included the 'symmetric' and 'asymmetric' and for eta, the value 'symmetric'. These were interchanged one after the other in each run and for every run, we also simultaneously check for the coherence score c_v. To get the coherence score, again, we make use of gensim's CoherenceModel [4] implementation.

Among the results we had, we considered values for hyperparameters that resulted in a better coherence value and began distributing documents into different domains. While distributing, we also regarded the distribution of documents to be a considerable factor and eliminated those that resulted in poor distribution. Here, the distribution where the bulk of the documents was assigned to only a few topics with other topics distributed scantly was ignored.

We present the results for topic modeling in 5.3.1. For all the experiments in this section including the further sub-sections, we consider the topics obtained from this method as our domains.

4.3.2 E2E ABSA with Self-Training

The crux of this task would be similar to the section 4.2.1. We train the model initially with REST consisting of 1799 samples for 3000 training steps to complete an iteration. At the end of an iteration, the model is tasked with predicting the unlabeled samples intended to act as pseudo-labeled samples for further iterations.

In addition to this, the model is tasked with also predicting another unlabeled set not intended to act as pseudo-labeled samples but for evaluation against the individual domains. This set consisted of 8250 samples drawn from 740 documents distributed to 5 different domains as

[2] https://radimrehurek.com/gensim/models/tfidfmodel.html

[3] https://radimrehurek.com/gensim/models/ldamulticore.html

[4] https://radimrehurek.com/gensim/models/coherencemodel.html

explained in section 4.3.1. The number of sentence samples for each domain can be summarized as follows:

- Domain-1: 1738 sentence samples
- Domain-2: 1704 sentence samples
- Domain-3: 2506 sentence samples
- Domain-4: 1105 sentence samples
- Domain-5: 1147 sentence samples

The predictions for these samples are tasked to the model at the end of every iteration and we then record the results.

The other settings would remain the same as done in 4.2.1. The evaluation of these samples is carried out in two different steps. For sentiment evaluation, we use the sentiment labels available from CMU-MOSEI. For evaluating aspect terms, since we do not have any aspect labels in CMU-MOSEI, we evaluate them against aspect labels provided by three annotators. More details regarding this are elaborated in the sections 3.3.1 and 3.3.2. The results for them are discussed in the section 5.3.2

4.3.3 E2E ABSA with Tri-Training

The approach is similar to the previous section. We leverage 8250 sentence samples distributed among five domains for evaluation. The main difference here is that at every iteration, for each model involved in a single implementation, we predict samples consisting of data for different domains. These samples are unseen by any model at any iteration of training and are only used for the evaluation of sentiment and aspect terms for different domains. We keep the other settings intact as described in section 4.2.2. For evaluation, we follow the procedure discussed in sections 3.3.1 and 3.3.2. The results are discussed in section 5.3.2.

4.4 E2E-ABSA across Erroneous Transcriptions

The final task for our work concerns the E2E-ASBA under the influence of faulty transcriptions containing erroneous sentences and spontaneous speech instances. First, we explain the reasons for this task. We then give the error rates of a few selected samples to demonstrate the differences between manual transcriptions and that of an external Automatic Speech Recognition Systems (ASRs). We then give an account of how the Transcriptions from an ASR were prepared. We end this section by detailing the procedure of modeling E2E-ABSA with these transcriptions.

The authors of the dataset Muse-CAR, Stappen et al. (2021) obtained captions for their dataset using the ASRs. To the best of our knowledge, it is the only dataset that obtains the captions from an external system rather than manually annotating them. The authors surmise that there is a considerable error rate while transcribing through an external service. Again, this would impact the predicted sentiment and provide a rich ground for analyses. This forms the first reason for the evaluation. Table 4.1 provides a few examples showing the differences between the manual transcriptions versus the same samples transcribed with the Amazon Transcribe.

Also, the CMU-MOSEI is a collection of videos (which are annotated for sentiment in mostly sentence levels) that contains essentially a single speaker giving his views on a specific topic. Some of these are carried out unscripted and hence involve various instances of repetition, hesitation, etc. Hence, this provides an opportunity to investigate sentiment detection under the context of spontaneous speeches. This is the second reason behind the task.

While considering various ASRs, we initially trialed a few video files on both Google's Speech-to-Text [5] and Amazon Web Service's (AWS) Transcribe [6]. While on an initial inspection, the transcription from both the services seemed adequate, the transcriptions from AWS Transcribe contained the additional timestamps info, which we deemed very useful for the task at hand. Hence, for this task, we employed AWS's Transcribe service to transcribe a fifth of our original unlabeled set and conduct experiments and evaluations on various settings. More details regarding this along with the results are presented in the section 5.4.

We explain the preparation of data in the next section and the further section explains the approach to the task.

	From Manual Annotation	**From AWS Transcribe**	**Word-Error Rate (WER)**
	And now an interesting question to ask is how do people actually set prices in real life? Well, why don't look around and ask? If you ask your, let's say, restaurant uncle, he would probably say "I take the cost something on top so that I will earn some money" and this is what we call markup pricing	*And now an interesting question to ask is how do people actually set prices in real life?Well why not walk around and ask if you ask your restaurant uncle? He'll probably say I take the cost and I add something on top so that some money and this is what we call mark up pricing.*	20.69%
	The second issue that Obama brings up and I think one that needs to be looked at very, very carefully is he talks about creating a new lending fund that is going to ensure that more money will make its way to households to buy automobiles, to fund college education and also to entrepreneurs.	*the second issue that Obama brings up and I think one that needs to be looked at very, very carefully is he talks about creating a new lending fund*	46.29%
	Numerous Commercial lenders look at your personal net worth because they will only lend you the requested amount that is either equal to or greater than you Personal Net Net Worth.	*commercial lenders. Look at your personal network because they will only lend you their requested amount that is either equal to or greater than your personal network.*	25.81%

[5] https://cloud.google.com/speech-to-text/

[6] https://aws.amazon.com/transcribe/

	From Manual Annotation	From AWS Transcribe	Word-Error Rate (WER)
	So we've been having a lot of discussion across companies of really I think exciting stuff where we may be able to tie companies' efforts together that will have a big effect on the environment as well as stanch some of the financial problems.	*And so we've had a lot of discussions across companies really. We may be able to tie company's efforts together that will I think exciting stuff. have a big effect on the environment as well as uh,to tackle some of the financial problems.*	38.64%
	A couple weeks back a couple of journalists posted that they got these invitations from Activision for what was called the Soundial festival.They had all these band names on it but none of the band names seemed familiar to anybody.	*a couple of weeks back. A bunch of journalists posted online that they got these invitations from Activision for what was called The Sound I'll Festival. And they had all these band names on it, but none of the band names seem familiar to anybody.*	17.07%

Table 4.1: Sample sentences transcribed manually vs via AWS Transcribe, with Word-Error Rate

4.4.1 Preparing Transcriptions from AWS Transcribe

To Transcribe the video files from CMU-MOSEI, we made use of two services from AWS. One is the AWS Transcribe itself, the other being the S3. All the individual files were initially uploaded onto the S3 buckets which are essentially containers for any Multimedia items. Then using the S3-URL for each item, one could make use of any service in the AWS eco-system that involves the usage of the said Multimedia items of any form.

While initially, we understood that a form of Batch-processing could be present to streamline the process of pulling individual files from S3 buckets and onto the AWS Transcribe, any efforts made were not useful since the S3-URLs were unique for an item. Moreover, the Transcribe process lists only the recent 100 transcription jobs, and any jobs before that became inaccessible. This meant that these jobs would have to be done manually one by one. It became very clear that given the time constraints, we would not be able to transcribe the entire set of 2467 video files.

With this new development, we aimed to reach a fifth of the whole dataset, and accordingly, 504 videos were transcribed. These 504 video files were selected randomly and care was taken to keep them topically as diverse as the unlabeled data set used in sections 4.2 and 4.3. After this, we move on to the next stage of preprocessing.

While obtaining the individual transcriptions, we noticed there were many instances of timestamp mismatch when compared against manually transcripted counterparts of the videos. With the manual transcripted files from the CMU-MOSEI being the gold standard, to perform any kind of evaluation down the line, we would need to then align these instances first. For this, we incorporated a semi-automatic process involving three stages. In the first stage, we programmatically tried to align instances that found a timespan match with the gold standard instances, In the second stage, we considered the starting word and ending word of each sentence, and barring any repeated occurrence of the end word, we put together all the instances in between to be a part of a single sentence. In the last stage, we manually inspected the individual file output after these two stages and aligned the remaining sentences manually.

After this, as part of the last step, we prepared two sets of unlabeled samples, the first one to be used as pseudo-labeled samples for further iterations of training for the model and the second one used for evaluation. The first set comprised 4103 sentence samples and the second one, 1825 samples.

4.4.2 Semi-Supervised Modeling for E2E-ABSA

After processing transcriptions from AWS, we also create another set of manual transcriptions for the AWS transcribed counterpart, which means that both set contain the same sentences but each from a different source. We do this for two main reasons. First, to account for the now modified training samples from the runs made as in sections 4.2.1 and 4.2.2. Second, to compare the recorded E2E-ABSA performances from the erroneous sentences with the gold standard ones.

We then utilize the Self-Training and Tri-Training methods discussed before. There would be two runs made for a single implementation. To give an example, for Self-Training, we train the model for multiple iterations until it terminates, as explained before. But now, it is run once with the AWS transcribed sentences and once with the gold standard ones, as unlabeled data. But similar to the earlier implementations, we record the model's performance against the test set of labeled samples for every 100 training steps. Additionally, we also make the model predict on another set of 1825 samples at the end of every iteration. This would be used for Sentiment evaluation. Each run of a single implementation is continued for up to 3000 training steps. The same procedure is followed for Tri-Training implementations as well. We make separate trials with both the REST and LAPTOP being the labeled data for training. The results are presented in section 5.4.

5

Experiments, Results and Analysis

In this section, we lay out further settings for the above approaches, we also present the results for the same. In the first section, we discuss the experiments for the baseline models along with their results. In the next section, we present the corresponding results with the two Semi-supervised approaches. We also give out comparisons for the result. We then discuss the E2E-ABSA results for different domains and give out domain-level results. Finally, we lay out the E2E-ABSA results across erroneous transcriptions and in some specific settings.

5.1 Experiments for Baseline Model

For the baseline experiments, we train the model, once with a different ABSA layer against both the REST and LAPTOP consisting of 1799 and 2741 samples respectively. The model is trained for 3000 training steps for each implementation and is saved for evaluation after every 100 steps. After 1000 steps of training, for every 100 training steps, the model is evaluated against both the dev set and the test set and the performance metrics are recorded.

For training, we set the batch size to 16 for REST and 24 for LAPTOP. The learning rate is set to 2e-5. We also use the adam optimizer which is set to 1e-8.

In Tables 5.1 and 5.2, we lay out the results for baseline experiments. The evaluation is done on the test set and we present the respective values for the checkpoint that results in the best Macro-F1 score since Macro-F1 is the primary metric considered.

	Models	**Macro-f1**	**Micro-f1**	**Precision**	**Recall**
	BERT + Linear	0.4459	0.6476	0.6556	0.6398
	BERT + GRU	0.4181	0.6612	0.6650	0.6576
	BERT + SAN	0.5711	0.7158	0.7147	0.7170
	BERT + TFM	0.3989	0.6323	0.6535	0.6125

Table 5.1: Baseline results on the Test set of REST

	Models	Macro-f1	Micro-f1	Precision	Recall
	BERT + Linear	0.5672	0.6105	0.6273	0.5946
	BERT + GRU	0.5524	0.6107	0.6244	0.5978
	BERT + SAN	0.5637	0.6189	0.6244	0.6136
	BERT + TFM	0.5541	0.6100	0.6262	0.5946

Table 5.2: Baseline results on the Test set of LAPTOP

5.2 Experiments with Semi-Supervised Approaches

In this section, we look into the runs made with the two Semi-supervised methods. In the first sub-section, we discuss the experimental settings for the Self-Training method. We then discuss the results following the method. In the next sub-section, we then present the experimental settings for the Tri-Training method along with the results.

5.2.1 E2E-ABSA with Self-Training

Similar to the baseline setting, in the Self-Training approach, we train the model for 3000 training steps for each iteration. We use the set containing 19961 sentence samples as the unlabeled data.

As explained in the section 4.2.1, with Self-Training, the model predicts the unlabeled samples at every iteration, starting with the complete unlabeled set at the zeroth iteration. A fraction of these predictions that contains the E2E-ABSA tag would then be added as pseudo-labeled samples, to the next iteration of training. Also, at this re-training stage, the model that yields the best Macro-F1 score at the previous iteration is selected. This model is then re-trained with the addition of these pseudo-labeled samples. This procedure is carried out to a maximum of 20 iterations or at an iteration that produces less than 30 pseudo-labeled samples. Also, at the end of every iteration, just before re-training, the model is tested against the test set of labeled data i.e, the test set of REST and LAPTOP (separate runs), to record the performance. We use the scores obtained from these evaluations as a basis to measure the effectiveness of Self-Training methods for the E2E-ABSA task.

Figures 5.1 and 5.2 give a snapshot of the performances of various models through the iterations. Each recorded value is against the test set and is the highest Macro-F1 score for that iteration, considering all the 30 checkpoints of training that make up an iteration. For REST, shown in figure 5.1, the performances were at their best at initial iterations and there is a clear decline after the first iteration. There is another decline after iterations three or four and the performances plateau after the eighth. But the performances with LAPTOP, shown in figure 5.2 were mostly uniform throughout iterations. There was still some deterioration in value with some model runs (with BERT-GRU and BERT-Linear), but the downturn was not very steep. Some of the best performances were from the later iterations (especially with BERT-SAN and BERT-TFM).

Tables 5.3 and 5.4 compare the results between the baseline models and that of Self-Training. For the results with Self-Training, we recognize the checkpoint that yields the maximum Macro-F1 score, considering all the iterations of an implementation. We also present the iteration at which the values were recorded in a separate column.

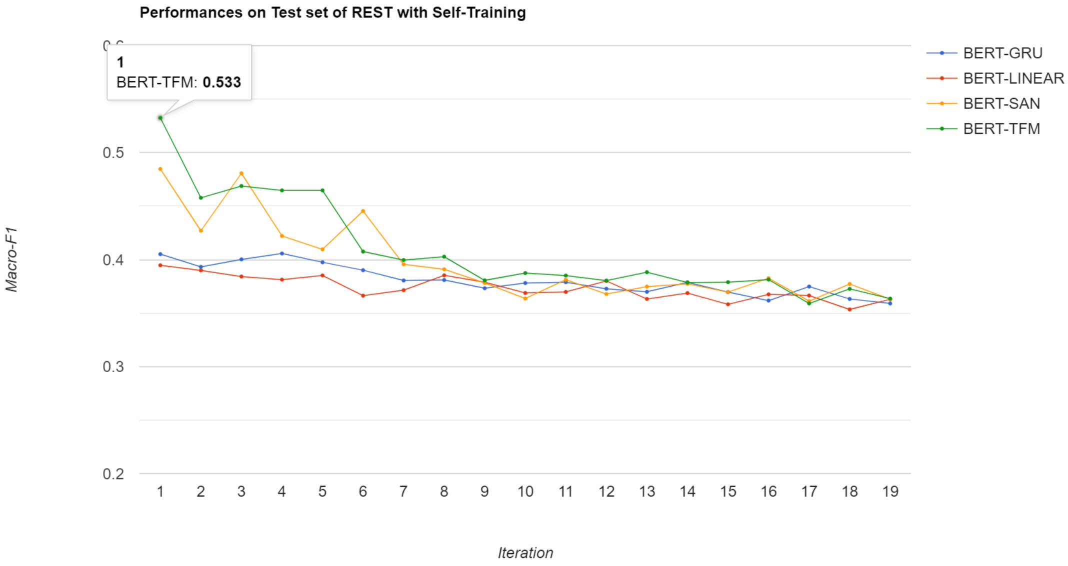


Figure 5.1: Model Performance with Self-Training across all iterations, with REST

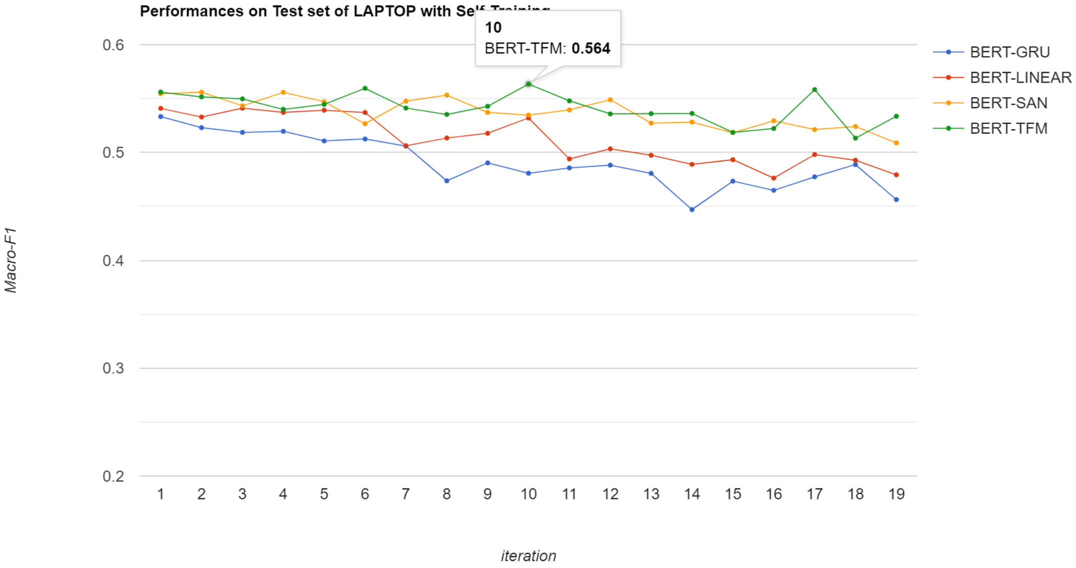


Figure 5.2: Model Performance with Self-Training across all iterations, with LAPTOP

	Models	Iteration	Macro-f1	Micro-f1	Precision	Recall
Baseline	BERT + Linear	–	**0.4459**	**0.6476**	0.6556	**0.6398**
	BERT + GRU	–	**0.4181**	**0.6612**	**0.6650**	**0.6576**
	BERT + SAN	–	**0.5711**	**0.7158**	0.7147	**0.7170**
	BERT + TFM	–	0.3989	0.6323	0.6535	0.6125
Self-Training	BERT + Linear	1	0.3947	0.6361	**0.6598**	0.6141
	BERT + GRU	4	0.4057	0.6469	0.6443	0.6495
	BERT + SAN	1	0.4847	0.6757	**0.7264**	0.6318
	BERT + TFM	1	**0.5324**	**0.6974**	**0.6908**	**0.7041**

Table 5.3: Comparison between the Baseline results with the Self-Training counterparts, with REST.

	Models	Iteration	Macro-f1	Micro-f1	Precision	Recall
Baseline	BERT + Linear	–	**0.5672**	**0.6105**	**0.6273**	**0.5946**
	BERT + GRU	–	**0.5524**	**0.6107**	**0.6244**	**0.5978**
	BERT + SAN	–	**0.5637**	**0.6189**	0.6244	**0.6136**
	BERT + TFM	–	0.5541	**0.6100**	0.6262	**0.5946**
Self-Training	BERT + Linear	3	0.5411	0.5893	0.6162	0.5647
	BERT + GRU	1	0.5333	0.5845	0.5970	0.5726
	BERT + SAN	2	0.5559	0.6040	**0.6261**	0.5836
	BERT + TFM	10	**0.5635**	0.6094	**0.6267**	0.5931

Table 5.4: Comparison between the Baseline results with the Self-Training counterparts, with LAPTOP.

As can be seen from the tables, only the BERT+TFM runs improved the results over baseline. This holds to both REST and LAPTOP. But there was a decline in performance with all the other models.

5.2.2 E2E-ABSA with Tri-Training

Under this method, we conducted four different runs with both the REST and LAPTOP, each with a combination of three models of the available four. We call each of these runs a single implementation. Hence, there were four different implementations to both REST and LAPTOP. Each of these implementations contained two phases, the priming phase where we initialize all three models and use them to predict the whole unlabeled set, and the update phase where we prepare pseudo-labeled samples for re-training and if the individual models are eligible for it, we perform the training.

As part of the unlabeled data set, we again leveraged the 19961 samples curated from 2467 manually transcribed documents. Training (or re-training) is conducted for 3000 training steps, to keep the comparisons consistent.

In the update phase, we initially prepare an appropriate number of pseudo-labeled samples for each model. After the initialized model is re-trained with the addition of these samples, we then test the models with the test set of the original labeled data to record the performance of the updated model. The checkpoint that yields the best Macro-F1 score would then be chosen for error measurement and for predicting the unlabeled set again. It is vital to state two things. First, The error measurement is done to check whether the updated model's classification error is better than its previous iteration. And if it is, it is allowed to continue execution. If not, the whole implementation comes to an end. The method to calculate this is detailed in section 4.2.2. Here, it is important to state that the error for all three models should have to be lesser than its error from the previous iteration, otherwise there is no update possible since, for any model, it's error calculation itself is dependent on the other two models. Hence, the termination of a single implementation is dependent on the error rate of all three models. Second, while the whole unlabeled set is employed for prediction, we ultimately sample a few of them, to keep the error rate as low as possible.

For consistency, we keep the other hyper parameters the same as in previous experiments.

Figure 5.3 to 5.10 gives the snapshot of all the implementations. Figure 5.3 to 5.6 gives the outlook for runs with REST. Figure 5.7 to 5.10 gives the outlook for runs with LAPTOP. For REST, all implementations terminated at the end of the second iteration, as can be seen. However, for LAPTOP, every implementation terminated just after the first iteration. Because of this, it allows us to have a closer look at the model performance at every 100 steps of training.

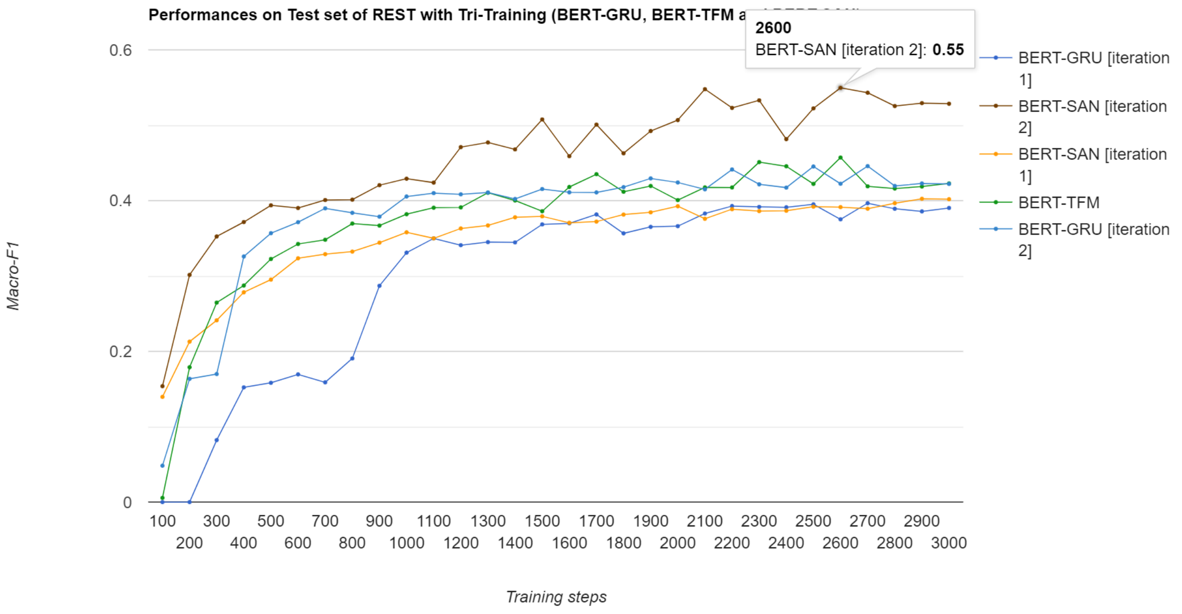


Figure 5.3: Model Performances with Tri-Training across different training steps, with BERT-GRU, BERT-TFM, and BERT-SAN and with REST

From the figures 5.3 to 5.6, it is clear that unlike the runs with Self-Training, the second iteration of every model led to an overall improvement over the first iteration. This conforms to the key idea of Tri-Training that there is a controlled addition of pseudo-labeled samples for re-training.

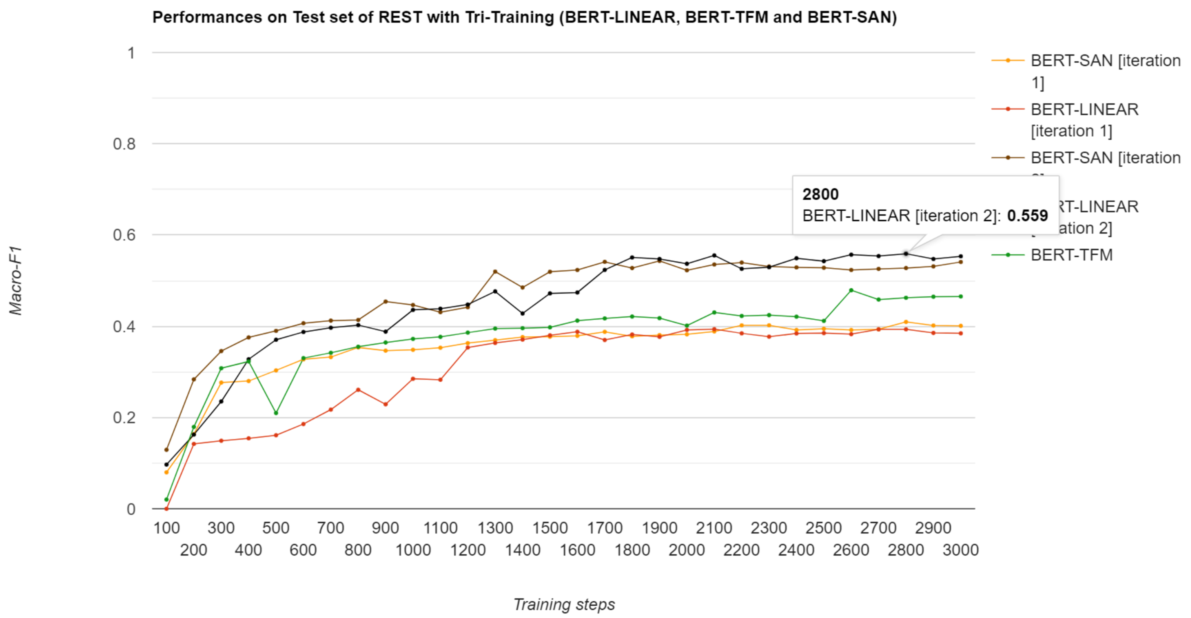


Figure 5.4: Model Performances with Tri-Training across different training steps, with BERT-Linear, BERT-TFM, and BERT-SAN and with REST

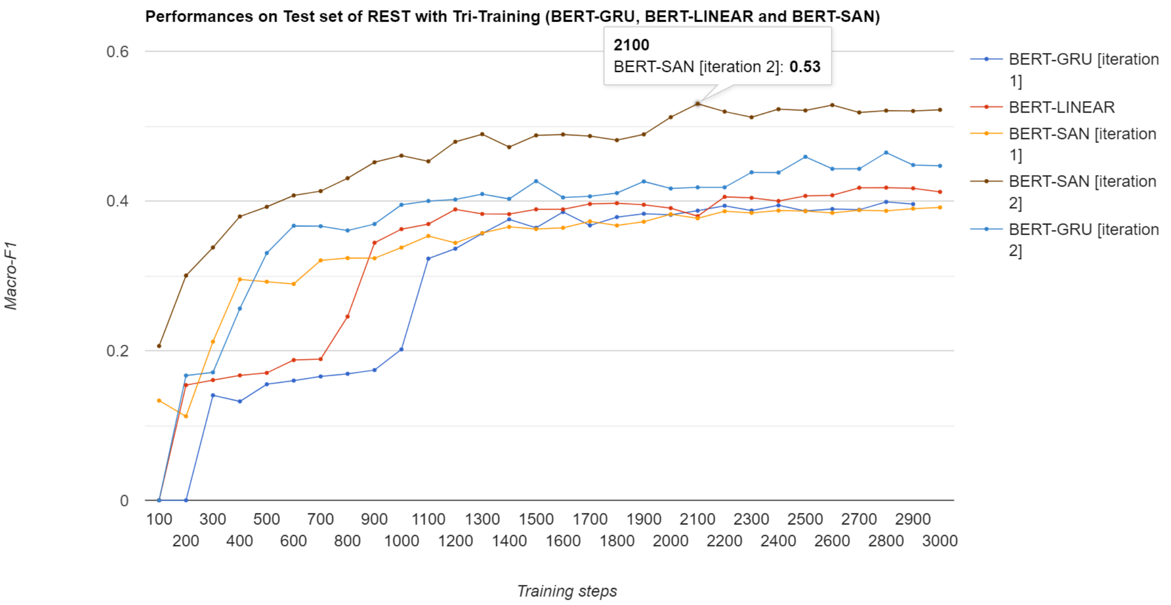


Figure 5.5: Model Performances with Tri-Training across different training steps, with BERT-GRU, BERT-Linear, and BERT-SAN and with REST

It can also be seen that the model performances improved over the baseline results and the self-training counterparts. This can be seen from Tables 5.5 and 5.6 that gives an overall comparison of all methods. The values presented for the Tri-Training are the best Macro-F1 values for that individual model considering all the Tri-Training implementations it was part of.

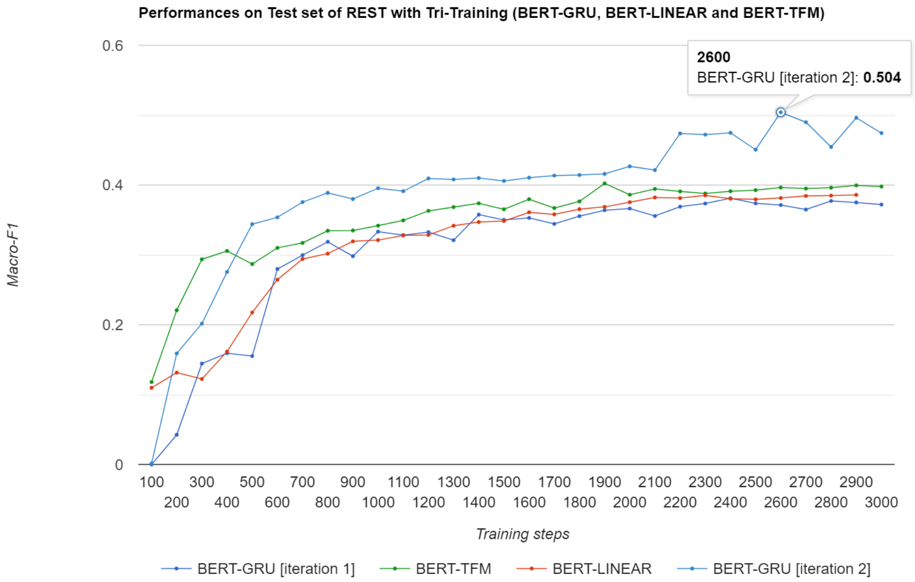


Figure 5.6: Model Performances with Tri-Training across different training steps, with BERT-GRU, BERT-TFM, and BERT-Linear and with REST

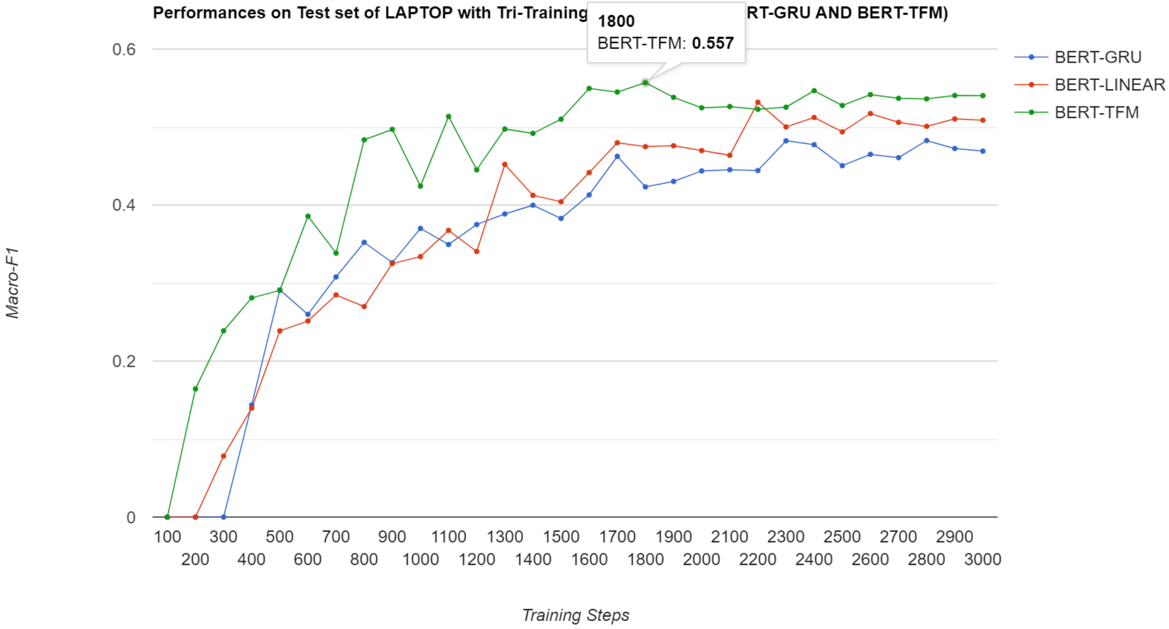


Figure 5.7: Model Performances with Tri-Training across different training steps, with BERT-GRU, BERT-TFM, and BERT-Linear and with LAPTOP

In the table, we also present the iteration info and all the models involved in the corresponding implementation.

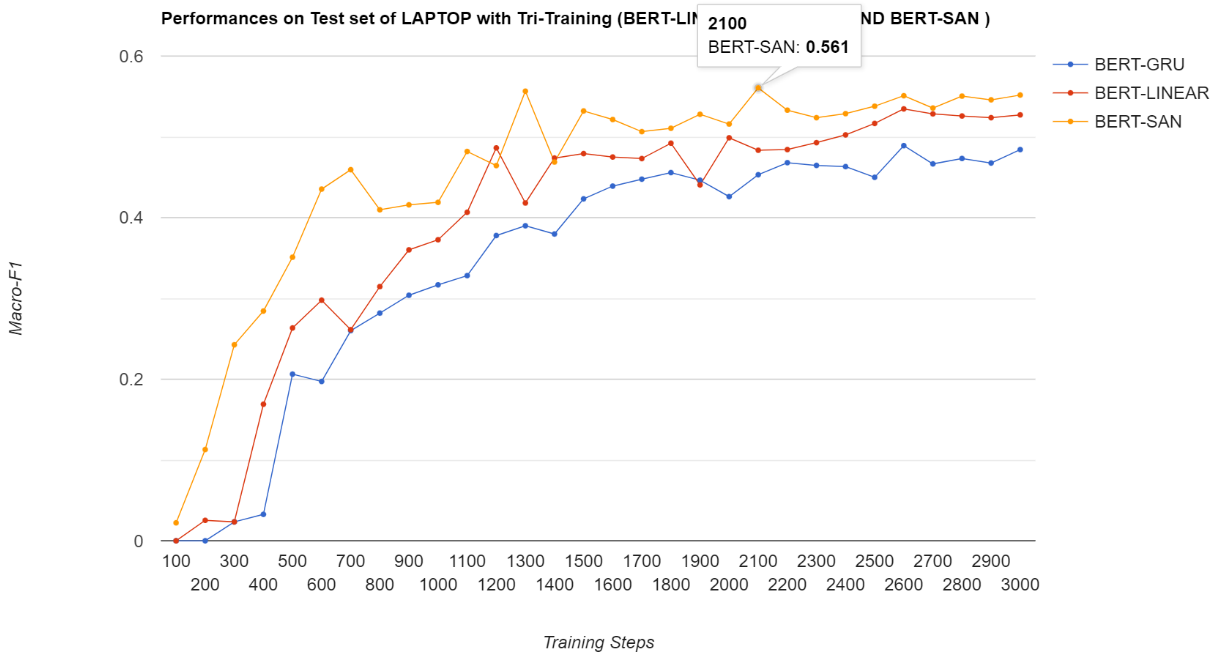


Figure 5.8: Model Performances with Tri-Training across different training steps, with BERT-GRU, BERT-Linear, and BERT-SAN and with LAPTOP

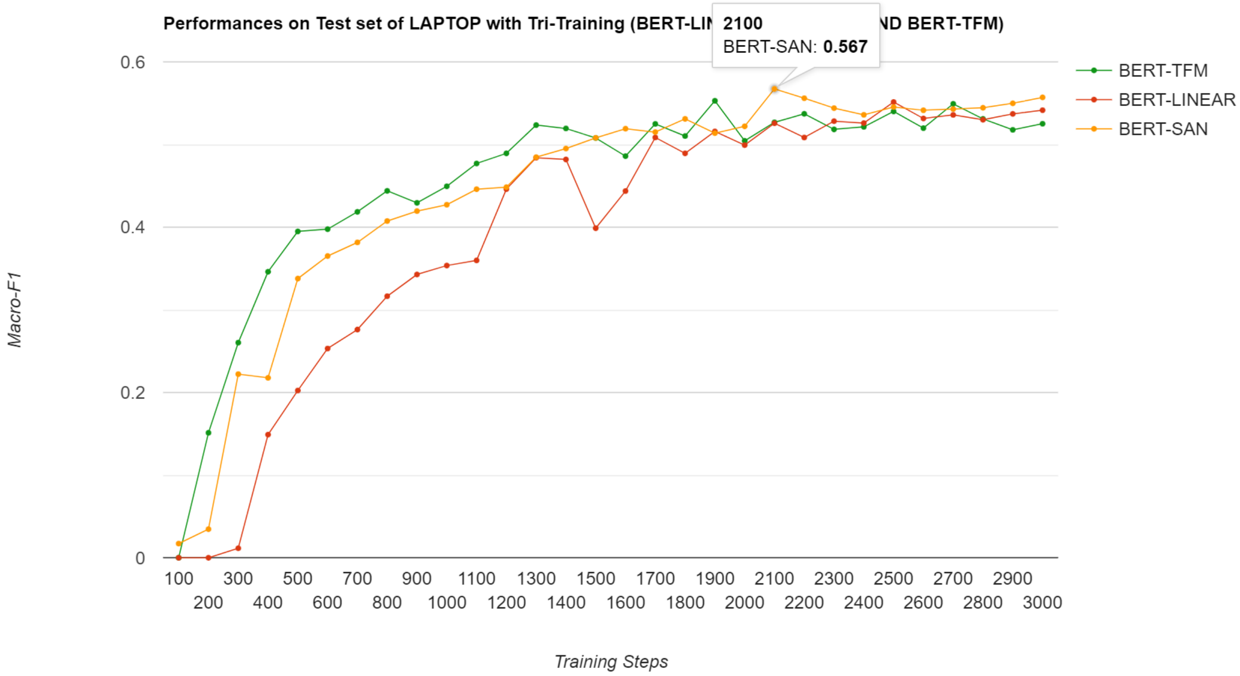


Figure 5.9: Model Performances with Tri-Training across different training steps, with BERT-Linear, BERT-TFM, and BERT-SAN and with LAPTOP

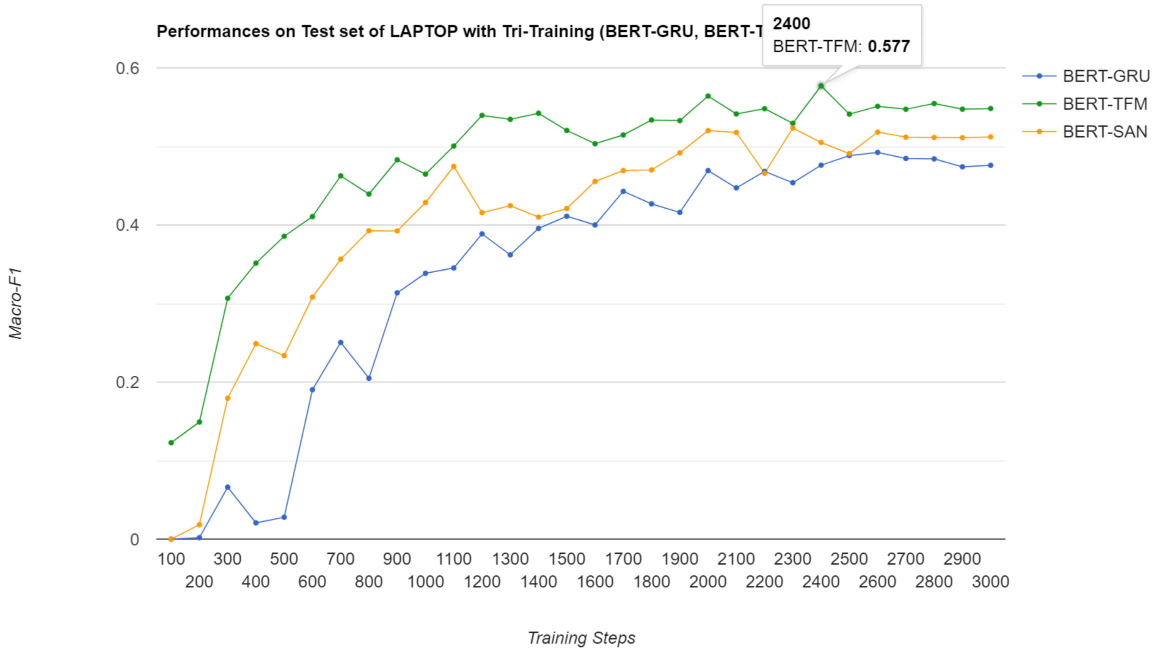


Figure 5.10: Model Performances with Tri-Training across different training steps, with BERT-GRU, BERT-TFM, and BERT-SAN and with LAPTOP

	Models	Iteration	Models $h_1 - h_2 - h_3$	Macro-f1	Micro-f1	Precision	Recall
Baseline	BERT + Linear	–	–	0.4459	0.6476	0.6556	0.6398
	BERT + GRU	–	–	0.4181	0.6612	0.6650	0.6576
	BERT + SAN	–	–	**0.5711**	**0.7158**	0.7147	**0.7170**
	BERT + TFM	–	–	0.3989	0.6323	0.6535	0.6125
Self-Training	BERT + Linear	1	–	0.3947	0.6361	0.6598	0.6141
	BERT + GRU	4	–	0.4057	0.6469	0.6443	0.6495
	BERT + SAN	1	–	0.4847	0.6757	0.7264	0.6318
	BERT + TFM	1	–	**0.5324**	**0.6974**	0.6908	**0.7041**
Tri-Training	BERT + Linear	2	san-tfm-linear	**0.5591**	**0.7011**	**0.7183**	**0.6849**
	BERT + GRU	2	gru-tfm-linear	**0.5041**	**0.6970**	**0.7081**	**0.6865**
	BERT + SAN	2	san-gru-tfm	0.5497	0.7055	**0.7331**	0.6800
	BERT + TFM	1	san-tfm-linear	0.4791	0.6816	**0.7095**	0.6559

Table 5.5: Comparison between the Baseline results, and their Self-Training, and Tri-Training counterparts, with REST.

	Models	Iteration	Models $h_1 - h_2 - h_3$	Macro-f1	Micro-f1	Precision	Recall
Baseline	BERT + Linear	–	–	**0.5672**	**0.6105**	0.6273	**0.5946**
	BERT + GRU	–	–	**0.5524**	**0.6107**	**0.6244**	**0.5978**
	BERT + SAN	–	–	0.5637	**0.6189**	0.6244	**0.6136**
	BERT + TFM	–	–	0.5541	0.6100	0.6262	0.5946
Self-Training	BERT + Linear	3	–	0.5411	0.5893	0.6162	0.5647
	BERT + GRU	1	–	0.5333	0.5845	0.5970	0.5726
	BERT + SAN	2	–	0.5559	0.6040	0.6261	0.5836
	BERT + TFM	10	–	0.5635	0.6094	0.6267	0.5931
Tri-Training	BERT + Linear	1	san-tfm-linear	0.5515	0.6058	**0.6453**	0.5710
	BERT + GRU	1	gru-tfm-san	0.4924	0.5680	0.6120	0.5300
	BERT + SAN	1	san-linear-tfm	**0.5674**	0.6052	**0.6521**	0.5647
	BERT + TFM	1	gru-tfm-san	**0.5772**	**0.6237**	**0.6448**	**0.6041**

Table 5.6: Comparison between the Baseline results, and their Self-Training, and Tri-Training counterparts, with LAPTOP.

5.3 E2E-ABSA across Domains

In the first part of this section, we present additional info and results for the procedure of classifying domains. In the next part, we discuss the experiments and present the results of E2E-ABSA for each of these domains.

5.3.1 Domain Segregation using Topic Modeling

As explained in section 4.3.1, we employed Topic Modeling for the distribution of domains. Specifically, we used Latent Dirichlet Allocation (LDA) for this purpose. Out of 2467 documents in total, the LDA model was trained with a corpus consisting of 1727 documents and was tasked to distribute the rest of 740 documents into multiple domains. We identify these 740 documents that are classified into topics as domains.

We then made use of gensim's *LDAMulticore* implementation of LDA and ran it against both the bag of words corpus and TF-IDF vectors. We ran the model against the number of topics set to between 4 and 20 to evaluate coherence values in all settings. Other hyperparameters such as *alpha* and *eta* were varied by a factor of 0.10 in the range of 0.01 to 1. Other values for *alpha* included the 'symmetric' and 'asymmetric' and for *eta*, the value 'symmetric'. These were interchanged one after the other in each run and for every run, we also simultaneously checked the coherence score c_v. We made use of gensim's *CoherenceModel* implementation to get the coherence score.

Among the results we obtained from the model, the one with 5 topics had a reasonable document distribution and a coherence score of 0.43. The other hyperparameters in this setting were:

- $alpha = 0.81$
- $eta =$ symmetric
- $random\_state = 100$
- $chunksize = 10$
- $passes = 10$

The distribution of 740 documents among 5 topics or, in our case, 5 domains were: 175-138-220-104-103, in that order. The topic to words distribution for the individual topics can be seen in the table 5.7.

Topic	**Word distribution**
Topic-1	$0.007 \times state + 0.007 \times work + 0.007 \times school + 0.006 \times united+$ $0.006 \times states + 0.006 \times country + 0.006 \times world + 0.005 \times countries$
Topic-2	$0.032 \times movie + 0.02 \times stutter + 0.012 \times kind + 0.008 \times story + 0.008 \times mean +$ $0.008 \times pretty + 0.008 \times love + 0.008 \times basically$
Topic-3	$0.011 \times business + 0.01 \times money + 0.008 \times company + 0.008 \times talking + 0.007 \times$ $speech + 0.006 \times financial + 0.006 \times loan + 0.005 \times patrick$
Topic-4	$0.01 \times video + 0.007 \times online + 0.007 \times marketing + 0.006 \times consumers +$ $0.006 \times course + 0.006 \times experience + 0.006 \times time + 0.005 \times that$
Topic-5	$0.007 \times jesus + 0.006 \times world + 0.006 \times health + 0.006 \times water + 0.006 \times body +$ $0.006 \times customer + 0.006 \times change + 0.006 \times view$

Table 5.7: Topic to words distribution. Here the numerical values indicate the probability of the corresponding word for the respective topic. The words themselves are arranged in the decreasing order of the probabilities.

from the above table 5.7 and a close inspection of a few documents for each domain, we could closely associate each domain with the following concepts:

- **Domain-1** ≈ Common topics such as school/work/politics etc.
- **Domain-2** ≈ Movie reviews/ Tv series reviews/ stories.
- **Domain-3** ≈ Financial/Business.
- **Domain-4** ≈ Marketing/Promotion
- **Domain-5** ≈ Other Misc.

5.3.2 E2E-ABSA across Domains

In this section, we briefly present the outlook for both the Self-Training and Tri-Training methods for cross-domain evaluation. We then finally present the results of these evaluations for all the domains.

With Self-Training and Tri-Training

Out of the 740 documents classified into five different domains, we derived 8250 sentence samples in total. The composition of sentence samples for each of the five domains is given in section 4.3.2.

Also, as previously outlined in the sections 4.3.2 and 4.3.3, we use these sentence samples from each domain for prediction at the end of every iteration. This is analogous to both the Self-Training and Tri-Training implementations. Only that, in the case of the latter, we make every model that is part of the implementation predict these sentences. The rest of the settings remain intact for consistency.

For evaluation of the predictions, since the sentence samples were part of the unlabeled set that contained sentiment and emotion labels, we could leverage only the former for sentiment evaluation. Evaluation for aspect terms, however, is done with the help of additional annotations. With the techniques used to evaluate sentiment and aspect words already presented in sections 3.3.1 and 3.3.2, we present the results in the next section.

Evaluation on Sentiment

Section 3.3.1 presents in detail about the sentiment labels available as part of CMU-MOSEI. We also explain how we leverage these labels for sentiment evaluation. It is important to mention that during aligning the computational sequences from CMU-MOSEI i.e., the *CMU_MOSEI_ Labels* and the *CMU_MOSEI_TimestampedWords*, we lost many sentences and their labels. After alignment, when we tried to access labels for the combined set of 8250 sentence samples from 5 domains, we only managed to obtain them for 3693 samples.

Also, in section Section 3.3.1, we also presented the logic to compare the predictions made for sentiment by the model and the label from CMU-MOSEI, both of which were in a different format. We evaluate the predictions made on samples belonging to different domains based on this semi-automatic logic.

	Models	Iteration	Models $h_1 - h_2 - h_3$	Macro-f1	Micro-f1	Precision	Recall
Self-Training	BERT + Linear	1	–	0.4683	0.7048	0.7049	0.7049
	BERT + GRU	1	–	0.4673	0.7383	0.7383	0.7383
	BERT + SAN	4	–	0.4770	0.7164	0.7164	0.7164
	BERT + TFM	1	–	0.4726	0.7350	0.7351	0.7351
Tri-Training	BERT + Linear	1	san-tfm-linear, gru-tfm-linear, linear-gru-san	0.4683	0.7049	0.7049	0.7049
	BERT + GRU	1	gru-tfm-san	0.4839	0.7673	0.7674	0.7674
	BERT + SAN	1	san-tfm-linear	0.4592	0.7031	0.7031	0.7031
	BERT + TFM	1	gru-tfm-linear	0.4572	0.6961	0.6961	0.6961

Table 5.8: Comparison between the Self-Training and Tri-Training results on Sentiment, with REST. Evaluation is done with all domains combined.

With the above arrangement, we then present the results for both the Self-Training and Tri-Training implementations in tables 5.8 and 5.9. Here the presented results are for all domains

	Models	Iteration	Models $h_1 - h_2 - h_3$	Macro-f1	Micro-f1	Precision	Recall
Self-Training	BERT + Linear	6	–	0.5298	0.5684	0.5684	0.5684
	BERT + GRU	5	–	0.5149	0.5585	0.5585	0.5585
	BERT + SAN	6	–	0.5529	0.6114	0.6114	0.6114
	BERT + TFM	6	–	0.5438	0.5724	0.5725	0.5725
Tri-Training	BERT + Linear	1	san-tfm-linear	0.5248	0.5421	0.5421	0.5421
	BERT + GRU	1	gru-linear-san	0.5077	0.5245	0.5245	0.5245
	BERT + SAN	1	san-tfm-linear	0.5609	0.5858	0.5858	0.5858
	BERT + TFM	1	san-tfm-linear	0.5362	0.5601	0.5601	0.5601

Table 5.9: Comparison between the Self-Training and Tri-Training results on Sentiment, with LAPTOP. Evaluation is done for all domains combined.

together combined. Again, we also give the iterations at which the individual models scored such performances in a separate column. In the case of Tri-training, the models involved in the implementation are also included in a separate column.

The results for experiments with LAPTOP generally outscored the REST in terms of Macro-F1. But the Micro-F1 scores for experiments with REST were far superior to that of LAPTOP. We can associate two reasons for this behavior: First, this could partially be due to the comparatively lesser number of samples in REST, and second, there were class imbalances concerning the sentiment in the samples. Table 5.23 gives an account for the same. The imbalances are more prominent with REST than the LAPTOP. Since Micro-F1 generally accounts for this class imbalance and because REST has more of this, it comes off with a relatively better score. It is for the same reason the Macro-F1 scores suffer due to the contributions from all classes (in this case the sentiments) being treated equally.

	Models	Iteration	Models $h_1 - h_2 - h_3$	Macro-f1	Micro-f1	Precision	Recall
Self-Training	BERT + Linear	1	–	0.4522	0.6776	0.6776	0.6776
	BERT + GRU	1	–	0.565	0.7721	0.7721	0.7721
	BERT + SAN	3	–	0.5073	0.7835	0.7835	0.7835
	BERT + TFM	1	–	0.4994	0.7344	0.7344	0.7344
Tri-Training	BERT + Linear	2	san-tfm-linear	0.5807	0.8695	0.8695	0.8695
	BERT + GRU	1	gru-tfm-san	0.512	0.7994	0.7994	0.7994
	BERT + SAN	2	san-gru-linear	0.6443	0.8888	0.8888	0.8888
	BERT + TFM	1	san-tfm-linear	0.4486	0.6841	0.6841	0.6841

Table 5.10: Comparison between the results of Self-Training and Tri-Training, on the Sentiment, with the REST and Domain-1.

We also present the results for Domain-1 in tables 5.10 and 5.11. As can be seen from the tables, the results from the models trained with REST generalized better to Domain-1, in terms of sentiment. This is reasonable since the samples from Domain-1, are roughly more relatable to that of REST since most samples from LAPTOP were either specifically talking about a whole product or a certain facet of it but usually containing domain-specific jargon. This meant that LAPTOP might not generalize well with jargon for topics like politics, work, etc., present in Domain-1.

	Models	**Iteration**	**Models $h_1 - h_2 - h_3$**	**Macro-f1**	**Micro-f1**	**Precision**	**Recall**
Self-Training	BERT + Linear	3	–	0.5531	0.5652	0.5652	0.5652
	BERT + GRU	3	–	0.5569	0.5607	0.5607	0.5607
	BERT + SAN	3	–	0.5695	0.6146	0.6146	0.6146
	BERT + TFM	11	–	0.5362	0.5736	0.5736	0.5736
Tri-Training	BERT + Linear	1	san-gru-linear	0.4558	0.5277	0.5277	0.5277
	BERT + GRU	1	gru-linear-san	0.487	0.4285	0.4285	0.4285
	BERT + SAN	1	san-gru-linear, san-tfm-linear	0.5234	0.5714	0.5714	0.5714
	BERT + TFM	1	san-tfm-linear	0.4958	0.4827	0.4827	0.4827

Table 5.11: Comparison between the results of Self-Training and Tri-Training, on the Sentiment, with the LAPTOP and Domain-1.

Results for Domain-2 are in tables 5.12 and 5.13. Unlike the results for Domain-1, the runs with LAPTOP resulted in a better Macro-F1 score than that of REST.

	Models	**Iteration**	**Models $h_1 - h_2 - h_3$**	**Macro-f1**	**Micro-f1**	**Precision**	**Recall**
Self-Training	BERT + Linear	1	–	0.4661	0.6817	0.6817	0.6817
	BERT + GRU	2	–	0.4621	0.6806	0.6806	0.6806
	BERT + SAN	5	–	0.4677	0.7053	0.7053	0.7053
	BERT + TFM	4	–	0.4963	0.7394	0.7394	0.7394
Tri-Training	BERT + Linear	1	san-tfm-linear, gru-tfm-linear, linear-gru-san	0.4661	0.6817	0.6817	0.6817
	BERT + GRU	1	gru-tfm-linear	0.4364	0.6904	0.6904	0.6904
	BERT + SAN	2	san-gru-linear	0.484	0.846	0.846	0.846
	BERT + TFM	1	gru-tfm-linear	0.4421	0.6736	0.6736	0.6736

Table 5.12: Comparison between the results of Self-Training and Tri-Training, on the Sentiment, with the REST and Domain-2.

Samples from Domain-2 had the most movie review samples out of all domains. These review samples specifically spoke about a certain aspect of a movie/show that we think might have related more to the component-specific samples from LAPTOP than the REST.

	Models	Iteration	Models $h_1 - h_2 - h_3$	Macro-f1	Micro-f1	Precision	Recall
Self-Training	BERT + Linear	1	–	0.5851	0.5740	0.5740	0.5740
	BERT + GRU	1	–	0.5276	0.5199	0.5199	0.5199
	BERT + SAN	1	–	0.6108	0.6348	0.6349	0.6349
	BERT + TFM	3	–	0.5945	0.6181	0.6181	0.6181
Tri-Training	BERT + Linear	1	gru-tfm-linear	0.7555	0.6666	0.6666	0.6666
	BERT + GRU	1	gru-san-linear	0.7662	0.7499	0.7499	0.7499
	BERT + SAN	1	san-tfm-linear	0.7643	0.7692	0.7692	0.7692
	BERT + TFM	1	gru-tfm-linear	0.7488	0.7777	0.7777	0.7777

Table 5.13: Comparison between the results of Self-Training and Tri-Training, on the Sentiment, with the LAPTOP and Domain-2.

Tables 5.14 and 5.15 correspond to the sentiment results for Domain-3. The results to this domain were the best in all of our cross-domain analysis evaluations. In terms of Macro-F1, the results were mostly similar with both the REST and LAPTOP. However, the Micro-F1 results were extremely better with REST.

	Models	Iteration	Models $h_1 - h_2 - h_3$	Macro-f1	Micro-f1	Precision	Recall
Self-Training	BERT + Linear	1	–	0.4863	0.7412	0.7412	0.7412
	BERT + GRU	1	–	0.4868	0.7739	0.7739	0.7739
	BERT + SAN	3	–	0.5363	0.7949	0.7949	0.7949
	BERT + TFM	1	–	0.5225	0.8108	0.8108	0.8108
Tri-Training	BERT + Linear	1	linear-gru-san	0.5636	0.8709	0.8709	0.8709
	BERT + GRU	1	gru-tfm-san	0.6570	0.9523	0.9523	0.9523
	BERT + SAN	1	san-gru-linear	0.5248	0.7701	0.7701	0.7701
	BERT + TFM	1	san-tfm-linear	0.5892	0.9443	0.9443	0.9443

Table 5.14: Comparison between the results of Self-Training and Tri-Training, on the Sentiment, with the REST and Domain-3.

The samples from Domain-3 mostly consisted of themes such as investing, stocks, and other financial-themed concepts. We see that samples from REST generalize well with these samples, in terms of sentiment.

	Models	**Iteration**	**Models $h_1 - h_2 - h_3$**	**Macro-f1**	**Micro-f1**	**Precision**	**Recall**
Self-Training	BERT + Linear	2	–	0.5366	0.55	0.55	0.55
	BERT + GRU	8	–	0.5287	0.5787	0.5787	0.5787
	BERT + SAN	4	–	0.5717	0.6064	0.6064	0.6064
	BERT + TFM	5	–	0.5704	0.5917	0.5917	0.5917
Tri-Training	BERT + Linear	1	linear-gru-san	0.5421	0.5833	0.5833	0.5833
	BERT + GRU	1	gru-linear-san	0.5962	0.6451	0.6451	0.6451
	BERT + SAN	1	san-tfm-linear	0.6242	0.6779	0.6779	0.6779
	BERT + TFM	1	san-tfm-linear	0.6230	0.6462	0.6463	0.6463

Table 5.15: Comparison between the results of Self-Training and Tri-Training, on the Sentiment, with the LAPTOP and Domain-3.

Results for Domain-4 are presented in tables 5.16 and 5.17. Domain-4 mostly contained samples from the area of Marketing, self-promotion, product promotions, etc. These samples related most to the LAPTOP than the REST, in terms of sentiment. We think that some of the product-related promotions were similar in nature to the samples from LAPTOP and hence the results.

	Models	**Iteration**	**Models $h_1 - h_2 - h_3$**	**Macro-f1**	**Micro-f1**	**Precision**	**Recall**
Self-Training	BERT + Linear	20	–	0.4399	0.6948	0.6948	0.6948
	BERT + GRU	1	–	0.4894	0.7659	0.7659	0.7659
	BERT + SAN	11	–	0.4709	0.7268	0.7268	0.7268
	BERT + TFM	2	–	0.4674	0.7641	0.7641	0.7641
Tri-Training	BERT + Linear	2	san-tfm-linear	0.5078	0.7993	0.7998	0.7998
	BERT + GRU	1	gru-tfm-san	0.4888	0.7499	0.7499	0.7499
	BERT + SAN	1	san-tfm-gru	0.4466	0.7113	0.7113	0.7113
	BERT + TFM	1	gru-tfm-linear	0.4592	0.7107	0.7107	0.7107

Table 5.16: Comparison between the results of Self-Training and Tri-Training, on the Sentiment, with the REST and Domain-4.

	Models	Iteration	Models $h_1 - h_2 - h_3$	Macro-f1	Micro-f1	Precision	Recall
Self-Training	BERT + Linear	7	–	0.5718	0.6428	0.6428	0.6428
	BERT + GRU	4	–	0.6160	0.6233	0.6233	0.6233
	BERT + SAN	10	–	0.6273	0.6774	0.6774	0.6774
	BERT + TFM	7	–	0.6411	0.6568	0.6568	0.6568
Tri-Training	BERT + Linear	1	san-gru-linear	0.625	0.6135	0.6135	0.6135
	BERT + GRU	1	gru-linear-san	0.5269	0.5172	0.5172	0.5172
	BERT + SAN	1	san-tfm-linear	0.6502	0.6169	0.6170	0.6170
	BERT + TFM	1	gru-tfm-linear	0.7752	0.7045	0.7045	0.7045

Table 5.17: Comparison between the results of Self-Training and Tri-Training, on the Sentiment, with the LAPTOP and Domain-4.

Finally, 5.18 and 5.19 presents the results for Domain-5. The samples from this domain included all the other themes not found in Domain-1 to Domain-4. In that sense, the specificity of this domain to a certain concept was comparatively lesser than the others. This could explain the comparatively lower values with both REST and LAPTOP. But the Micro-F1, precision, and recall values with REST were superior to that of LAPTOP.

	Models	Iteration	Models $h_1 - h_2 - h_3$	Macro-f1	Micro-f1	Precision	Recall
Self-Training	BERT + Linear	1	–	0.4693	0.7009	0.7009	0.7009
	BERT + GRU	1	–	0.4153	0.6730	0.6730	0.6730
	BERT + SAN	4	–	0.4888	0.6750	0.6750	0.6750
	BERT + TFM	4	–	0.4659	0.7091	0.7091	0.7091
Tri-Training	BERT + Linear	1	san-tfm-linear, gru-tfm-linear	0.4693	0.7009	0.7009	0.7009
	BERT + GRU	1	gru-tfm-linear	0.4555	0.6328	0.6328	0.6328
	BERT + SAN	1	san-gru-linear	0.4627	0.7173	0.7173	0.7173
	BERT + TFM	1	gru-tfm-san	0.4882	0.7599	0.7599	0.7599

Table 5.18: Comparison between the results of Self-Training and Tri-Training, on the Sentiment, with the REST and Domain-5.

	Models	**Iteration**	**Models $h_1 - h_2 - h_3$**	**Macro-f1**	**Micro-f1**	**Precision**	**Recall**
Self-Training	BERT + Linear	7	–	0.4983	0.5403	0.5404	0.5404
	BERT + GRU	5	–	0.5280	0.5746	0.5746	0.5746
	BERT + SAN	8	–	0.5294	0.5802	0.5802	0.5802
	BERT + TFM	13	–	0.5139	0.5392	0.5392	0.5392
Tri-Training	BERT + Linear	1	san-tfm-linear	0.4393	0.4726	0.4727	0.4727
	BERT + GRU	1	gru-tfm-linear	0.4358	0.4999	0.4999	0.4999
	BERT + SAN	1	san-tfm-linear	0.4436	0.4716	0.4716	0.4716
	BERT + TFM	1	gru-tfm-san, linear-tfm-san	0.4241	0.5073	0.5073	0.5073

Table 5.19: Comparison between the results of Self-Training and Tri-Training, on the Sentiment, with the LAPTOP and Domain-5.

Evaluation on Aspect

In section 3.3.2, we laid out the need for additional annotations to undergo this evaluation. We also presented the guidelines that the annotators were presented for the annotation process. Since it was impossible to annotate all 8250 samples, we sampled a set of 200 samples that were shared with the annotators. We hence evaluate the model's prediction on these select samples and since these were very few in number, we only present the evaluation results on the aspect for all domains combined.

For annotation, the annotators identified possible aspect words on the samples. Additionally, we also asked the annotators to provide the sentiment info along the aspect words. While this sentiment info is not used for evaluation (as it is already realized with true labels in the previous section), we use them to infer the distribution of aspect words to these sentiments (more on this later).

	Models	**Iteration**	**F1**	**Precision**	**Recall**
with REST	BERT + Linear	11	0.1651	0.2812	0.1169
	BERT + GRU	12	0.1238	0.2857	0.079
	BERT + SAN	15	0.1579	0.2432	0.1169
	BERT + TFM	17	0.167	0.2903	0.1169
with LAPTOP	BERT + Linear	15	0.1495	0.2666	0.1039
	BERT + GRU	13	0.1766	0.36	0.117
	BERT + SAN	10	0.16	0.3478	0.1039
	BERT + TFM	8	0.1348	0.4286	0.08

Table 5.20: Results for the evaluation of Aspect, with Self-Training, and both REST and LAPTOP.

Table 5.20 gives the results of aspect evaluation on REST and LAPTOP, with Self-Training. Table 5.21 gives the results corresponding to Tri-Training runs. The results presented are the best results out of all three annotation sets used. The results for other annotations contained F1 values in the range of 0.08 to 0.1 for Self-Training. In the case of Tri-Training, the values were in the range of 0.05 to 0.07. Hence, we just mention this and provide values corresponding to relatively better scores.

	Models	**Iteration**	**Models** $h_1 - h_2 - h_3$	**F1**	**Precision**	**Recall**
with REST	BERT + Linear	1	san-tfm-linear, gru-tfm-linear, linear-gru-san	0.082	0.2353	0.05
	BERT + GRU	2	gru-tfm-linear	0.056	0.4286	0.03
	BERT + SAN	1	san-tfm-gru	0.069	0.3	0.0389
	BERT + TFM	1	gru-tfm-san	0.063	0.158	0.039
with LAPTOP	BERT + Linear	1	san-tfm-linear	0.063	0.158	0.039
	BERT + GRU	1	gru-linear-tfm	0.082	0.2353	0.05
	BERT + SAN	1	san-tfm-linear	0.1	0.99	0.05
	BERT + TFM	1	san-tfm-linear, gru-tfm-linear	0.092	0.5714	0.05

Table 5.21: Results for the evaluation of Aspect, with Tri-Training, and both REST and LAPTOP.

With results being subpar and indicating scope for improvement, a closer inspection of the sample annotations and predictions yielded the following observations:

The predictions were most successful in instances of positive aspect words but failed in most occasions of neutral aspect words. This behavior was in line with the predictions made on the test set of REST (and LAPTOP) where the selection of aspect words was most successful when the underlying sentiments were in the absolutes i.e, positive or negative.

There were two reasons for this. The first reason concerns the distribution of sentiment on the 200 samples tasked with annotating aspect words. Since the annotators were also tasked to denote their opinion on the sentiment associated to the aspect words, we could conduct a detailed analysis of the distribution of sentiment in their annotations.

Out of 200 samples submitted to the annotators, While Annotator 1 found 66 samples to be containing at least a single aspect word, Annotator 2 and Annotator 3 found 49 and 28 samples to be containing aspect words. The table 5.22 shows the distribution of the aspect words to the sentiment in all these annotations.

From the table, it can be seen that most of the aspect words were marked with the neutral sentiment and there were almost no negative marked aspect words found in the samples.

The second reason concerns the number of labeled samples with neutral sentiment for REST and LAPTOP. The distribution is given in the table 5.23. For LAPTOP, the number of aspect words with neutral sentiment is relatively lower by almost half when compared to either of the other sentiments. For REST, this number is much lower.

Hence, we believe that the combination of relatively fewer neutral sentiment samples in the original training set and the subsequent presence of a higher number of neutral sentiment

Annotators	Number of samples containing Aspect words	Distribution of number of Aspect words for Sentiment		
		NEU	POS	NEG
Annotator-1	66	75	5	0
Annotator-2	49	36	24	2
Annotator-3	28	25	4	2

Table 5.22: Distribution of Sentiment to the Aspect words for samples employed for Aspect evaluation.

Train Set	Number of samples containing Aspect words	Distribution of number of Aspect words for Sentiment		
		NEU	POS	NEG
REST	1114	77	1644	543
LAPTOP	1308	666	1222	1142

Table 5.23: Distribution of Sentiment to the Aspect words, for the Original labeled data.

samples in the annotations of the unlabeled set was one of the reasons for the results.

Additionally, we tried to uncover the agreement between all three annotations and found that the opinions were diverse too. From table 5.22, it can be seen that Annotator 1 and Annotator 2 found aspect words for 66 and 49 samples respectively. But, the opinions were in-agreement for only 25 samples. In terms of aspect words, they agreed on 28 aspect words in total. Between Annotator 2 and Annotator 3 the agreement found was only on 6 samples and 6 aspect words. And finally, between Annotator 1 and Annotator 3, the agreement was made on 10 samples and aspect words in total. This also indicates the varied nature of opinions themselves.

5.4 E2E-ABSA across Erroneous Transcriptions

Section 4.4.2 explains in detail the procedure for this section. We conduct two separate runs for a single implementation. As unlabeled data, the same implementation uses AWS transcribed samples in one run and in another, uses the manually transcribed (gold standard) counterparts. This holds for both the Self-Training runs as well as of Tri-Training. As part of unlabeled data for training, we prepared a set of 4106 samples. As part of the evaluation for sentiment, we prepared another set of 1825 samples. Both of these were from CMU-MOSEI. To evaluate the direct consequence of using either the AWS transcribed samples or the gold standard samples as the unlabeled set, we also conduct the E2E-ABSA evaluation on the test set of the labeled data.

On the whole, however, the above study is done twice, once with both the REST and LAPTOP.

We first present the direct E2E-ABSA results with the runs made incorporating the AWS transcribed samples and gold standard samples as the unlabeled data, in section 5.4.1. We then discuss the results for the evaluation of sentiment under two scenarios:

- In section 5.4.2, Sentiment evaluation for instances that were erroneous compared to the gold standard samples, compared.

- In section 5.4.3,Sentiment evaluation under only the spontaneous speech instances, compared with both the AWS transcribed and the gold standard samples.

5.4.1 E2E-ABSA across AWS Transcriptions and Gold Standard

In this section, we present the E2E-ABSA results for both the Self-Training and Tri-Training implementations with both the AWS transcribed samples and the gold standard samples incorporated as unlabeled samples. The results were carried out on the test set of the labeled data of REST (on tables 5.24 and 5.26) and LAPTOP(5.25 and 5.27)

	Models	**Iteration**	**Macro-f1**	**Micro-f1**	**Precision**	**Recall**
with AWS	BERT + Linear	1	0.5127	0.6661	0.6854	0.6479
	BERT + GRU	1	**0.5675**	**0.7122**	**0.7122**	0.7122
	BERT + SAN	3	0.5760	0.7084	0.7084	**0.7084**
	BERT + TFM	2	0.5952	**0.7115**	**0.7367**	0.6881
gold standard	BERT + Linear	2	**0.5214**	**0.6957**	**0.6985**	**0.6929**
	BERT + GRU	4	0.5538	0.7012	0.6862	**0.7170**
	BERT + SAN	1	**0.5780**	**0.7098**	**0.7241**	0.6961
	BERT + TFM	1	**0.5981**	0.7040	0.7122	**0.6961**

Table 5.24: Comparison between the E2E-ABSA results for samples from AWS Transcribe and gold standard, with Self-Training and REST.

	Models	**Iteration**	**Macro-f1**	**Micro-f1**	**Precision**	**Recall**
with AWS	BERT + Linear	4	0.5754	0.6125	0.6315	0.5946
	BERT + GRU	1	0.5875	0.6257	0.6238	**0.6278**
	BERT + SAN	6	0.5817	0.6291	0.6509	**0.6088**
	BERT + TFM	2	0.5816	**0.6302**	0.6515	**0.6104**
gold standard	BERT + Linear	4	**0.5844**	**0.6202**	**0.6373**	**0.6041**
	BERT + GRU	2	**0.5958**	**0.6344**	**0.6480**	0.6215
	BERT + SAN	4	**0.5897**	**0.6350**	**0.6714**	0.6025
	BERT + TFM	2	**0.5877**	0.6269	**0.6591**	0.5978

Table 5.25: Comparison between the E2E-ABSA results for samples from AWS Transcribe and gold standard, with Self-Training and LAPTOP.

As can be seen from the above tables, the results with Self-Training and from the AWS transcriptions were very close to the gold standard ones. The results with gold standard labels were slightly better overall, but the differences were not so stark. We can even see that in some cases, surprisingly, the results ended up being better with the AWS Transcriptions.

But in Tri-Training runs, the AWS transcribed ones were better than the gold standard ones. The results are given in tables 5.26 and 5.27.

	Models	**Iteration**	**Models $h_1 - h_2 - h_3$**	**Macro-f1**	**Micro-f1**	**Precision**	**Recall**
with AWS	BERT + Linear	2	san-linear-gru	0.5650	0.7018	0.6979	**0.7058**
	BERT + GRU	2	san-tfm-gru	**0.5691**	**0.7233**	**0.7280**	**0.7186**
	BERT + SAN	2	san-tfm-gru	0.5967	0.7262	0.7274	**0.7251**
	BERT + TFM	1	san-tfm-gru	**0.6074**	**0.7293**	**0.7421**	0.7170
gold standard	BERT + Linear	2	linear-tfm-gru	**0.5764**	**0.7151**	**0.7351**	0.6961
	BERT + GRU	2	san-linear-gru	0.5587	0.7082	0.7043	0.7122
	BERT + SAN	2	san-tfm-gru	**0.5987**	**0.7274**	**0.7364**	0.7186
	BERT + TFM	1	san-tfm-gru	0.5999	0.7123	0.6986	**0.7267**

Table 5.26: Comparison between the E2E-ABSA results for samples from AWS Transcribe and gold standard, with Tri-Training and REST.

One reason for this could be that even though the word error rates between the transcripted samples and the gold standard ones were significant enough to be noticeable, it could be seen that the major notion of the samples was somewhat still retained. This could be because the major contributors to the word error rate were the errors in the subject and/or object words, and, in some cases, the articles and prepositions present in the samples. While this meant that there was still noticeable errors in transcription, the majority of it did not take away the impact on deduction of the inherent sentiment while testing on the labeled data.

Another interesting observation is the surprisingly better results when compared with the Self-Training and Tri-Training experiments run as part of the sections 4.2.1 and 4.2.2, the results of which were in the tables 5.5 and 5.6.

	Models	**Iteration**	**Models $h_1 - h_2 - h_3$**	**Macro-f1**	**Micro-f1**	**Precision**	**Recall**
with AWS	BERT + Linear	1	linear-gru-tfm	0.5819	0.6233	0.6421	0.6057
	BERT + GRU	2	san-gru-tfm	**0.5952**	**0.6416**	**0.6494**	**0.6341**
	BERT + SAN	2	linear-san-tfm	**0.5810**	0.6253	**0.6673**	0.5883
	BERT + TFM	1	linear-san-tfm	**0.6004**	**0.6441**	0.6579	**0.6309**
gold standard	BERT + Linear	2	linear-gru-tfm	**0.5958**	**0.6333**	**0.6598**	**0.6088**
	BERT + GRU	1	linear-gru-tfm	0.5781	0.6235	0.6391	0.6088
	BERT + SAN	1	san-tfm-gru	0.5809	**0.6291**	0.6620	**0.5994**
	BERT + TFM	1	linear-gru-tfm	0.5976	0.6393	**0.6791**	0.6041

Table 5.27: Comparison between the E2E-ABSA results for samples from AWS Transcribe and gold standard, with Tri-Training and LAPTOP.

We think that the experiments in this study, supplemented by approximately one-fifth of the original unlabeled data size had a constructive outcome. A controlled supply of pseudo-labeled samples resulting in better results could be one of the overall main consequences of this whole work. The experiments of this study and the comparison between Tri-Training and Self-Training runs for E2E-ABSA (as in tables 5.5 and 5.6), the former which incorporates this notion of controlled addition of training samples are an excellent demonstration of the same.

5.4.2 Evaluating Sentiment with Erroneous Transcriptions

As part of this task, we compare the sentiment inference from the Semi-supervised models both with the AWS transcripted data and the gold standard ones. For this task, we only consider those samples that were erroneous when compared to the gold standard ones, with the other samples excluded from the study. The main aim is to inspect if there is any significant change in inferencing sentiment when the underlying textual info differs from its original counterpart.

with Self-Training		with AWS Transcribe		with Gold Standard		
	Models	**Iteration**	**Macro-f1**	**Iteration**	**Macro-f1**	**Percentage Difference**
with REST	BERT + Linear	5	**0.324**	10	0.3166	2.31%
	BERT + GRU	3	**0.3718**	11	0.3526	5.3%
	BERT + SAN	16	**0.3627**	2	0.3247	11.06%
	BERT + TFM	10	**0.3511**	5	0.3095	12.59%
with LAPTOP	BERT + Linear	6	0.5487	2	**0.5633**	-2.63%
	BERT + GRU	12	**0.5613**	9	0.4670	18.34%
	BERT + SAN	2	0.5481	14	**0.5513**	-0.58%
	BERT + TFM	10	0.5296	1	**0.6332**	-17.82%

Table 5.28: Results for the evaluation on sentiment, for erroneous samples from AWS Transcribe and the gold standard, with Self-Training.

Out of the 151 test files containing 1825 samples, we found that there were errors in samples from 115 files. However, not all of the samples from these files were erroneous. We found that 1232 samples contained at least a single error. Since labels were missing for numerous samples from CMU-MOSEI, we ended up obtaining only 275 samples that could be evaluated.

Tables 5.28 and 5.29 gives a look at the results. The former presents the results with Self-Training, and the latter, with Tri-Training implementations. For clarity, we give only the highest Macro-F1 scores obtained in the individual runs while omitting the other metrics. Other metadata info such as iteration and in the case of Tri-Training, the individual models used are also presented. We also present the percentage difference between the results to denote the magnitude of the difference. The percentages are from the perspective of the AWS samples since they are the focus of this study.

As can be seen from the tables, the results with Self-Training surprisingly have the runs with AWS transcribed samples outscoring the gold standard counterparts. For Tri-Training implementations, however, the runs involving the gold standard samples were better than the AWS ones.

with Tri-Training		with AWS Transcribe			with Gold Standard			
	Models	**Iteration**	**Models $h_1 - h_2 - h_3$**	**Macro-f1**	**Iteration**	**Models $h_1 - h_2 - h_3$**	**Macro-f1**	**Percentage Difference**
with REST	BERT + Linear	2	san-linear-tfm	0.2947	3	san-linear-tfm	**0.3333**	-12.29%
	BERT + GRU	3	gru-linear-tfm	0.2962	3	san-gru-tfm	**0.3888**	-27.04%
	BERT + SAN	1	san-gru-tfm	0.3553	1	san-gru-tfm	**0.3587**	-0.95%
	BERT + TFM	1	gru-linear-tfm	0.3324	2	san-gru-tfm	**0.3333**	-0.27%
with LAPTOP	BERT + Linear	2	san-linear-tfm	0.5970	1	linear-gru-tfm	**0.7312**	-20.21%
	BERT + GRU	2	san-gru-tfm	0.6785	1	san-gru-tfm	**0.7714**	-12.81%
	BERT + SAN	1	san-linear-tfm	**0.6499**	1	san-linear-gru	0.6111	6.15%
	BERT + TFM	1	linear-gru-tfm	0.4841	1	san-linear-tfm	**0.5238**	-7.87%

Table 5.29: Results for the evaluation on sentiment, for erroneous samples from AWS Transcribe and the gold standard, with Tri-Training

5.4.3 Evaluating Sentiment on Spontaneous instances

In this study, we try to evaluate the sentiment for AWS transcripted samples and the gold standard ones in the setting of spontaneous speech instances.

Some videos from CMU-MOSEI are scripted as the person delivering a speech reads off of a manuscript. But it also contains videos where the user presents his views spontaneously. Here, the instances usually contain repetition, hesitation, or faltering between their delivery. We intend to investigate whether the inherent sentiment in those deliveries could still be determined. Along with these, we also conduct experiments from the transcripts obtained from AWS for such instances.

with Self-Training		with AWS Transcribe		with Gold Standard		
	Models	**Iteration**	**Macro-f1**	**Iteration**	**Macro-f1**	**Percentage Difference**
with REST	BERT + Linear	7	**0.3644**	7	0.3139	14.89%
	BERT + GRU	9	0.3434	2	**0.3889**	-12.43%
	BERT + SAN	13	**0.3936**	11	0.3314	17.16%
	BERT + TFM	1	**0.3889**	9	0.3308	16.15%
with LAPTOP	BERT + Linear	15	0.5714	15	**0.6194**	-8.06%
	BERT + GRU	12	**0.6166**	2	0.5116	18.16%
	BERT + SAN	6	0.5288	4	**0.5608**	-5.87%
	BERT + TFM	14	**0.5238**	1	0.5167	1.36%

Table 5.30: Results for the evaluation on sentiment, for the spontaneous speech samples from AWS Transcribe and the gold standard, with Self-Training.

For this, we first identify such samples. Since the unlabeled set for this study involved 4125 sentence samples for training and 1825 samples for test, this allowed us to examine every sentence sample out of the 1825 samples. Also, since these were from 151 videos, we skimmed through these videos to identify those that contained such speech instances. Out of 151 videos, we found 72 videos and 667 samples in total containing such instances. But during cross-examining sentiment labels for them, we found that the labels were only available in 32 videos and for 77 samples. we use these samples as the test set for sentiment evaluation.

Table 5.30 presents results with Self-Training implementations, while table 5.31 presents the results with Tri-Training. Similar to the previous section, the runs with Self-Training saw the ones with AWS Transcriptions outscoring its counterpart. For Tri-Training runs shown below, the ones with the gold standard transcriptions had better results.

with Tri-Training		with AWS Transcribe			with Gold Standard			
	Models	**Iteration**	**Models $h_1 - h_2 - h_3$**	**Macro-f1**	**Iteration**	**Models $h_1 - h_2 - h_3$**	**Macro-f1**	**Percentage Difference**
with REST	BERT + Linear	3	gru-linear-tfm	0.4449	1	gru-linear-tfm	**0.6666**	-39.89%
	BERT + GRU	1	gru-linear-tfm	0.3532	2	gru-linear-tfm	**0.4091**	-14.66%
	BERT + SAN	2	san-linear-tfm	0.3333	1	san-gru-tfm	**0.5185**	-43.48%
	BERT + TFM	2	gru-linear-tfm	**0.3571**	2	san-linear-tfm	0.3333	6.89%
with LAPTOP	BERT + Linear	2	san-linear-tfm	0.511	1	linear-gru-tfm	**0.6296**	-20.79%
	BERT + GRU	2	san-gru-tfm	0.4888	1	san-gru-tfm	**0.5237**	-6.89%
	BERT + SAN	2	san-linear-tfm	0.4539	1	san-linear-gru	**0.4993**	-9.53%
	BERT + TFM	1	san-linear-tfm	0.3	1	san-linear-tfm	**0.4126**	-31.60%

Table 5.31: Results for the evaluation on sentiment, for spontaneous speech samples from AWS Transcribe and the gold standard, with Tri-Training.

6 Conclusion

In our work on "Cross-domain Aspect-based Sentiment Analysis with Multimodal Sources", we investigated the effectiveness of Semi-supervised approaches to the Aspect-based Sentiment Analysis problem. We tackle the Aspect-based Sentiment Analysis as an E2E-ABSA problem which is a sequence labeling task tagging individual words for the possible aspect and sentiments.

First, to implement the baseline for our work, we created a computational model based on BERT. Specifically, we fine-tune the model *bert-base-uncased* with additional layers added to it to perform the task of sequence labeling. For this downstream task of labeling, we leveraged the work of Li et al. (2019c).

We then utilize two Semi-supervised methods, namely the Self-Training, and the Tri-Training, and then investigate their effectiveness to the E2E-ABSA. With Self-Training, the labeled data from the restaurant domain (REST), and the model BERT+TFM, we achieved the Macro-F1 increment of 0.1335 over the corresponding baseline. With Tri-Training and REST, we had two models, the BERT+Linear and BERT+GRU improving upon the respective baseline models by a Macro-F1 factor of 0.1132 and 0.086 respectively. These improvements with Tri-Training were also better than the scores from their respective Self-Training runs. These results show a constructive outcome for the application of Semi-supervised methods to the E2E-ABSA problem.

As part of the next research task, we investigated cross-domain analysis of the E2E-ABSA. With this evaluation being one of the key reasons to employ Self-Training and Tri-Training in our work, we segregated a part of unlabeled data into five domains. The predictions of Semi-supervised models on these samples are evaluated separately against the sentiment and aspect. The evaluation against sentiment had its best results with the samples belonging to Domain-3 (with the best Micro-F1 score of 0.95), while, the results for Domain-1, Domain-2, and Domain-4 were slightly lower (with the best Micro-F1 scores being 0.89, 0.85 and 0.8 respectively). The results for aspect detection, however, performed worse and while evaluating for performance across domains, the best results were had with the models from Self-Training runs. The best results were from the BERT+TFM model for REST and the BERT+GRU model for LAPTOP (with the F1 scores of 0.1666 and 0.1764 respectively). Even though the evaluation of aspect words was done on a small sample of 200 sentences, we conclude that the models work better in detecting sentiment than the aspect words, and in the case of the latter, there is scope for improvement.

For the third research task involving the evaluation of E2E-ABSA in the setting of erroneous transcriptions and spontaneous speech instances, we obtained transcriptions from an external Automatic Speech Recognition system. For this, we made use of Amazon Web Service's (AWS) Transcribe. On conducting concurrent experiments for E2E-ABSA but employing unlabeled samples from the AWS in one run and the gold standard in another, we then compared their results. The results from Self-Training had on more occasions, experiments conducted with gold standard samples outscoring the AWS transcribed counterpart. But the Tri-Training experiments had the runs from AWS transcribed samples accomplish better than the gold standard on more trials. We conclude here that even though there were considerable error rates when data samples were transcribed from an external system, it still retained the essential notion to derive sentiments and hence the lack of relatively lower results.

We then compared the inferences of sentiment on samples containing erroneous transcriptions alone. With Self-Training, the results from the AWS transcriptions, although erroneous, outscored the ones from the gold standard samples. But, with Tri-Training, the results with gold standard samples were better overall.

In the final part of the third research task, we evaluated sentiment on instances of spontaneous speech containing hesitation, pauses, and repetition. Here, the runs with Self-Training saw the ones with AWS Transcriptions outscoring its counterpart. For Tri-Training runs, the ones with the gold standard transcriptions had better results.

To conclude, through our study we found that there is scope for improving the E2E-ABSA performance with the application of Semi-supervised methods. While the Self-Training method has fewer advantages, the Tri-Training implementations led to improvements on more occasions than not. We also found that the domain adaptation from these methods was achievable, with room for improvement in the aspect detection front. Finally, the experiment results on erroneous transcriptions did not lead to an inferior E2E-ABSA performance although the inference of sentiment alone would work better with just the Self-Training method.

6.1 Future works

In our work, along with inspecting the effectiveness of Semi-supervised learning, we also investigated the domain adaptation capabilities of these methods. With the reasonably good performance of the classical Tri-Training method, an investigation into its variants such as Tri-training with Disagreement (Søgaard, 2010), and Asymmetric Tri-Training (Saito et al., 2017) could be noteworthy. Also, with the neutral sentiment-heavy samples of CMU-MOSEI, experiments with sentiment-balanced labeled data might lead to promising results, both in terms of detecting sentiment and aspect terms.

Bibliography

Francisca Adoma Acheampong, Henry Nunoo-Mensah, and Wenyu Chen. Transformer models for text-based emotion detection: a review of bert-based approaches. *Artificial Intelligence Review*, 54(8):5789–5829, 2021.

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. * sem 2013 shared task: Semantic textual similarity. In *Second joint conference on lexical and computational semantics (* SEM), volume 1: proceedings of the Main conference and the shared task: semantic textual similarity*, pages 32–43, 2013.

Mohammad Ehsan Basiri, Shahla Nemati, Moloud Abdar, Erik Cambria, and U Rajendra Acharya. Abcdm: An attention-based bidirectional cnn-rnn deep model for sentiment analysis. *Future Generation Computer Systems*, 115:279–294, 2021.

David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N Chang, Sungbok Lee, and Shrikanth S Narayanan. Iemocap: Interactive emotional dyadic motion capture database. *Language resources and evaluation*, 42(4):335–359, 2008.

Erik Cambria, Jie Fu, Federica Bisio, and Soujanya Poria. Affectivespace 2: Enabling affective intuition for concept-level sentiment analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

Erik Cambria, Soujanya Poria, Rajiv Bajpai, and Björn Schuller. Senticnet 4: A semantic resource for sentiment analysis based on conceptual primitives. In *Proceedings of COLING 2016, the 26th international conference on computational linguistics: Technical papers*, pages 2666–2677, 2016.

Bo Chen, Wai Lam, Ivor Tsang, and Tak-Lam Wong. Extracting discriminative concepts for domain adaptation in text mining. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 179–188, 2009.

Shaowei Chen, Jie Liu, Yu Wang, Wenzheng Zhang, and Ziming Chi. Synchronous double-channel recurrent network for aspect-opinion pair extraction. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 6515–6524, 2020.

Zhiyuan Chen, Arjun Mukherjee, Bing Liu, Meichun Hsu, Malu Castellanos, and Riddhiman Ghosh. Exploiting domain knowledge in aspect extraction. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1655–1667, 2013.

Zhuang Chen and Tieyun Qian. Enhancing aspect term extraction with soft prototypes. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2107–2117, 2020a.

Zhuang Chen and Tieyun Qian. Relation-aware collaborative learning for unified aspect-based sentiment analysis. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3685–3694, 2020b.

Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

Stephen Della Pietra, Vincent Della Pietra, and John Lafferty. Inducing features of random fields. *IEEE transactions on pattern analysis and machine intelligence*, 19(4):380–393, 1997.

Julien Deonna and Fabrice Teroni. *The emotions: A philosophical introduction.* Routledge, 2012.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Mauro Dragoni and Giulio Petrucci. A neural word embeddings approach for multi-domain sentiment analysis. *IEEE Transactions on Affective Computing*, 8(4):457–470, 2017.

Paul Ekman, Wallace V Freisen, and Sonia Ancoli. Facial signs of emotional experience. *Journal of personality and social psychology*, 39(6):1125, 1980.

Zhifang Fan, Zhen Wu, Xinyu Dai, Shujian Huang, and Jiajun Chen. Target-oriented opinion words extraction with target-fused neural sequence labeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2509–2518, 2019.

Andrea Gasparetto, Matteo Marcuzzo, Alessandro Zangari, and Andrea Albarelli. A survey on text classification algorithms: From text to predictions. *Information*, 13(2):83, 2022.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. An unsupervised neural attention model for aspect extraction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 388–397, 2017.

Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. An interactive multi-task learning network for end-to-end aspect-based sentiment analysis. *arXiv preprint arXiv:1906.06906*, 2019.

Ruining He and Julian McAuley. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*, pages 507–517, 2016.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

Badr Hssina, Abdelkarim Merbouha, Hanane Ezzikouri, and Mohammed Erritali. A comparative study of decision tree id3 and c4. 5. *International Journal of Advanced Computer Science and Applications*, 4(2):13–19, 2014.

Minghao Hu, Yuxing Peng, Zhen Huang, Dongsheng Li, and Yiwei Lv. Open-domain targeted sentiment analysis via span-based extraction and classification. *arXiv preprint arXiv:1906.03820*, 2019.

Amir Hussain and Erik Cambria. Semi-supervised learning for big social data analysis. *Neurocomputing*, 275:1662–1673, 2018.

Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. Target-dependent twitter sentiment classification. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, pages 151–160, 2011.

Yohan Jo and Alice H Oh. Aspect and sentiment unification model for online review analysis. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 815–824, 2011.

Hanhoon Kang, Seong Joon Yoo, and Dongil Han. Senti-lexicon and improved naïve bayes algorithms for sentiment analysis of restaurant reviews. *Expert Systems with Applications*, 39 (5):6000–6010, 2012.

Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura Barnes, and Donald Brown. Text classification algorithms: A survey. *Information*, 10(4):150, 2019.

Raymond YK Lau, Chunping Li, and Stephen SY Liao. Social analytics: Learning fuzzy product ontologies for aspect-oriented sentiment analysis. *Decision Support Systems*, 65:80–94, 2014.

Kun Li, Chengbo Chen, Xiaojun Quan, Qing Ling, and Yan Song. Conditional augmentation for aspect term extraction via masked sequence-to-sequence generation. *arXiv preprint arXiv:2004.14769*, 2020a.

Ning Li, Chi-Yin Chow, and Jia-Dong Zhang. Seeded-btm: enabling biterm topic model with seeds for product aspect mining. In *2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, pages 2751–2758. IEEE, 2019a.

Ning Li, Chi-Yin Chow, and Jia-Dong Zhang. Seml: A semi-supervised multi-task learning framework for aspect-based sentiment analysis. *IEEE Access*, 8:189287–189297, 2020b.

Xin Li, Lidong Bing, Piji Li, and Wai Lam. A unified model for opinion target extraction and target sentiment prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6714–6721, 2019b.

Xin Li, Lidong Bing, Wenxuan Zhang, and Wai Lam. Exploiting bert for end-to-end aspect-based sentiment analysis. *arXiv preprint arXiv:1910.00883*, 2019c.

Yung-Ming Li and Tsung-Ying Li. Deriving market intelligence from microblogs. *Decision Support Systems*, 55(1):206–217, 2013.

Yunlong Liang, Fandong Meng, Jinchao Zhang, Yufeng Chen, Jinan Xu, and Jie Zhou. An iterative multi-knowledge transfer network for aspect-based sentiment analysis. *arXiv preprint arXiv:2004.01935*, 2020.

Pengfei Liu, Shafiq Joty, and Helen Meng. Fine-grained opinion mining with recurrent neural networks and word embeddings. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1433–1443, 2015.

Huaishao Luo, Tianrui Li, Bing Liu, and Junbo Zhang. Doer: dual cross-shared rnn for aspect term-polarity co-extraction. *arXiv preprint arXiv:1906.01794*, 2019.

Huaishao Luo, Lei Ji, Tianrui Li, Nan Duan, and Daxin Jiang. Grace: Gradient harmonized and cascaded labeling for aspect-based sentiment analysis. *arXiv preprint arXiv:2009.10557*, 2020.

Yukun Ma, Haiyun Peng, Tahir Khan, Erik Cambria, and Amir Hussain. Sentic lstm: a hybrid network for targeted aspect-based sentiment analysis. *Cognitive Computation*, 10(4):639–650, 2018.

David McClosky, Eugene Charniak, and Mark Johnson. Effective self-training for parsing. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 152–159, 2006.

Zhengjie Miao, Yuliang Li, Xiaolan Wang, and Wang-Chiew Tan. Snippext: Semi-supervised opinion mining with augmented data. In *Proceedings of The Web Conference 2020*, pages 617–628, 2020.

Louis-Philippe Morency, Rada Mihalcea, and Payal Doshi. Towards multimodal sentiment analysis: Harvesting opinions from the web. In *Proceedings of the 13th international conference on multimodal interfaces*, pages 169–176, 2011.

Arjun Mukherjee and Bing Liu. Aspect extraction through semi-supervised modeling. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 339–348, 2012.

Ambreen Nazir, Yuan Rao, Lianwei Wu, and Ling Sun. Issues and challenges of aspect-based sentiment analysis: a comprehensive survey. *IEEE Transactions on Affective Computing*, 2020.

Harsh H Patel and Purvi Prajapati. Study and analysis of decision tree based classification algorithms. *International Journal of Computer Sciences and Engineering*, 6(10):74–78, 2018.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. SemEval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35, Dublin, Ireland, August 2014. Association for Computational Linguistics. doi: 10.3115/v1/S14-2004. URL `https://aclanthology.org/S14-2004`.

Maria Pontiki, Dimitrios Galanis, Harris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. Semeval-2015 task 12: Aspect based sentiment analysis. In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, pages 486–495, 2015.

Maria Pontiki, Dimitrios Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad Al-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, et al. Semeval-2016 task 5: Aspect based sentiment analysis. In *International workshop on semantic evaluation*, pages 19–30, 2016.

Soujanya Poria, Iti Chaturvedi, Erik Cambria, and Amir Hussain. Convolutional mkl based multimodal emotion recognition and sentiment analysis. In *2016 IEEE 16th international conference on data mining (ICDM)*, pages 439–448. IEEE, 2016.

Guozheng Rao, Weihang Huang, Zhiyong Feng, and Qiong Cong. Lstm with sentence representations for document-level sentiment classification. *Neurocomputing*, 308:49–57, 2018.

Roi Reichart and Ari Rappoport. Self-training for enhancement and domain adaptation of statistical parsers trained on small datasets. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 616–623, 2007.

R Revathy and R Lawrance. Comparative analysis of c4. 5 and c5. 0 algorithms on crop pest data. *International Journal of Innovative Research in Computer and Communication Engineering*, 5(1):50–58, 2017.

Sebastian Ruder and Barbara Plank. Strong baselines for neural semi-supervised learning under domain shift. *arXiv preprint arXiv:1804.09530*, 2018.

Kenji Sagae. Self-training without reranking for parser domain adaptation and its impact on semantic role labeling. In *Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing*, pages 37–44, 2010.

Kuniaki Saito, Yoshitaka Ushiku, and Tatsuya Harada. Asymmetric tri-training for unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 2988–2997. PMLR, 2017.

Erik F Sang and Jorn Veenstra. Representing text chunks. *arXiv preprint cs/9907006*, 1999.

Jürgen Schmidhuber, Sepp Hochreiter, et al. Long short-term memory. *Neural Comput*, 9(8): 1735–1780, 1997.

Kim Schouten and Flavius Frasincar. Survey on aspect-level sentiment analysis. *IEEE Transactions on Knowledge and Data Engineering*, 28(3):813–830, 2015.

Mrityunjay Singh, Amit Kumar Jakhar, and Shivam Pandey. Sentiment analysis on the impact of coronavirus in social life using the bert model. *Social Network Analysis and Mining*, 11(1): 1–11, 2021.

Sonia Singh and Priyanka Gupta. Comparative study id3, cart and c4. 5 decision tree algorithm: a survey. *International Journal of Advanced Information Science and Technology (IJAIST)*, 27(27):97–103, 2014.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.

Anders Søgaard. Simple semi-supervised training of part-of-speech taggers. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 205–208, 2010.

Mohammad Soleymani, David Garcia, Brendan Jou, Björn Schuller, Shih-Fu Chang, and Maja Pantic. A survey of multimodal sentiment analysis. *Image and Vision Computing*, 65:3–14, 2017.

Lukas Stappen, Alice Baird, Lea Schumann, and Björn Schuller. The multimodal sentiment analysis in car reviews (muse-car) dataset: Collection, insights and improvements. *arXiv preprint arXiv:2101.06053*, 2021.

Duyu Tang, Furu Wei, Bing Qin, Ting Liu, and Ming Zhou. Coooolll: A deep learning system for twitter sentiment classification. In *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*, pages 208–212, 2014.

Alper Kursat Uysal and Yi Lu Murphey. Sentiment classification: Feature selection based approaches versus deep learning. In *2017 IEEE International Conference on Computer and Information Technology (CIT)*, pages 23–30. IEEE, 2017.

Vincent Van Asch and Walter Daelemans. Predicting the effectiveness of self-training: Application to sentiment classification. *arXiv preprint arXiv:1601.03288*, 2016.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

Peerapon Vateekul and Thanabhat Koomsubha. A study of sentiment analysis using deep learning techniques on thai twitter data. In *2016 13th International joint conference on computer science and software engineering (JCSSE)*, pages 1–6. IEEE, 2016.

Feixiang Wang, Man Lan, and Wenting Wang. Towards a one-stop solution to both aspect extraction and sentiment analysis tasks with neural multi-task learning. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2018.

Qianlong Wang, Zhiyuan Wen, Qin Zhao, Min Yang, and Ruifeng Xu. Progressive self-training with discriminator for aspect term extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 257–268, 2021.

Mayur Wankhade, Annavarapu Chandra Sekhara Rao, and Chaitanya Kulkarni. A survey on sentiment analysis methods, applications, and challenges. *Artificial Intelligence Review*, pages 1–50, 2022.

Martin Wöllmer, Felix Weninger, Tobias Knaup, Björn Schuller, Congkai Sun, Kenji Sagae, and Louis-Philippe Morency. Youtube movie reviews: Sentiment analysis in an audio-visual context. *IEEE Intelligent Systems*, 28(3):46–53, 2013.

Hu Xu, Bing Liu, Lei Shu, and Philip S Yu. Double embeddings and cnn-based sequence labeling for aspect extraction. *arXiv preprint arXiv:1805.04601*, 2018.

Lu Xu, Hao Li, Wei Lu, and Lidong Bing. Position-aware tagging for aspect sentiment triplet extraction. *arXiv preprint arXiv:2010.02609*, 2020.

David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *33rd annual meeting of the association for computational linguistics*, pages 189–196, 1995.

Qussai Yaseen et al. Spam email detection using deep learning techniques. *Procedia Computer Science*, 184:853–858, 2021.

yelp,2014. Y.dataset.yelp dataset challenge. available: https://www.yelp.com/dataset/challenge. Sunday Mail, Mar. 5, 2019.

Zhigang Yuan, Sixing Wu, Fangzhao Wu, Junxin Liu, and Yongfeng Huang. Domain attention model for multi-domain sentiment classification. *Knowledge-Based Systems*, 155:1–10, 2018.

Amir Zadeh, Rowan Zellers, Eli Pincus, and Louis-Philippe Morency. Mosi: multimodal corpus of sentiment intensity and subjectivity analysis in online opinion videos. *arXiv preprint arXiv:1606.06259*, 2016.

AmirAli Bagher Zadeh, Paul Pu Liang, Soujanya Poria, Erik Cambria, and Louis-Philippe Morency. Multimodal language analysis in the wild: Cmu-mosei dataset and interpretable dynamic fusion graph. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2236–2246, 2018.

Wenxuan Zhang, Xin Li, Yang Deng, Lidong Bing, and Wai Lam. A survey on aspect-based sentiment analysis: Tasks, methods, and challenges. *arXiv preprint arXiv:2203.01054*, 2022.

He Zhao, Longtao Huang, Rong Zhang, Quan Lu, and Hui Xue. Spanmlt: A span-based multi-task learning framework for pair-wise aspect and opinion terms extraction. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 3239–3248, 2020.

Xin Zhao, Jing Jiang, Hongfei Yan, and Xiaoming Li. Jointly modeling aspects and opinions with a maxent-lda hybrid. ACL, 2010.

Zhi-Hua Zhou and Ming Li. Tri-training: Exploiting unlabeled data using three classifiers. *IEEE Transactions on knowledge and Data Engineering*, 17(11):1529–1541, 2005.

Xiaojin Zhu and Andrew B Goldberg. Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, 3(1):1–130, 2009.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27, 2015.