A Weak Supervision Approach for Few-Shot Aspect Based Sentiment Analysis

Robert Vacareanu[2,3*] **Siddharth Varia**[1] **Kishaloy Halder**[1] **Shuai Wang**[1]
Giovanni Paolini[1] **Neha Anna John**[1] **Miguel Ballesteros**[1] **Smaranda Muresan**[1]
[1]AWS AI Labs
[2]University of Arizona, Tucson, AZ, USA
[3]Technical University of Cluj-Napoca, Romania
{siddhvar, kishaloh, wshui, paoling, nehajohn, ballemig, smaranm}@amazon.com
rvacareanu@arizona.edu

Abstract

We explore how weak supervision on abundant unlabeled data can be leveraged to improve few-shot performance in aspect-based sentiment analysis (ABSA) tasks. We propose a pipeline approach to construct a noisy ABSA dataset, and we use it to adapt a pre-trained sequence-to-sequence model to the ABSA tasks. We test the resulting model on three widely used ABSA datasets, before and after fine-tuning. Our proposed method preserves the full fine-tuning performance while showing significant improvements (15.84% absolute F1) in the few-shot learning scenario for the harder tasks. In zero-shot (i.e., without fine-tuning), our method outperforms the previous state of the art on the aspect extraction sentiment classification (AESC) task and is, additionally, capable of performing the harder aspect sentiment triplet extraction (ASTE) task.

1 Introduction

Aspect Based Sentiment Analysis (ABSA) is a fine-grained variant of sentiment analysis (Hu and Liu, 2004; Pontiki et al., 2014, 2015, 2016; Zhang et al., 2021a; Shu et al., 2022; Zhang et al., 2022), where the task is to predict the sentiment expressed towards an entity or a certain aspect of an entity, instead of just the sentence-level sentiment (*e.g.,* traditional sentiment analysis tasks (Socher et al., 2013; dos Santos and de C. Gatti, 2014)).

For illustration, for a review *The pizza was great, but the service was terrible*, a sentence-level sentiment analysis model might identify the sentiment as *neutral*. The need for ABSA stems from such complex interactions between the target and the polarity of the sentiment (Pontiki et al., 2014). An ABSA model has to identify the sentiment towards *pizza* as *positive*, and *service* as *negative*, for a holistic understanding of the text. Furthermore, ABSA tasks can include the identification of the opinion terms (i.e. *great*, *terrible*), and the aspect categories (i.e. `FOOD`, `SERVICE`) (Zhang et al., 2021a).

Although traditionally considered as a structured prediction task in the ABSA literature, recent works have shown how sequence-to-sequence (seq-to-seq) models can be effective in these tasks with a generative approach (Yan et al., 2021; Zhang et al., 2021a). Such approaches leverage the knowledge gained from one task to seamlessly perform well in another. As such, we build upon the Instruction Tuning with Multi-Task Learning approach (Varia et al., 2022) and address the following five ABSA tasks: (i) Aspect-term Extraction (AE), (ii) Aspect-term Extraction and Sentiment Classification (AESC), (iii) Target Aspect Sentiment Detection (TASD), (iv) Aspect Sentiment Triplet Extraction (ASTE), and (v) Aspect Sentiment Quadruple Prediction (ASQP).

Sentence-level sentiment annotations are comparatively cheaper and are available at scale through automated proxies (*e.g.,* ★ or ★★ become *negative*, ★★★★ or ★★★★★ become *positive*, in the Amazon/Yelp review corpus (Zhang et al., 2015b)). On the contrary, ABSA requires understanding at sub-sentence level with multiple words or phrases being related to each other, making it prohibitively costly to annotate at scale.[1] However, the abundance of generic review data presents a promising opportunity to improve the performance of a pre-trained language model (PLM) beyond simply fine-tuning it on the small annotated ABSA corpora.

Towards this end, we first construct a noisily annotated ABSA corpus out of generic customer review data without any direct supervision. We utilize this noisy corpus to pre-train a seq-to-seq model on multiple ABSA tasks. We show that such

[*]Work done during internship at AWS AI Labs

[1]This is evident from the corpus size of 2.1k vs 700k for Rest16 and Yelp-full, respectively.

models are capable of learning in zero/few-shot in final downstream ABSA tasks. Our contributions are the following: (i) We propose a weakly supervised method to obtain annotations for three out of the five ABSA tasks explored in the literature; (ii) We introduce a pre-training step to improve the few-shot performance on the downstream task of PLMs; (iii) We comprehensively evaluate our proposed method in three scenarios (full fine-tuning, few-shot, and zero-shot learning), yielding as much as 15.84% F1 improvement over the SOTA baselines. We release the sources to create the few-shot benchmarking datasets[2].

2 Related Work

Aspect-Based Sentiment Analysis has received tremendous attention in the past years (Tulkens and van Cranenburgh, 2020; Zhang et al., 2021a; Shu et al., 2022; Zhang et al., 2022), either handling single tasks, such as aspect term extraction (He et al., 2017; Liu et al., 2015; Tulkens and van Cranenburgh, 2020), aspect category detection (Tulkens and van Cranenburgh, 2020), aspect sentiment classification (Vo and Zhang, 2015; Xu et al., 2019; Li et al., 2021; Wang et al., 2021), or handling compound tasks (Zhang et al., 2015a; Yu et al., 2021; Xu et al., 2020; Zhang et al., 2021a). For the latter group, it typically includes either a pipeline approach (Peng et al., 2020; Yan et al., 2021) or an end-to-end (E2E) approach (Xu et al., 2020; Zhang et al., 2021a,b).

In the pipeline approach the final prediction is constructed using the output of multiple components. The disadvantage of such models is that the error is propagated throughout the system (Zhang et al., 2022).

In the E2E approach, the model learns the interactions jointly between the multiple prediction tasks, which is believed to improve the final performance (Xu et al., 2020; Zhang et al., 2022). Our proposed approach falls in this category. Typical E2E approaches include: (i) treating it as a token classification task (Xu et al., 2019; Shu et al., 2019; Xu et al., 2020), (ii) framing it as a machine reading comprehension task (Chen et al., 2021; Liu et al., 2022), natural language inference task (Shu et al., 2022), or as a language generation task (Zhang et al., 2021b; Yan et al., 2021; Zhang et al., 2021a; Varia et al., 2022).

Our proposed approach treats the ABSA tasks as a generation task, similar to (Zhang et al., 2021a; Varia et al., 2022). We build upon the paradigm called Instruction Tuning with in Multi-Task Learning (IT-MTL), introduced in (Varia et al., 2022), resulting in a single model capable of handling different ABSA tasks. However, none of these methods takes advantage of the vast amount of review data available, other than just pre-training on them with some generic language modeling objectives.

3 Method

We introduce an additional step in the classical `pretrain → finetune` approach (Howard and Ruder, 2018; Devlin et al., 2019; Raffel et al., 2020), transforming it into `pretrain →` **`Noisy ABSA Pre-Training`** `(NAPT) → finetune` for ABSA. We propose an approach for building a weakly annotated dataset for the intermediate NAPT step. We use this noisy dataset to enhance the knowledge of a pretrained model with the intuition that exposing the model to tasks which are well aligned with the final downstream task, improves the performance. We then consider this as the backbone base model, and finetune it on the downstream task as usual. Our proposed approach is applicable to any generic seq-to-seq model.

3.1 Dataset Construction

The first step in our proposed method is to weakly annotated a dataset without any direct supervision.[3] Our proposed approach annotates a dataset with tuples of the form aspect-terms, opinion-terms, and sentiment polarity. We follow a pipeline approach as shown in Table 1(Xu et al., 2013; Zhang et al., 2022), but without using any direct ABSA supervision. We describe each step in greater detail next.

3.1.1 Aspect-term Extraction

The first step in our proposed dataset creation procedure is aspect-term extraction. We use `spacy` tokenizer to obtain POS tags and then consider 20% of the most frequent nouns in the text. These nouns serve as candidate aspect terms. We note that this method implicitly assumes that dataset D consists of a single domain. Nevertheless, this is a small assumption as the reviews are typically directed towards a product of a known category (He and McAuley, 2016; Zhang et al., 2015b). We extend

[2] `https://github.com/robertvacareanu/NoisyABSAPreTraining`

[3] We use models which were trained on different tasks, but no model has seen any aspect-based sentiment analysis data.

Sentence: *The pizza was great, but the service was terrible.*		
Step	**Heuristic / Method**	**Resulting Annotations**
#1	Extract frequent nouns as Aspect-terms	pizza, service
#2	Extract matches with an opinion lexicon as Opinion-terms	great, terrible
#3	Predict entailment of form *{aspect} is {opinion}* for every aspect, opinion combinations using a pre-trained NLI model	<pizza, great>, <service, terrible>
#4	Classify documents of form *{aspect} is {opinion}* with a pre-trained sentiment analysis model	<pizza, great, positive>, <service, terrible, negative>

Table 1: A step-by-step illustration of our noisy dataset construction pipeline. It follows a pipeline approach, and yields <aspect, opinion, sentiment> triplets in the end for each sentence in a generic review corpus.

this method to multi-word aspect terms by considering collocations of length ≤ 4 filtered by their POS tags. For example, we allow bigrams of the form `NN-NN` like *chicken breast* (*cf* Table 16 for all patterns used). Finally, we filter out the sentences from which no aspect term was extracted.

3.1.2 Opinion-term Extraction

The second step in our proposed algorithm is opinion term extraction. We take a lexicon-based approach to opinion extraction (Ding et al., 2008; Kanayama and Nasukawa, 2006; Hu and Liu, 2004). In particular, we use the opinion lexicon from (Hu and Liu, 2004) and perform word matching on the target text. If negations *e.g., no* or *not* appear before the opinion word, we include it in the final extraction as well. We filter out the sentences from which no opinion term was extracted.

3.1.3 Linking Opinion-terms with Aspect-terms

So far the resulting dataset consists of noisy aspect, and opinion terms, but without the association between them. For example, for a sentence such as *The pizza was great , but the service was terrible.*, the proposed algorithm would extract *pizza* and *service* as the aspect terms and *great* and *terrible* as the opinion terms, respectively. But at this point we do not know that *great* refers to *pizza* and *terrible* refers to *service*. We reformulate this problem as a natural language inference problem (Dagan et al., 2005; Shu et al., 2022). We use an `MPNet`[4] model (Song et al., 2020) and construct artificial sentences to determine which opinion-term refers to which aspect-term. More precisely, we construct sentences such as `<aspect-term> is <opinion-term>`, for each aspect- and opinion-term.[5] Then, we use the original sentence (e.g. *The pizza was great , but the service was terrible.*) as the premise and our artificially constructed sentence as the hypothesis (e.g. *pizza is great*). We interpret a high entailment score (≥ 0.75) as evidence that the opinion term refers to that particular aspect term. We discard aspect- and opinion-term pairs where the entailment score was below the threshold.

Alternative Approach: We consider an alternate approach where the linking is based on constituency-parse rules which turns out disadvantageous. Constituency parsing is considerably slower and the rules are non-trivial to formulate.

3.1.4 Sentiment Extraction

The last step in our proposed dataset creation method is to add the sentiment (Hu and Liu, 2004) to each `<aspect-term, opinion-term>` tuple. We use a sentence-level classifier on top of artificially constructed sentences (Sanh et al., 2019). For example, for a tuple such as <pizza, great>, we feed the sentence *pizza is great* through a sentence-level sentiment classifier.[6] Then, we label the <aspect term, opinion term> tuple with the sentiment prediction if the model's confidence is above a certain threshold (≥ 0.75), otherwise we discard the tuple. At the end of this step, for the sentence *The pizza was great , but the service was terrible.* we have the following `<aspect-term, opinion-term, sentiment>` noisy annotations: <pizza, great, positive>, <service, terrible, negative>. We consider an alternative for this step using the sentiments associated in the opinion lexicon, but a classifier allows for confidence filtering.

Throughout our proposed dataset creation process we use external resources, such an opinion lexicon, an NLI model and a sentence-level sentiment classifier. However, these resources do not consume any annotated ABSA data by any means.

[4]huggingface.co/symanto/mpnet-base-snli-mnli

[5]We relax strict grammatical correctness *e.g.,* the formulation might result in *burgers is great* instead of *burgers are great*).

[6]huggingface.co/distilbert-base-uncased-finetuned-sst-2-english

3.2 Noisy ABSA Pre-training (NAPT)

The phase consists of exposing the model to tasks that are more aligned with the final downstream task, *i.e.,* ABSA in our case. We factorize the triplets from the noisy dataset into five separate but overlapping tasks: (i) aspect-term extraction, (ii) opinion-term extraction, (iii) aspect-term and opinion-term extraction, (iv) aspect-term extraction and sentiment prediction, and (v) aspect-term extraction, opinion-term extraction and sentiment prediction. Note that there exists a correspondence between our NAPT tasks and classical ABSA tasks: tasks (i), (iv) and (v) correspond to Aspect Extraction (AE), Aspect Extraction Sentiment Classification (AESC), and Aspect Sentiment Triplet Extraction (ASTE), respectively. We use the noisy ABSA dataset to pre-train the base model. We train the model parameters in a multi-task learning framework (*cf* Figure 1) using instruction tuning with a diverse set of instructions (Sanh et al., 2022). At the end of NAPT, the resulting model is imbued with the capability of performing multiple ABSA tasks. This can serve as a drop-in replacement to the off-the-shelf pre-trained checkpoints that are widely used in the generative ABSA literature.

3.2.1 Addressing Overfitting

The primary goal of our proposed NAPT phase is to *enhance* the pre-trained model while retaining existing knowledge from pre-training objectives, in other words, avoiding catastrophic forgetting and overfitting. We achieve this in a few different ways. First, instead of just randomly splitting the data into train/validation, we split the extracted aspect- and opinion-terms into two disjoint sets, favoring novel aspect- and opinion term constructions in the validation partition. We observe this split definition to be necessary to prevent overfitting of the base model. Additionally, we invoke three types of regularization:

- **Standard weight decay:** we add a standard ℓ^2 regularization term to the loss function.
- **Tuple Dropout:** we apply dropout over the tuples that the model is trained to extract to prevent it from overfitting to the noisy annotations. We randomly dropped 50% of the tuples from prediction targets of the seq-to-seq model.
- **Biased weight decay:** we use a biased variant of weight decay to prevent the parameters from diverging considerably from the initialization point, akin to (Kirkpatrick et al., 2017). Towards this, we use the ℓ^2 norm over the difference between the current (θ) and the initial weights of the model (θ_{init}), and add it to the loss.

Our final loss function ($\mathcal{L}$) is:

$$\mathcal{L} = CE_{loss} + \alpha \cdot \ell^2(\theta - \theta_{init}) + \beta \cdot \ell^2(\theta). \quad (1)$$

where α and β are hyperparameters, and CE_{loss} denotes the standard cross-entropy loss.

4 Experiments

We compare against state-of-the-art methods on three widely used ABSA datasets. We evaluate in three scenarios: (i) k-shot learning: where the model has access to at least k examples of each class, (ii) zero-shot evaluation: where the model has not seen any example at all from the gold-annotated ABSA data, and (iii) full-training: where the model has access to the complete gold-standard training data,

4.1 Experimental Setup

In all our experiments, we use `T5` (Raffel et al., 2020), particularly `t5-base` as the pre-trained seq-to-sed model, which has ~ 220M parameters. We experiment with `t5-large` as well to explore the impact of model size on the downstream performance (*cf* Appendix B). We use the standard evaluation metrics as previous work, which is F1 score over the exact match of the tuples. For zero-shot, we use the same evaluation procedure as (Shu et al., 2022), which is token-level F1 score.

We use a random subset of *Amazon Electronics* (He and McAuley, 2016), and *Yelp* reviews (Zhang et al., 2015b) to create our noisy-annotated dataset.[7] We split the reviews with ≥ 3 sentences using a sentence tokenizer. We split the noisy dataset into train/validation split. We enforce that there is no overlap in terms of aspect-terms between the train/validation splits. This results in approximately 190k examples for training and 12.5k examples for validation.

We repeat each experiment with 5 different random seeds. Additionally, we repeat the noisy ABSA pre-training step with 3 different random seeds. As a result, the numbers corresponding to our proposed method (i.e. the ones with `-APT`) represent an average of $5 \times 3 = 15$ runs, and all the

[7] 100K reviews from Amazon, and YELP each are used.

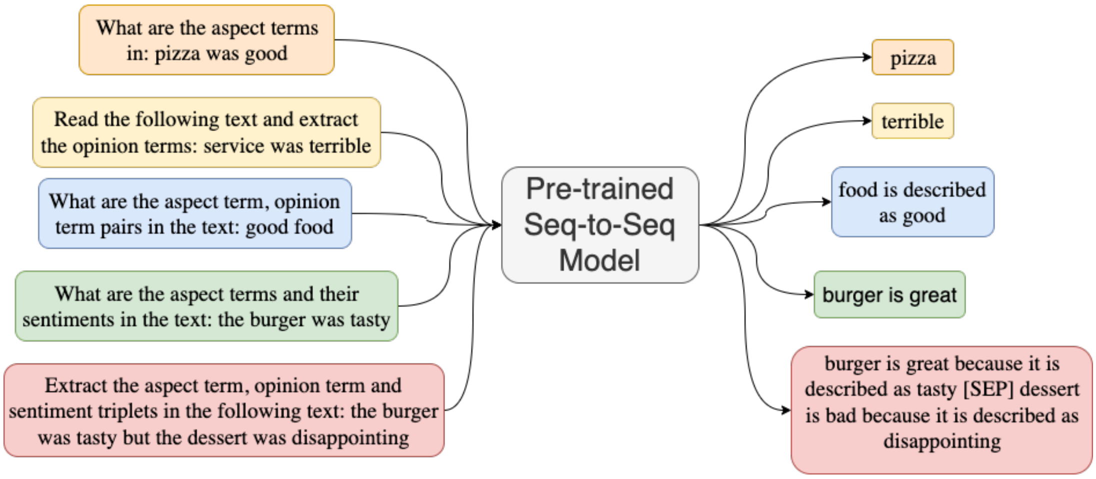


Figure 1: Overview of our proposed Noisy ABSA Pre-Training (NAPT). We start from a pretrained language model and extend its capabilities by instruction tuning it in a multi-task learning fashion. We use 5 different yet related tasks for the proposed NAPT step. The tasks we use are: (i) aspect-term extraction, (ii) opinion-term extraction, (iii) aspect-term extraction and opinion-term extraction, (iv) aspect term extraction and sentiment classification, and (v) aspect-term extraction, opinion-term extraction, and sentiment classification. This step results in a model capable of performing multiple ABSA tasks.

other numbers represent an average of 5 runs. We report the mean and (sample) standard deviation.

We present the results on the Aspect Sentiment Triplet Extraction (ASTE) and Aspect-term Extraction and Sentiment Classification (AESC) tasks available in all the datasets we use for evaluation.[8]

4.2 Datasets

We use three popular datasets for aspect-based sentiment analysis: REST15, REST16 and LAP14 (Pontiki et al., 2014, 2015, 2016), which cover two domains: restaurant and laptop, respectively. In particular, we use the version released by Zhang et al.. For k-shot, we use the same splits as (Varia et al., 2022) to ensure a fair comparison. Specifically, the k-shot datasets were created by sampling k examples for each attribute. The attributes are *aspect category*, and *sentiment* for restaurant, and laptop respectively.

4.3 Baselines

Since we introduce the NAPT step and build upon the existing Instruction Tuning with Multi-Task Learning (`IT-MTL`) paradigm, we refer to our proposed method as `IT-MTL-NAPT`. We compare this with standard fine-tuning based approaches that generally show strong performance in ABSA tasks *i.e.,* (i) text-only (`Text`), where we give the model the text review and train it to predict the gold text (Zhang et al., 2021a), (ii) instruction tuning (`IT`) and (iii) instruction tuning + multi-task learning, as per (Varia et al., 2022) (`IT-MTL`).

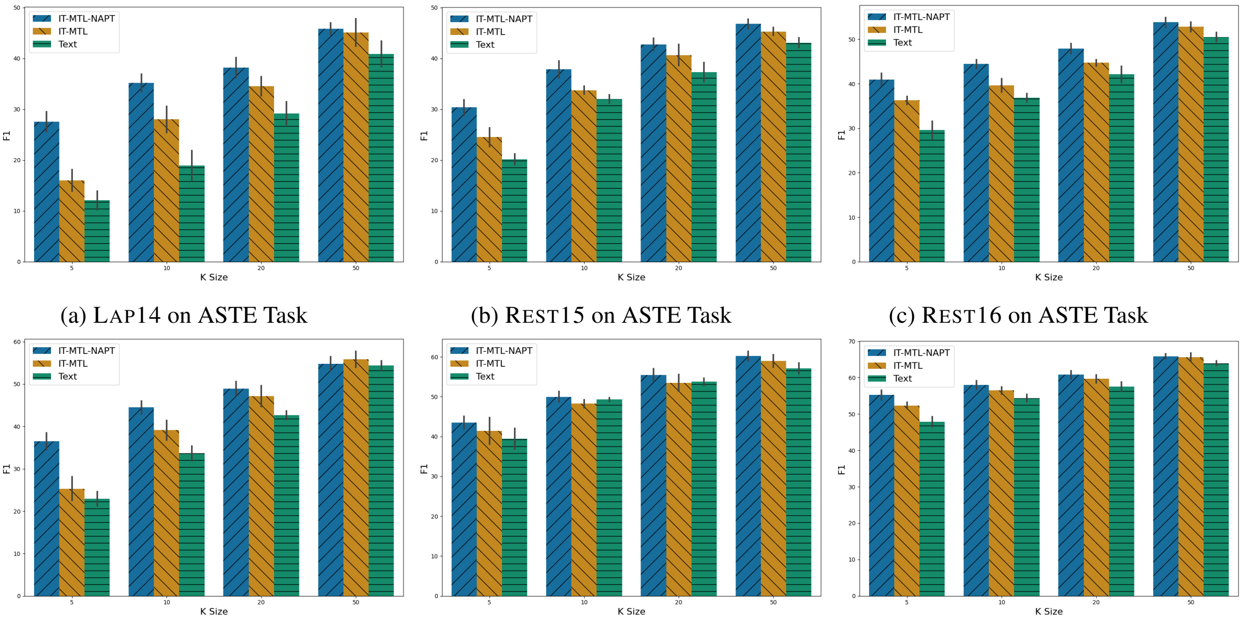
To succinctly show the effectiveness of proposed NAPT, we keep another baseline where a seq-to-seq model is further pre-trained with in-domain data using the same objective as that of `t5` *i.e.,* span prediction. We call it `IT-MTL-ID`.[9] The in-domain data is essentially the same as that of the NAPT corpus, but without the noisy annotations.

4.4 K-Shot Learning

Next, we compare between the two approaches in k-shot learning scenarios. We summarize our results in Figure 2. `IT`, and `IT-MTL-ID` perform similarly with the other baselines, so we skip them for clarity. We include all our results in Appendix B.2. First we observe that, our proposed method outperforms the baselines across all datasets in all k-shot scenarios, yielding as much as 15.84% F1 points (i.e. from 13.04%F1 to 28.88%F1) of improvement. Second, the performance improvement increases as the number of examples decrease, with the biggest improvement being in the k=5 case. This is expected because with the growing number of examples, all models are able to learn the task better. When using the full dataset, as we see in Table 3, both the proposed model and the baseline performances converge. Additionally, we observe that our proposed method brings the larger improvements on the harder tasks, as it gets difficult for the

[8]Results for **all** tasks are in Tables 10,11,12, and 7,8,9 for k-shot and full training respectively.

[9]As in, In-Domain (ID) pre-training occurs along with IT-MTL.



(a) LAP14 on ASTE Task (b) REST15 on ASTE Task (c) REST16 on ASTE Task

(d) LAP14 on AESC Task (e) REST15 on AESC Task (f) REST16 on AESC Task

Figure 2: Performance Comparison between our proposed method (`IT-MTL-NAPT`) and two baselines over 3 datasets on on the Aspect Sentiment Triplet Extraction (ASTE), Aspect-term Extraction and Sentiment Classification (AESC) tasks in top, and bottom rows respectively. We note that our proposed method helps in all the k splits. (larger is better)

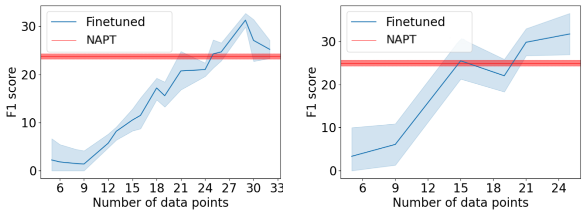
baselines to learn from only a few of examples.

4.5 Zero-Shot Evaluation

Our proposed NAPT step enables the model to perform the following ABSA tasks in zero-shot *i.e.,* without any gold-standard supervision: (i) Aspect-term Extraction (AE), (ii) Aspect-term Extraction and Sentiment Classification (AESC), and (iii) Aspect Sentiment Triplet Extraction (ASTE). We perform two experiments in the zero-shot setting. First, we investigate how much data does a baseline need to reach the performance obtained by our proposed model in the zero-shot setting. Second, we compare against previous work in the ASTE task (Shu et al., 2022).

4.5.1 Dataset Size Equivalence

We compare our proposed method in zero-shot setting against a baseline model trained on gold-annotated data, where we vary the number of training data points. This experiment shows how many annotated data points, on average, is the noisy ABSA pre-training phase equivalent of. We observed that the improvement depends on the difficulty of the task and of the dataset, respectively. For example, Figure 3 shows that for the ASTE task, one would need ~ $15, 25$ annotated data points to obtain a comparable performance with our proposed method for REST15 and LAP14 respectively. We remark that the number of data points vary according to the difficulty of the task and with the difficulty of the dataset, ranging between ~ $6 - 25$ data points for AE, and ASTE task for LAP14 respectively.



(a) LAP14 (b) REST15

Figure 3: Data size equivalence comparison between `t5` models that are `finetuned` on downstream corpus vs our proposed NAPT for ASTE task in (a) LAP14, (b) REST15 respectively. The finetuned models need ~ $15 - 25$ completely annotated data points to equalize our proposed method.

4.5.2 Performance Comparison with Baselines

We compare the zero-shot performance of our proposed method with previous work on ABSA (Shu et al., 2022), summarized in Table 2. Our proposed model outperforms the previous state-of-the-art results for AESC by as much as 6.94%F1 points in the restaurant domain. The improvement for the laptop domain is smaller, we attribute this to the NAPT dataset being biased towards the restaurant domain in terms of size. It is interesting to note that our model's backbone *i.e.,* `t5-base` is able to outperform CORN altough it has almost half the number of parameters as that of its counterpart *i.e.,* `bart-large`.

Model	REST	LAP
CORN	37.20 ±0.50	40.30 ±0.60
IT-MTL-NAPT	**44.14** ±0.30	**40.51** ±0.43

Table 2: Comparison of our proposed method with previous work on zero-shot Aspect Extraction Sentiment Classification (AESC). Our proposed method outperforms the previous work on both datasets. Metric is token-level F1 score.

4.6 Full-Training

We compare the performance of our proposed method (i.e. `pretrain` → `NAPT` → `finetune`) with the standard method of `pretrain` → `finetune` and report the result in Table 3, for all the datasets. Overall in the full-training scenario, our proposed method performs comparably with or better than the baseline. We observe during our preliminary experiments that the training dynamics change drastically between the `pretrain` → `NAPT` → `finetune` and `pretrain` → `finetune`.

5 Discussion

In this section, we would like to discuss a few important aspects of our approach apart from the main experiments.

5.1 Ablation

To better understand how different components of our NAPT strategy influence the final downstream performance, we conduct the following ablation studies.

Model	Dataset		
	LAP14	REST15	REST16
Text	59.50 ± 1.35	51.74 ± 0.84	62.95 ± 0.61
IT	**60.47** ± 1.36	52.78 ± 0.81	**63.77** ± 0.82
IT-MTL	60.17 ± 1.19	53.17 ± 0.67	62.69 ± 0.69
IT-MTL-ID	58.24 ± 1.03	53.42 ± 1.27	62.38 ± 0.69
IT-MTL-NAPT	59.97 ± 1.28	**53.57** ± 1.42	61.67 ± 0.65

Table 3: F1 scores of our proposed method (`IT-MTL-NAPT`) and 4 competitive baselines on the Aspect Sentiment Triplet Extraction task over 3 datasets under training on full dataset. We observe similar levels of performance.

Regarding NAPT Tasks: We analyze the importance of NAPT with multiple tasks and their impact on the downstream performance. Our analysis shows that there exists a positive correlation between the NAPT complexity and downstream performance. We average the downstream performance across every task and every k-shot split and train on the downstream task in a multi-task learning fashion. We summarize our results in Table 4. Our experiments show that it helps in general to align the NAPT and finetuning objectives. If the NAPT phase is done in a multi-task learning fashion, it is beneficial for the model if the same is done for finetuning on the downstream task as well. Additionally, we observe that that harder NAPT tasks are beneficial for the downstream task regardless of the way the training on the downstream task is performed, as the F1 scores reflect the relative order in difficulty of the tasks (*i.e.,* ASTE > AESC > AE).

NAPT Task	Dataset		
	LAP14	REST15	REST16
AE	43.47	46.72	50.76
AESC	44.94	46.99	50.75
ASTE	46.30	47.14	51.17
MTL	47.45	47.32	51.65

Table 4: Ablation study over NAPT tasks in terms of macro F1 scores averaged across all the tasks and 4 k-shot settings. It shows that having all the tasks during NAPT achieves the best scores.

Regarding NAPT Regularization: We analyze the importance on the downstream performance of each regularization technique used during the NAPT phase. We report the performance in Table 6. We analyze the influence of: (i) Tuple Dropout, (ii) Biased weight decay, and (iii) Weight decay. We observe that our proposed approach is robust to hyperparameters, obtaining similar performance with various combinations of the 3 regularization techniques. We attribute this to the way the NAPT dataset is split into train and validation: enforcing disjoint sets of aspect-terms. This allows us to detect when the model starts to overfit.[10]

5.2 Sentiment Prediction: Error Analysis

Quantitative: We first compare the percentage of correct predictions over each sentiment class,

[10]Preliminary experiments shows that regularization was needed, but the training and testing splits contained overlapping aspect terms and opinion terms.

Task : Input	Gold	w/o NAPT	w/ NAPT
ASTE: Given the text: Finally, the biggest problem has been tech support., what are the aspect terms and their sentiments?	<tech support, *negative*>	<support, *negative*>	<tech support, *negative*>
ASTE: What are the aspect terms and their sentiments in the text: Of course, for a student, weight is always an issue.?	<weight, *neutral*>	<weight, *neutral*>	<weight, *negative*>
AESC: Given the text: the mouse buttons are hard to push., what are the aspect term, opinion term, and sentiment triplets?	<mouse buttons, hard, *negative*>	< , , >	<mouse buttons, hard, *negative*>
AESC: Given the text: The resolution is even higher then any other laptop on the market., what are the aspect term, opinion term and sentiment triplets?	<resolution, higher, *positive*>	<resolution, higher, *positive*>	<laptop, higher, *positive*>

Table 5: Predictions made by an instruction tuned model with and without NAPT in low-shot scenarios.

Ablation Config.			Dataset		
Tuple Dropout	Weight Decay	Biased Weight	LAP14	REST15	REST16
✓	✓	✓	47.45	47.32	51.65
✓	✓	×	47.57	47.10	51.39
✓	×	✓	47.62	47.26	51.65
✓	×	×	47.39	47.17	51.37
×	✓	✓	47.55	47.65	51.80
×	✓	×	46.43	47.44	51.49
×	×	✓	46.78	47.12	51.11
×	×	×	46.90	47.27	51.49

Table 6: Ablation study over different regularization techniques in terms of macro F1 scores averaged across all tasks and 4 k-shot settings.

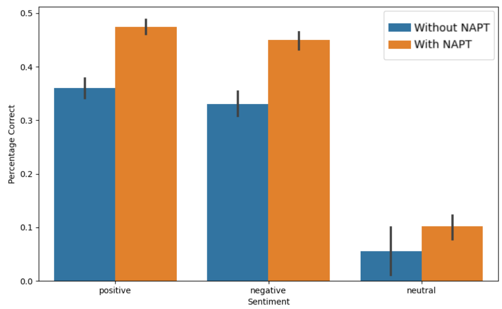


Figure 4: Comparison on the percentage of correct predictions over each sentiment class for an instruction tuned model with vs without the proposed NAPT on the LAP14 dataset and $k = 10$. With NAPT, it performs better on each sentiment class, even though *neutral* class does not appear in the noisy dataset (larger is better).

namely *positive*, *negative*, and *neutral*. We compare instruction tuning with and without our proposed NAPT step. We highlight the results in Figure 4. We observe that our proposed method performs better for every sentiment class. Moreover, we note that our proposed method outperforms the baseline even for the *neutral* sentiment class, a class which has not been seen during the NAPT phase. This suggests that NAPT can help the model learn faster even unseen tasks.

Qualitative: We present examples of the predictions made by an instruction tuned model with and without our proposed NAPT in Table 5. We show 4 predictions, 2 for ASTE (first two rows) and 2 for AESC (bottom two) on LAP14, in low-shot scenarios. We observe that the baseline has difficulties extracting the full aspect term (first row), while our proposed method is able extract the complete triple. The metric used does not reward partial matching. In the second row, the baseline correctly generates the gold output, while our proposed method predicts a *negative* sentiment. In this case, the input can be considered ambiguous, as there is no explicit sentiment expressed in it. Also, for more complex tasks, such as aspect sentiment triplet extraction (AESC), the baseline has difficulties generating a valid prediction, while our proposed method is able to generate the correct prediction (third row). Lastly, we observe that although with NAPT we predict incorrectly (last row), it rather falls back to a term relevant to the domain (*i.e.,* laptop).

6 Conclusion

In this paper, we proposed to add an intermediate step in the `pretrain→finetune` paradigm, called Noisy ABSA Pre-Training. We motivate this newly introduced step with the hypothesis that exposing the model to tasks more aligned with the downstream task will improve its performance, especially in low-data regimes such as in few-shot or complete zero-shot. We constructed a noisy dataset with a heuristic based pipeline approach consisting of three steps that utilize well-studied NLP resources and models. It serves as the training dataset for the noisy pre-training phase. We then evaluated with customer reviews from three datasets covering two domains, *laptop* and *restaurant*, and obtained large improvements in the zero/few-shot cases while achieving similar performance under fine-tuning on full dataset. We also discussed caveats around introducing catastrophic forgetting of general purpose pre-trained language models through such noisy pre-training, and introduced a few regularization techniques to help alleviate it.

Limitations

We believe our proposed noisy pre-training step should apply to other structured prediction tasks, however we have not evaluated the approach on anything other than ABSA related tasks. Additionally, the noisy corpus construction process is heavily dependent on English based resources and pre-trained models. It might be non-trivial to extend the approach to other languages. Finally, we presented some extrinsic evaluation regarding the quality of the noisy corpus we create *e.g.,* equivalence in terms of gold-annotated data size (Section 4.5.1). We leave any intrinsic evaluation of it by means of human supervision or otherwise for future work.

References

Shaowei Chen, Yu Wang, Jie Liu, and Yuelin Wang. 2021. Bidirectional machine reading comprehension for aspect sentiment triplet extraction. *ArXiv*, abs/2103.07665.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *MLCW*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Xiaowen Ding, B. Liu, and Philip S. Yu. 2008. A holistic lexicon-based approach to opinion mining. In *WSDM '08*.

Cícero Nogueira dos Santos and Maíra A. de C. Gatti. 2014. Deep convolutional neural networks for sentiment analysis of short texts. In *COLING*.

Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2017. An unsupervised neural attention model for aspect extraction. In *ACL*.

Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. *Proceedings of the 25th International Conference on World Wide Web*.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *ACL*.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*.

Hiroshi Kanayama and Tetsuya Nasukawa. 2006. Fully automatic lexicon expansion for domain-oriented sentiment analysis. In *EMNLP*.

James Kirkpatrick, Razvan Pascanu, Neil C. Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114:3521 – 3526.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario vSavsko, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clement Delangue, Th'eo Matussiere, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander M. Rush, and Thomas Wolf. 2021. Datasets: A community library for natural language processing. *ArXiv*, abs/2109.02846.

Ruifan Li, Hao Chen, Fangxiang Feng, Zhanyu Ma, Xiaojie Wang, and Eduard H. Hovy. 2021. Dual graph convolutional networks for aspect-based sentiment analysis. In *ACL*.

Pengfei Liu, Shafiq R. Joty, and Helen M. Meng. 2015. Fine-grained opinion mining with recurrent neural networks and word embeddings. In *EMNLP*.

Shu Liu, Kai-Wen Li, and Zuhe Li. 2022. A robustly optimized bmrc for aspect sentiment triplet extraction. In *NAACL*.

Haiyun Peng, Lu Xu, Lidong Bing, Fei Huang, Wei Lu, and Luo Si. 2020. Knowing what, how and why: A near complete solution for aspect-based sentiment analysis. In *AAAI*.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad AL-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, Véronique Hoste, Marianna Apidianaki, Xavier Tannier, Natalia Loukachevitch, Evgeniy Kotelnikov, Nuria Bel, Salud María Jiménez-Zafra, and Gülşen Eryiğit. 2016. SemEval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 19–30, San Diego, California. Association for Computational Linguistics.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2015.

SemEval-2015 task 12: Aspect based sentiment analysis. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 486–495, Denver, Colorado. Association for Computational Linguistics.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. 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. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang A. Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal V. Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Févry, Jason Alan Fries, Ryan Teehan, Stella Rose Biderman, Leo Gao, Tali Bers, Thomas Wolf, and Alexander M. Rush. 2022. Multitask prompted training enables zero-shot task generalization. *ArXiv*, abs/2110.08207.

Lei Shu, Jiahua Chen, Bing Liu, and Hu Xu. 2022. Zero-shot aspect-based sentiment analysis. *ArXiv*, abs/2202.01924.

Lei Shu, Hu Xu, and Bing Liu. 2019. Controlled cnn-based sequence labeling for aspect extraction. *ArXiv*, abs/1905.06407.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, A. Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *ArXiv*, abs/2004.09297.

Stéphan Tulkens and Andreas van Cranenburgh. 2020. Embarrassingly simple unsupervised aspect extraction. In *ACL*.

Siddharth Varia, Shuai Wang, Kishaloy Halder, Robert Vacareanu, Miguel Ballesteros, Yassine Benajiba, Neha Anna John, Rishita Anubhai, Smaranda Muresan, and Dan Roth. 2022. Instruction tuning for few-shot aspect-based sentiment analysis. *ArXiv*.

Duy-Tin Vo and Yue Zhang. 2015. Target-dependent twitter sentiment classification with rich automatic features. In *IJCAI*.

Bo Wang, Tao Shen, Guodong Long, Tianyi Zhou, and Yi Chang. 2021. Eliminating sentiment bias for aspect-level sentiment classification with unsupervised opinion extraction. In *EMNLP*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2020. Transformers: State-of-the-art natural language processing. In *EMNLP*.

Hu Xu, Bing Liu, Lei Shu, and Philip S. Yu. 2019. Bert post-training for review reading comprehension and aspect-based sentiment analysis. *ArXiv*, abs/1904.02232.

Liheng Xu, Kang Liu, Siwei Lai, Yubo Chen, and Jun Zhao. 2013. Mining opinion words and opinion targets in a two-stage framework. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1764–1773, Sofia, Bulgaria. Association for Computational Linguistics.

Lu Xu, Hao Li, Wei Lu, and Lidong Bing. 2020. Position-aware tagging for aspect sentiment triplet extraction. In *EMNLP*.

Hang Yan, Junqi Dai, Tuo Ji, Xipeng Qiu, and Zheng Zhang. 2021. A unified generative framework for aspect-based sentiment analysis. *ArXiv*, abs/2106.04300.

Guoxin Yu, Jiwei Li, Ling Luo, Yuxian Meng, Xiang Ao, and Qing He. 2021. Self question-answering: Aspect-based sentiment analysis by role flipped machine reading comprehension. In *EMNLP*.

Meishan Zhang, Yue Zhang, and Duy-Tin Vo. 2015a. Neural networks for open domain targeted sentiment. In *EMNLP*.

Wenxuan Zhang, Yang Deng, Xin Li, Yifei Yuan, Lidong Bing, and Wai Lam. 2021a. Aspect sentiment quad prediction as paraphrase generation. *ArXiv*, abs/2110.00796.

Wenxuan Zhang, Xin Li, Yang Deng, Lidong Bing, and Wai Lam. 2021b. Towards generative aspect-based sentiment analysis. In *ACL*.

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

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015b. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

A Implementation details

We use HuggingFace's implementation of transformers (Wolf et al., 2020; Lhoest et al., 2021). We use similar parameters as (Varia et al., 2022). We run our experiments on NVIDIA Tesla V100 GPUs.

B All Experiments

For completeness, we include here all the models investigated over the 3 datasets, LAP14, REST15, and REST16, respectively.

B.1 Full-Training

We report the results (*test*) on Full Training in Tables 7, 8, 9.

B.2 K-Shot Learning

We report the results (*test*) on K-Shot Learning in Tables 10, 11, 12.

B.3 Cross Domain

We experiment with pre-training on a different domain than the domain of the downstream task. Concretely, we perform two experiments: (i) we perform NAPT on restaurant domain, then finetune on the laptop domain, and (ii) we perform NAPT on the laptop domain, then finetune on the restaurant domain. We include the results with our proposed model trained with NAPT on restaurant data and finetuned on LAP14 in Table 13. We include the results with our proposed model trained with NAPT on laptop data and finetuned on REST15 and REST16 in Table 14 and in Table 15, respectively. We observed that our proposed model is still able to transfer the knowledge learned during the NAPT phase. Our proposed model still outperforms the baseline, brining as much as 11.49% F1 points for the ASTE task in the laptop domain. In some cases we noticed a slight increase in the final performance compared to the model trained with NAPT on the full dataset. This suggests that the model trained on the full dataset overfits to the noisy data.

C Multi-word Patterns

In Table 16 we list all the patterns that were used to filter 2-grams, 3-grams and 4-grams during aspect term extraction.

Model	NAPT	Task (F1 ↑)			Average
		AE	AESC	ASTE	
Text (t5-base)	No	76.13±1.06	66.57±1.01	59.50±1.35	67.40±7.13
IT (t5-base)	No	77.09±0.68	66.25±0.45	60.47±1.36	67.94±7.18
	Yes	76.96±1.17	66.08±0.80	60.03±1.23	67.69±7.16
IT-MTL (t5-base)	No	77.64±0.75	66.54±1.09	60.17±1.19	68.11±7.53
	Yes	77.67±1.04	66.66±0.69	59.97±1.28	68.10±7.45
IT (t5-large)	No	77.18±1.64	67.20±1.23	60.24±0.61	68.21±7.28
	Yes	76.79±1.05	66.66±1.16	60.98±1.78	68.14±6.78
IT-MTL (t5-large)	No	77.89±0.53	66.44±1.06	59.83±2.32	68.05±7.85
	Yes	77.95±1.00	65.62±1.23	59.34±1.42	67.64±7.95
IT (continued pre-training) (t5-base)	No	75.77±0.71	65.99±0.98	59.28±0.64	67.01±7.05
	Yes	76.19±1.33	66.28±1.36	59.38±1.25	67.28±7.09
IT-MTL (continued pre-training) (t5-base)	No	76.37±0.82	65.85±1.03	58.24±1.03	66.82±7.74
	Yes	76.68±0.88	65.95±1.06	58.44±1.26	67.03±7.64

Table 7: Comparison of full dataset training performances on all 3 ABSA tasks for LAP14.

Model	NAPT	Task (F1 ↑)					Average
		AE	AESC	TASD	ASTE	ASQP	
Text (t5-base)	No	72.76±0.96	66.43±1.45	60.05±0.67	51.74±0.84	46.66±0.67	59.53±9.72
IT (t5-base)	No	73.54±1.20	67.09±0.53	59.78±0.91	52.78±0.81	46.79±0.59	59.99±9.82
	Yes	72.89±1.31	65.98±1.29	59.30±0.77	52.62±1.13	46.49±0.71	59.45±9.48
IT-MTL (t5-base)	No	73.85±1.14	67.46±0.80	59.88±1.02	53.17±0.67	47.17±1.03	60.30±9.81
	Yes	74.55±1.26	67.53±1.37	59.29±1.67	53.57±1.42	47.30±1.21	60.45±9.86
IT (t5-large)	No	74.24±0.74	69.83±1.10	62.82±0.69	55.96±0.41	49.61±0.55	62.49±9.16
	Yes	74.68±0.72	69.94±1.18	62.82±0.94	54.72±1.53	49.48±1.04	62.33±9.47
IT-MTL (t5-large)	No	75.79±0.69	70.18±1.31	62.84±1.37	54.16±0.95	48.86±1.13	62.37±10.17
	Yes	74.80±0.94	68.26±0.96	61.11±1.10	53.69±1.40	48.41±1.26	61.25±9.70
IT (continued pre-training) (t5-base)	No	73.05±1.05	67.17±1.16	59.09±0.91	51.89±1.09	46.51±0.36	59.54±9.92
	Yes	72.82±1.11	67.44±0.99	60.42±0.95	53.07±0.88	47.56±1.50	60.26±9.31
IT-MTL (continued pre-training) (t5-base)	No	74.14±0.47	68.06±0.49	60.97±0.59	53.42±1.27	47.49±0.90	60.82±9.84
	Yes	74.66±1.06	68.59±0.78	61.14±0.88	53.42±0.75	48.41±0.55	61.24±9.69

Table 8: Comparison of full dataset training performances on all 5 ABSA tasks for REST15.

Model	NAPT	Task (F1 ↑) AE	AESC	TASD	ASTE	ASQP	Average
Text (t5-base)	No	78.40±1.14	73.64±1.30	67.05±0.96	62.95±0.61	57.77±1.13	67.96±7.58
IT (t5-base)	No	79.74±0.98	74.24±0.54	68.04±0.86	63.77±0.82	58.41±0.73	68.84±7.72
	Yes	78.69±1.30	72.90±0.98	67.40±1.20	61.96±0.94	57.57±1.25	67.70±7.66
IT-MTL (t5-base)	No	79.90±0.62	74.51±0.91	67.59±0.75	62.69±0.69	57.72±0.76	68.48±8.15
	Yes	78.53±0.75	73.31±0.87	66.72±0.98	61.67±0.65	56.78±0.65	67.40±7.90
IT (t5-large)	No	79.66±0.98	76.90±0.93	70.24±1.13	65.15±0.20	60.13±1.06	70.42±7.42
	Yes	78.87±1.11	75.25±0.80	70.40±0.81	64.61±1.11	59.76±0.86	69.78±7.06
IT-MTL (t5-large)	No	79.67±0.50	75.01±0.95	69.12±1.04	62.84±0.98	58.79±0.99	69.09±7.85
	Yes	79.33±0.78	74.66±0.72	67.11±1.66	62.43±0.99	57.17±1.17	68.14±8.18
IT (continued pre-training) (t5-base)	No	79.22±0.59	74.05±0.70	67.58±1.61	62.69±1.58	57.73±0.82	68.25±7.92
	Yes	79.06±0.92	74.38±1.30	68.40±1.21	62.33±1.25	58.24±0.83	68.48±7.74
IT-MTL (continued pre-training) (t5-base)	No	79.25±0.58	74.13±0.56	67.72±0.80	62.38±0.69	58.04±0.87	68.30±7.86
	Yes	78.72±0.73	73.88±0.95	67.16±1.00	62.00±1.15	56.61±1.01	67.68±8.05

Table 9: Comparison of full dataset training performances on all 5 ABSA tasks for REST16.

K	Model	NAPT	Task (F1 ↑) AE	AESC	ASTE	Average
5	Text (t5-base)	No	37.45±2.94	22.91±1.65	12.06±1.83	24.14±10.96
	IT (t5-base)	No	44.59±1.15	26.81±2.35	13.04±0.91	28.14±13.45
		Yes	47.46±2.76	38.85±2.11	28.88±1.58	38.40±7.98
	IT-MTL (t5-base)	No	36.63±3.03	25.31±2.78	15.96±2.11	25.97±9.09
		Yes	47.02±2.60	36.49±1.97	27.53±1.97	37.02±8.34
	IT (t5-large)	No	43.01±2.09	26.73±2.86	16.14±2.19	28.63±11.66
		Yes	46.92±2.71	37.52±2.44	25.81±2.62	36.75±9.13
	IT-MTL (t5-large)	No	40.88±3.65	27.47±2.72	17.37±2.51	28.57±10.35
		Yes	45.30±3.29	32.47±5.05	23.54±5.34	33.77±10.13
	IT (continued pre-training) (t5-base)	No	36.59±0.91	22.82±1.20	12.38±0.88	23.93±10.31
		Yes	45.83±1.80	38.85±1.31	28.15±1.84	37.61±7.53
	IT-MTL (continued pre-training) (t5-base)	No	26.25±2.32	22.40±1.26	13.62±1.98	20.76±5.75
		Yes	45.28±1.27	36.61±1.46	27.33±2.02	36.41±7.58
10	Text (t5-base)	No	46.85±2.12	33.67±1.71	18.95±2.91	33.16±11.99
	IT (t5-base)	No	52.12±2.42	37.49±1.91	25.22±0.83	38.28±11.51
		Yes	55.98±2.16	45.02±1.64	36.62±2.61	45.87±8.29
	IT-MTL (t5-base)	No	48.71±1.89	39.13±2.29	28.00±2.59	38.61±9.01
		Yes	55.81±2.14	44.49±1.50	35.15±1.71	45.15±8.72
	IT (t5-large)	No	49.44±9.70	36.64±3.64	25.10±1.46	37.06±11.71
		Yes	53.13±4.59	43.35±2.91	34.94±1.49	43.81±8.19
	IT-MTL (t5-large)	No	49.23±4.91	36.13±2.07	27.16±3.74	37.51±10.01
		Yes	51.99±3.47	41.45±2.28	31.05±4.58	41.50±9.35
	IT (continued pre-training) (t5-base)	No	41.61±6.49	33.89±1.69	21.36±2.57	32.29±9.45
		Yes	55.69±2.27	45.77±1.55	34.51±1.20	45.32±8.91
	IT-MTL (continued pre-training) (t5-base)	No	41.65±1.78	34.44±2.71	24.55±1.50	33.55±7.50
		Yes	56.16±2.60	46.17±1.79	35.25±1.06	45.86±8.84
20	Text (t5-base)	No	56.56±1.15	42.64±0.99	29.18±2.23	42.79±11.66
	IT (t5-base)	No	59.08±1.97	44.82±1.24	33.24±1.53	45.71±11.04
		Yes	61.67±1.81	48.88±1.10	41.20±2.01	50.58±8.70
	IT-MTL (t5-base)	No	57.98±3.72	47.14±2.42	34.55±1.85	46.56±10.24
		Yes	61.05±1.62	48.94±1.68	38.17±1.96	49.38±9.60
	IT (t5-large)	No	59.30±2.38	46.88±2.92	34.44±2.61	46.88±10.79
		Yes	61.43±1.44	49.00±3.37	38.52±1.84	49.65±9.79
	IT-MTL (t5-large)	No	61.02±2.89	46.78±4.32	36.00±1.17	47.93±10.99
		Yes	61.16±1.97	49.68±2.13	38.10±2.41	49.65±9.80
	IT (continued pre-training) (t5-base)	No	53.92±1.64	43.56±1.02	28.45±1.62	41.98±10.91
		Yes	60.06±2.47	49.73±1.48	40.19±1.64	49.99±8.42
	IT-MTL (continued pre-training) (t5-base)	No	55.64±2.04	45.44±1.97	32.12±1.28	44.40±10.11
		Yes	60.93±1.36	49.85±1.65	37.96±1.78	49.58±9.61
50	Text (t5-base)	No	65.31±1.86	54.35±1.15	40.84±2.53	53.50±10.51
	IT (t5-base)	No	68.95±1.22	54.92±1.07	44.67±2.12	56.18±10.40
		Yes	68.14±1.12	54.67±1.82	46.56±1.38	56.46±9.11
	IT-MTL (t5-base)	No	67.54±1.62	55.86±1.90	45.10±2.69	56.16±9.69
		Yes	68.23±1.34	54.79±1.68	45.85±1.11	56.29±9.40
	IT (t5-large)	No	68.27±3.17	56.37±1.48	45.26±1.55	56.64±9.94
		Yes	68.36±1.15	57.99±2.05	47.23±2.36	57.86±8.97
	IT-MTL (t5-large)	No	69.92±1.23	56.33±1.24	44.87±2.10	57.04±10.70
		Yes	70.07±1.30	55.99±0.95	45.99±2.25	57.35±10.16
	IT (continued pre-training) (t5-base)	No	63.36±1.05	48.97±0.84	37.31±1.78	49.88±11.09
		Yes	68.78±1.42	55.20±1.08	45.50±1.44	56.49±9.74
	IT-MTL (continued pre-training) (t5-base)	No	63.72±0.64	53.02±1.08	40.83±1.10	52.53±9.72
		Yes	69.19±1.31	55.73±1.11	45.44±1.56	56.79±9.92

Table 10: Comparison of k-Shot performances on all 3 ABSA tasks for LAP14.

K	Model	NAPT	Task (F1 ↑) ATE	AESC	TASD	ASTE	ASQP	Average
5	Text (t5-base)	No	44.55±2.55	39.44±2.64	24.62±1.56	20.11±1.05	12.88±0.91	28.32±12.26
	IT	No	49.33±0.66	42.48±1.84	24.75±0.65	24.44±1.09	15.52±1.47	31.31±12.87
	(t5-base)	Yes	50.05±2.91	43.95±1.79	30.46±1.87	31.59±1.35	21.72±0.90	35.56±10.37
	IT-MTL	No	48.14±2.79	41.42±3.28	24.79±2.33	24.49±1.85	15.28±1.64	30.82±12.53
	(t5-base)	Yes	51.11±1.81	43.51±1.55	27.12±1.97	30.35±1.48	18.98±1.39	34.21±11.76
	IT	No	46.40±1.56	41.24±0.86	24.73±1.99	22.72±1.95	16.04±3.00	30.23±11.96
	(t5-large)	Yes	47.87±4.76	43.01±2.77	28.42±7.70	30.49±1.43	20.85±1.79	34.13±10.84
	IT-MTL	No	44.54±2.84	36.25±1.78	19.08±3.03	18.92±3.92	10.57±2.01	25.87±13.05
	(t5-large)	Yes	48.47±1.98	40.38±2.76	23.79±3.88	26.97±3.56	16.25±3.37	31.17±12.16
	IT	No	46.06±2.36	39.34±3.07	24.67±1.17	22.70±0.85	14.47±1.62	29.45±11.92
	(continued pre-training) (t5-base)	Yes	50.40±1.76	44.06±1.59	29.32±2.16	31.31±2.31	22.20±2.32	35.46±10.53
	IT-MTL	No	47.78±2.49	39.59±1.24	24.33±1.43	22.93±0.56	14.55±1.32	29.84±12.40
	(continued pre-training) (t5-base)	Yes	50.87±2.76	44.15±2.18	29.30±2.79	31.60±2.05	20.98±2.28	35.38±11.06
10	Text (t5-base)	No	54.71±0.91	49.28±0.46	36.26±1.62	31.99±0.80	24.42±0.68	39.33±11.41
	IT	No	56.62±1.59	51.03±1.93	37.64±1.50	33.25±1.54	25.76±1.08	40.86±11.71
	(t5-base)	Yes	57.91±1.29	50.78±1.42	37.37±1.81	37.63±1.26	28.78±1.11	42.49±10.59
	IT-MTL	No	58.10±0.72	48.27±0.98	37.26±0.29	33.75±0.74	26.48±1.01	40.77±11.41
	(t5-base)	Yes	58.72±1.23	49.95±1.30	36.77±1.68	37.82±1.70	28.03±1.21	42.26±10.95
	IT	No	54.58±1.99	48.32±1.27	35.31±1.90	34.55±0.86	25.43±1.79	39.64±10.76
	(t5-large)	Yes	55.69±1.94	49.52±1.42	38.11±1.76	36.54±1.71	28.10±1.53	41.59±10.04
	IT-MTL	No	54.14±1.11	45.38±1.09	33.90±2.76	30.95±1.68	23.10±1.47	37.49±11.31
	(t5-large)	Yes	55.00±3.53	46.91±3.01	35.09±2.65	32.82±2.97	24.79±2.71	38.92±11.19
	IT	No	56.55±2.35	51.28±0.82	39.02±2.58	33.70±1.41	25.10±0.66	41.13±11.81
	(continued pre-training) (t5-base)	Yes	57.96±1.36	51.42±1.41	39.33±1.29	37.81±1.68	29.57±1.32	43.22±10.31
	IT-MTL	No	58.31±0.92	49.57±2.13	39.00±2.28	33.01±1.21	25.58±0.75	41.09±11.98
	(continued pre-training) (t5-base)	Yes	57.88±1.58	50.34±1.87	38.56±1.47	37.83±1.22	28.73±1.32	42.67±10.42
20	Text (t5-base)	No	58.91±1.69	53.77±0.90	42.37±1.55	37.27±1.85	30.45±0.83	44.55±10.76
	IT	No	62.08±1.85	53.91±2.18	42.89±0.86	38.35±0.83	30.77±1.19	45.60±11.45
	(t5-base)	Yes	61.84±1.18	53.80±1.19	44.13±1.19	41.93±1.13	34.23±1.30	47.19±9.76
	IT-MTL	No	63.77±1.86	53.47±2.10	43.27±1.33	40.66±2.07	33.27±0.76	46.89±10.97
	(t5-base)	Yes	63.77±1.15	55.48±1.55	44.24±1.18	42.77±1.16	34.71±0.91	48.19±10.36
	IT	No	59.97±1.49	55.11±1.86	45.59±1.00	40.27±1.10	34.40±1.80	47.07±9.67
	(t5-large)	Yes	62.13±1.32	55.85±1.68	46.35±2.68	41.79±0.71	35.69±1.19	48.36±9.75
	IT-MTL	No	62.26±1.55	54.59±2.62	45.04±1.44	40.39±2.01	34.23±1.12	47.30±10.35
	(t5-large)	Yes	63.19±1.70	55.67±2.23	44.23±1.40	41.77±1.48	34.43±1.25	47.86±10.49
	IT	No	62.30±1.44	55.82±1.49	45.16±1.25	38.23±1.54	31.58±0.96	46.62±11.52
	(continued pre-training) (t5-base)	Yes	62.85±1.38	56.12±0.90	45.51±1.57	42.07±1.53	34.48±1.13	48.21±10.25
	IT-MTL	No	63.42±0.89	55.09±0.49	46.43±1.13	40.40±1.45	32.85±0.67	47.64±11.00
	(continued pre-training) (t5-base)	Yes	63.91±1.21	56.14±1.47	46.40±1.18	42.80±1.34	36.15±0.92	49.08±9.97
50	Text (t5-base)	No	62.55±1.74	57.12±1.31	48.50±0.97	43.09±0.91	35.51±0.82	49.35±9.91
	IT	No	64.74±1.15	59.35±0.91	50.40±0.65	43.79±1.12	37.51±0.72	51.16±10.17
	(t5-base)	Yes	65.17±0.76	58.96±0.92	49.72±1.24	44.74±1.34	39.10±1.16	51.54±9.56
	IT-MTL	No	67.51±0.89	58.98±1.52	50.45±1.49	45.27±0.76	37.69±1.04	51.98±10.68
	(t5-base)	Yes	67.55±1.18	60.19±1.23	50.51±1.09	46.76±0.93	39.94±0.86	52.99±9.91
	IT	No	64.75±0.94	59.33±0.47	52.19±0.93	45.59±0.75	40.66±1.12	52.50±8.99
	(t5-large)	Yes	66.82±1.16	61.21±1.40	52.53±1.76	47.19±1.30	42.27±1.41	54.00±9.16
	IT-MTL	No	67.84±1.16	60.77±1.23	51.70±0.97	46.76±1.45	39.92±1.00	53.40±10.18
	(t5-large)	Yes	68.15±0.86	61.67±0.94	52.02±1.47	47.33±1.30	41.24±1.18	54.08±9.86
	IT	No	64.49±0.95	60.23±0.51	51.51±0.81	44.10±1.74	37.56±1.30	51.58±10.20
	(continued pre-training) (t5-base)	Yes	65.37±1.20	59.64±1.12	51.08±0.65	45.49±1.14	39.37±0.90	52.19±9.49
	IT-MTL	No	67.46±1.03	61.93±0.70	52.73±1.10	46.06±0.61	39.71±1.70	53.58±10.38
	(continued pre-training) (t5-base)	Yes	67.37±0.96	60.54±1.29	51.57±1.25	46.96±1.12	40.39±1.03	53.37±9.72

Table 11: Comparison of k-Shot performances on all 5 ABSA tasks for REST15.

K	Model	NAPT	Task (F1 ↑) AE	AESC	TASD	ASTE	ASQP	Average
5	Text (t5-base)	No	52.67±0.69	47.87±1.34	31.57±1.74	29.58±1.96	19.76±1.44	36.29±12.52
	IT (t5-base)	No	55.59±2.74	51.62±1.46	36.26±1.15	34.10±1.17	23.89±2.11	40.29±12.07
		Yes	61.54±1.35	55.32±2.05	39.13±2.11	40.18±1.60	28.64±1.82	44.96±12.09
	IT-MTL (t5-base)	No	59.78±1.32	52.35±0.82	36.88±1.77	36.27±0.90	25.86±1.63	42.23±12.50
		Yes	64.25±1.60	55.22±1.35	38.97±2.19	40.95±1.36	29.58±1.76	45.79±12.54
	IT (t5-large)	No	55.88±1.63	52.90±2.02	38.37±2.79	36.70±0.83	27.70±1.85	42.31±10.91
		Yes	62.01±1.48	55.91±2.68	37.09±7.90	41.14±1.78	32.13±2.04	45.66±12.13
	IT-MTL (t5-large)	No	56.81±2.44	48.65±1.32	32.64±2.56	32.47±1.80	23.36±1.16	38.79±12.52
		Yes	60.50±1.91	51.89±2.50	34.94±3.71	37.71±2.08	27.04±2.22	42.42±12.46
	IT (continued pre-training) (t5-base)	No	55.87±2.42	50.92±3.05	36.57±1.38	31.41±1.94	20.39±2.38	39.03±13.37
		Yes	62.22±1.99	56.70±1.43	37.00±2.16	39.19±1.67	27.18±1.64	44.46±13.22
	IT-MTL (continued pre-training) (t5-base)	No	55.93±1.71	48.95±2.26	34.71±2.20	32.02±0.88	22.79±1.46	38.88±12.31
		Yes	62.74±1.21	55.43±0.86	37.13±2.36	39.84±1.37	27.80±1.88	44.59±12.89
10	Text (t5-base)	No	59.45±0.89	54.33±1.03	38.85±1.95	36.82±0.91	29.31±1.17	43.75±11.59
	IT (t5-base)	No	62.14±1.14	57.02±2.17	40.34±2.22	40.37±0.74	29.90±0.94	45.95±12.20
		Yes	65.33±1.18	58.84±1.48	42.69±2.83	44.24±1.07	32.30±1.39	48.68±12.07
	IT-MTL (t5-base)	No	64.03±1.81	56.51±0.97	41.53±1.12	39.66±1.50	31.27±1.37	46.60±12.24
		Yes	65.85±1.08	57.96±1.14	41.66±2.32	44.42±1.00	32.77±2.38	48.53±12.04
	IT (t5-large)	No	59.01±1.07	51.11±3.59	42.76±1.89	39.66±1.81	31.75±2.54	44.86±9.84
		Yes	61.41±2.08	57.90±1.18	43.13±2.28	43.26±1.74	35.40±2.29	48.22±10.10
	IT-MTL (t5-large)	No	59.76±1.11	53.26±2.04	39.01±2.52	37.45±1.46	29.06±1.24	43.71±11.52
		Yes	61.85±1.89	54.15±2.23	39.64±2.10	39.74±2.18	31.13±2.01	45.30±11.39
	IT (continued pre-training) (t5-base)	No	59.25±2.32	56.57±2.06	39.28±2.12	37.84±1.59	26.17±1.79	43.82±12.78
		Yes	63.34±2.30	59.95±1.25	42.75±2.43	44.85±2.03	32.25±1.23	48.63±11.73
	IT-MTL (continued pre-training) (t5-base)	No	60.50±1.25	55.34±0.67	41.57±2.03	38.22±0.89	30.40±1.30	45.20±11.41
		Yes	65.10±1.28	57.91±1.32	43.31±1.75	43.55±1.59	34.27±1.53	48.83±11.28
20	Text (t5-base)	No	63.34±1.24	57.56±1.21	44.90±1.99	42.11±1.82	35.20±0.58	48.62±10.62
	IT (t5-base)	No	65.89±1.90	60.52±1.44	47.27±2.49	44.27±0.99	36.39±0.75	50.87±11.14
		Yes	66.73±1.49	60.78±1.11	50.49±1.21	47.75±1.07	40.14±1.28	53.18±9.61
	IT-MTL (t5-base)	No	65.82±0.96	59.66±1.06	49.30±0.99	44.71±0.72	38.71±0.76	51.64±10.10
		Yes	67.97±0.97	60.81±1.05	49.82±1.09	47.94±1.11	40.25±1.24	53.36±9.95
	IT (t5-large)	No	64.63±0.41	61.07±0.94	49.74±2.34	46.02±1.34	40.53±1.04	52.40±9.36
		Yes	65.24±1.28	60.14±2.72	51.82±1.85	48.44±0.97	41.14±1.09	53.35±8.76
	IT-MTL (t5-large)	No	66.26±2.38	59.48±2.01	48.37±2.96	44.70±3.16	37.42±3.04	51.25±10.85
		Yes	67.08±1.94	60.17±1.09	49.08±1.99	47.13±1.56	39.76±1.43	52.64±9.96
	IT (continued pre-training) (t5-base)	No	63.43±1.03	58.89±1.36	46.15±2.18	44.17±1.96	35.39±0.76	49.61±10.51
		Yes	65.85±0.78	60.97±0.69	49.82±1.10	47.38±1.23	39.20±1.17	52.64±9.71
	IT-MTL (continued pre-training) (t5-base)	No	66.16±1.22	60.56±0.99	49.84±1.06	44.86±2.36	38.42±0.86	51.97±10.43
		Yes	68.00±1.07	61.54±1.14	50.66±1.09	48.11±1.08	40.35±1.30	53.73±9.97
50	Text (t5-base)	No	69.06±0.70	63.97±0.59	55.42±0.70	50.50±0.99	45.91±1.56	56.97±8.73
	IT (t5-base)	No	70.11±0.84	65.75±1.08	55.06±0.94	51.58±1.23	47.56±1.36	58.01±8.78
		Yes	70.14±0.97	65.13±0.82	55.86±0.95	52.63±0.94	47.53±1.02	58.26±8.36
	IT-MTL (t5-base)	No	72.11±1.36	65.68±1.05	56.92±0.84	52.80±1.07	46.75±1.39	58.85±9.29
		Yes	71.92±0.88	65.88±0.70	56.56±0.99	53.83±1.08	47.88±1.37	59.21±8.72
	IT (t5-large)	No	70.57±0.96	67.34±1.68	58.99±1.29	53.13±0.93	48.87±0.94	59.78±8.46
		Yes	71.77±0.77	66.66±1.11	59.59±1.44	55.06±1.45	50.36±0.89	60.69±7.88
	IT-MTL (t5-large)	No	71.73±0.55	66.65±1.05	57.89±0.76	53.17±2.33	47.69±1.62	59.42±9.02
		Yes	72.38±0.83	66.70±0.77	58.48±1.27	53.89±1.56	48.45±1.53	59.98±8.78
	IT (continued pre-training) (t5-base)	No	69.80±1.11	65.11±0.51	55.94±1.51	50.75±1.06	45.25±1.11	57.37±9.27
		Yes	70.06±1.29	64.81±1.12	55.68±0.95	52.12±0.98	46.69±1.49	57.87±8.62
	IT-MTL (continued pre-training) (t5-base)	No	72.08±0.79	66.74±0.99	58.02±0.95	52.48±1.77	46.66±1.35	59.19±9.49
		Yes	71.20±0.87	65.79±1.19	56.68±0.96	53.31±0.87	47.10±0.85	58.82±8.76

Table 12: Comparison of k-Shot performances on all 5 ABSA tasks for REST16.

k	AE	AESC	ASTE	Average
5	47.55±2.06	36.55±2.35	24.53±2.25	33.06
10	55.93±2.80	45.55±2.39	35.38±1.80	43.33
20	64.55±1.47	52.18±1.07	41.67±1.97	52.51
50	69.52±0.71	56.25±1.44	46.49±1.97	57.30
Full Dataset	77.32±1.18	68.20±0.72	60.93±1.12	68.56

Table 13: Cross-Domain performance of `IT-MTL-NAPT` on LAP14. The NAPT was done only on *Yelp* corpus.

k	AE	AESC	TASD	ASTE	ASQP	Average
5	53.17±2.79	44.54±1.97	29.26±1.96	32.89±1.58	21.75±1.25	35.80
10	63.07±1.43	53.79±2.13	38.05±1.82	42.22±1.76	29.80±2.15	45.41
20	68.99±1.34	60.20±1.21	44.84±1.23	46.01±1.22	35.18±1.55	52.02
50	74.20±0.89	64.50±0.85	50.67±1.08	50.18±1.65	40.49±1.37	57.54
Full Dataset	79.39±1.07	72.37±1.02	62.92±1.11	58.95±1.11	51.38±0.90	65.32

Table 14: Cross-Domain performance of `IT-MTL-NAPT` on REST15. The NAPT was done only on *Amazon Reviews* corpus.

k	AE	AESC	TASD	ASTE	ASQP	Average
5	59.17±1.63	54.07±1.35	38.05±2.04	41.03±1.68	29.26±1.74	43.46
10	62.80±1.54	57.27±1.71	42.65±2.11	43.66±1.44	34.14±1.18	47.74
20	66.06±1.21	60.46±1.58	47.96±1.34	47.10±1.30	38.32±1.02	52.31
50	69.67±1.12	64.61±0.76	54.17±1.40	51.91±1.08	45.29±1.25	57.80
Full Dataset	80.72±0.81	75.72±0.89	68.95±0.97	64.04±0.84	58.02±0.97	68.84

Table 15: Cross-Domain performance of `IT-MTL-NAPT` on REST16. The NAPT was done only on *Amazon Reviews* corpus.

Multi-word Patterns
NN*-NN*
JJ*-NN*
VBG-NN*
VBN-NN*
NN*-NN*-NN*
NN*-IN-NN*
JJ*-NN*-NN*
JJ*-JJ*-NN*
VBN-JJ*-NN*
NN*-NN*-NN*-NN*
NN*-CC-NN*-NN*

Table 16: Multi-word Patterns used to filter 2-grams, 3-grams and 4-grams. '*' denotes any variant of the corresponding POS tags. For example, NN* captures NN, NNS, NNP, NNPS.