Prompt-Based Contrastive Pre-Training for Unified Aspect-Based Sentiment Analysis

Project-IV (EE57502) submitted to
Indian Institute of Technology Kharagpur
for the Award of the Degree

Master of Technology
in
Electrical Engineering
with specialization in
Instrumentation and Signal Processing
by

Nithish Kannen
18EE35010

Under the supervision of

Dr. Anirban Mukherjee, Dr. Pawan Goyal

Department of Electrical Engineering
Indian Institute of Technology Kharagpur
April 2023

DEPARTMENT OF ELECTRICAL ENGINEERING, INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR,
KHARAGPUR - 721302, INDIA

Certificate

This is to certify that the work contained in this thesis titled, ***"Prompt-Based Contrastive Pre-Training for Unified Aspect Based Sentiment Analysis"*** *is a bonafide work of* ***Nithish Kannen*** *(Roll no:* ***18EE35010****), carried out in the Department of Electrical Engineering, Indian Institute of Technology Kharagpur under my supervision and that it has not been submitted elsewhere for a degree.*

Dr. Anirban Mukherjee
Associate Professor
Department of Electrical Engineering
Indian Institute of Technology Kharagpur,
West Bengal

Dr. Pawan Goyal
Associate Professor
Department of Computer Science & Engineering
Indian Institute of Technology Kharagpur,
West Bengal

Declaration

I certify that

(a) The work contained in this report has been done by me under the guidance of my supervisor.

(b) The work has not been submitted to any other Institute for any degree or diploma.

(c) I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

(d) Whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references. Further, I have taken permission from the copyright owners of the sources, whenever necessary.

Date: April 2023
Place: Kharagpur

Nithish Kannen
18EE35010

TABLE OF CONTENTS

List of Figures

1. Abstract

Aspect-Based Sentiment Analysis, also abbreviated as ABSA is an important task in the Natural Language Understanding (NLU) domain that can widely impact Product Services, and E-commerce by enabling automatic information extraction to understand user sentiment or opinion towards aspects (entities). For example, in the sentence "*The **battery** was **down** but the **display** was **fire**" **battery*** and ***display*** are the **aspects,** and ***down*** and ***fire*** are the corresponding **opinions.** Note that a negative sentiment is expressed towards battery and positive towards display. Aspect Based Sentiment Analysis involves extracting aspect-oriented sentiments from sentences and is more sophisticated than sentence-level sentiment analysis. ABSA consists of nine subtasks, namely **Aspect Term Extraction (AE), Opinion Term Extraction (OE), Aspect and Opinion Pair Extraction (AO), Aspect-oriented Opinion Extraction (AOE), Aspect Term Extraction and Sentiment Classification (AESC), Aspect -level Sentiment Classification (ALSC), Aspect Sentiment Triplet Extraction (ASTE), Target Aspect Sentiment Detection (TASD) and Aspect Category Opinion Sentiment (ACOS)**. While most previous works propose a framework to handle one out of these array of tasks, we, on the other hand, propose a unified framework to handle all of the above ABSA sub-tasks with a single framework inspired from prompt based learning. Even more, we model this as a multitask problem in hope that the interdependencies and close connectedness between the different tasks can mutually benefit each other to improve individual task performance. Prior works have modelled structured ABSA tasks like Aspect Sentiment Triplet Extraction (ASTE) and Aspect Opinion Pair (AO) extraction with a tagging framework, however, the tagging scheme is not best suited to handle overlapping triplets present in documents since a token can only have one tag, but can be part of multiple triplets. Recent works have moved towards a Seq2Seq framework to better model structured predictions tasks like ASTE (Yan et al. 2021) and as a result, we look towards a generative framework as our backbone. Existing body of work on ABSA develop efficient fine-tuning strategies to model the tasks. However, the proposed

frameworks do not specifically take into consideration the **contrast** between the positive and negative sentiment triplets. Different from these, we propose **CONTRABSA**, a novel pre-training strategy using **CONTR**astive learning to improve the performance of the downstream **ABSA** tasks. Given a sentence and its associated (aspect, opinion, sentiment) triplets, first, we design aspect-based prompts with the opinion terms masked. We then (pre)train an encoder-decoder architecture by applying contrastive learning on the aspect-aware sentiment representations of the masked terms as produced by the decoder. This is followed by task-specific fine-tuning or multi-task mixture-of-task training which we will discuss in the next steps. To the best of our knowledge, this is the first work proposing a pre-training strategy for ABSA tasks. Extensive experiments on the SemEval 14 datasets across the nine subtasks demonstrate the efficacy of our proposed pre-training framework as well as our multi-task prompt based fine-tuning framework. **We outperform the previous state-of-the-art on several tasks and additionally show how the proposed pretraining qualitatively affects the embedding space.**

Keywords: aspect-based sentiment analysis, contrastive learning, prompting

2. Introduction

Aspect-Based Sentiment Analysis is a broad and highly researched field consisting of nine fine-grained sentiment analysis subtasks. All of them involve efficiently extracting opinion terms of corresponding aspects or entities and subsequently deriving the sentiment expressed towards the aspects in review sentences. Aspect-level Sentiment Classification (ALSC) (Xue and Li 2018), (H. Yang et al. 2021), (Bai, Liu, and Zhang 2021) is one of the most popular tasks of this umbrella which involves classifying the sentiment polarities for every given aspect term in a sentence.

Before we get into the different extraction tasks, we formally define the different sentiment elements involved in ABSA:

- **Aspect term**: it is an opinion target which explicitly appears in the given text, e.g., "pizza" in the sentence, "*The pizza is delicious.*"
- **Opinion term**: it is the expression given by the opinion holder to express sentiment towards the target. For instance, "delicious" is the opinion term in the following example, "*The pizza is delicious*".
- **Sentiment polarity**: it describes the orientation of the sentiment over an aspect category or an aspect term, which usually belongs to positive, negative, and neutral. For example, in the above example, the sentiment expressed towards *pizza* is positive.
- **Aspect category** defines a unique aspect of an entity and is supposed to fall into a predefined category set C, for each specific domain of interest. For example, food and service can be aspect categories for the restaurant domains.

The ABSA tasks revolve around the different combination extraction of these sentiment elements. The different tasks can be broadly seen as a) Single tasks, that require extraction of single sentiment terms like aspect alone, and b) Compound tasks, that require join extraction of 2 or more sentiment terms. For example, the joint extraction of

aspects and the corresponding opinion terms is a task of Aspect and Opinion extraction (AO).

ABSA consists of nine subtasks which are described below:

Subtask	Input	Output	Task Type
Aspect Term Extraction(*AE*)	**S**	a_1, a_2	Extraction
Opinion Term Extraction(*OE*)	**S**	o_1, o_2	Extraction
Aspect-level Sentiment Classification(*ALSC*)	**S** + a_1	s_1	Classification
	S + a_2	s_2	
Aspect-oriented Opinion Extraction(*AOE*)	**S** + a_1	o_1	Extraction
	S + a_2	o_2	
Aspect Term Extraction and Sentiment Classification(*AESC*)	**S**	(a_1, s_1), (a_2, s_2)	Extraction & Classification
Pair Extraction(*Pair*)	**S**	(a_1, o_1), (a_2, o_2)	Extraction
Triplet Extraction(*Triplet*)	**S**	(a_1, o_1, s_1), (a_2, o_2, s_2)	Extraction & Classification

Figure 1: Illustration of seven ABSA subtasks.

- **Aspect Term Extraction**(AE): Extracting all the aspect terms from a sentence.
- **Opinion Term Extraction** (OE): Extracting all the opinion terms from sentence.
- **Aspect-level Sentiment Classification** (ALSC): Predicting the sentiment polarities for every given aspect terms in a sentence.
- **Aspect-oriented Opinion Extraction** (AOE): Extracting the paired opinion terms for every given aspect terms in a sentence.
- **Aspect Term Extraction and Sentiment Classificatio**n (AESC): Extracting the aspect terms as well as the corresponding sentiment polarities simultaneously.
- **Aspect and Opinion Pair Extraction** (AO): Extracting the aspect terms as well as the corresponding opinion terms simultaneously.
- **Aspect Sentiment Triplet Extraction** (ASTE): Extracting all aspects terms with their corresponding opinion and sentiment term.

- **Target Aspect Sentiment Detection** (TASD): Extracting all aspects terms, aspect categories and sentiment terms.
- **Aspect Category Opinion Sentiment** (ACOS): Extracting all aspects terms, aspect category, opinion and the corresponding sentiment.

Sent 1:	The film was good , but could have been better .
Triplets	[Aspect ; Opinion ; Sentiment] (1) film ; good ; positive (2) film ; could have been better ; negative
Sent 2:	The weather was gloomy , but the food was tasty .
Triplets	(1) weather ; gloomy ; negative (2) food ; tasty ; positive

Figure 2. The top half shows the interaction between aspects (in yellow) and opinions (in red). The bottom half is an example showing triplets consisting of **aspects**, **opinion** and **sentiment**.

Consider the examples shown in Fig. 2. In the first example, the sentiment associated with the aspect ***film*** changes depending on the corresponding opinion terms; ***good*** suggests a positive polarity, and ***could have been better*** indicating a negative polarity. So, simply extracting the pairs opinion pairs film-positive, and film-negative without additionally capturing the reasoning opinion phrases may not provide the complete picture. For the second sentence, the opinion term ***gloomy*** has a higher chance of being related with aspect ***weather***, than with ***food***. It is clear that the three elements of a triplet are strongly interdependent and it is important to model the aspect-opinion interactions during their extraction. This task itself gets challenging when there are multiple aspects in a sentence with contrasting sentiments. Moreover, a single aspect may have multiple opinions that have contrasting sentiments and such cases lead to overlapping aspects across triplets. The sentiments expressed may be product-dependent, target-dependent or ambiguous, and correctly linking aspect terms to their opinion terms becomes crucial for this task.

The earliest methods (Peng et al. 2020) to solve structured ABSA tasks like ASTE resorted to the BIOES tagging scheme to tag each token into one of the many classes.

Note that this scheme is not the most optimal way to solve overlapping triplets. Moreover, the tagging scheme doesn't inherently capture the opinion-aspect interactions which are crucial for ASTE and also results in error propagation. Recent successful methods have moved towards a tagging-free sequence to sequence approach that can better capture the interactions while also better capturing contrasting opinions towards a single aspect. Inspired by state-of-the-art methods, we model all of the ABSA subtasks as a template-based language generation problem. More specifically, we directly use Encoder-Decoder pre-trained models like T5 to generate the triplets that follow a specific template. It is to note that the template makes it easier to decode the triplets post generation.

Prior works have invariably focussed on solving one task out of the above discussed umbrella of ABSA tasks. While they have managed to obtain high performance in the individual tasks, they are not trained to solve the other ABSA tasks which are closely related. One exception is the Unified ABSA (Yan et al. 2021) paper which proposed a single framework that can handle the nine ABSA tasks, however the model has to be individually fine-tuned on the target ABSA task before it can be used which is not very desirable. We instead propose a prompt-based plug-and-play generative framework that can solve all the ABSA subtasks. More specifically, we design task-specific prompts and prepend it to the model inputs. Next, we pool the task-specific inputs of all the nine ABSA tasks and train the model in a mixture-of-tasks approach (Zhong et al. 2022). We conjecture that such multi-task training will help the model benefit from the close relatedness and interdependencies between the ABSA tasks and subsequently benefit each of the ABSA task.

Another gap we attempt to bring to light is the absence of pre-training methodologies in the ABSA literature. Although there have been several attempts in the past to develop efficient fine-tuning strategies (Zhang, Deng, et al. 2021), (Wu et al. 2020), (Mukherjee et al. 2021) we argue that a pre-training component would further enhance the aspect-sentiment or aspect-opinion understanding. The few works introducing

pre-training strategies cater just to a single ABSA task and do not generalize beyond that. We wish to alleviate this issue with a pre-training strategy that can benefit most ABSA sub-tasks if not all. Towards this end, we propose a **Prompt-Based Contrastive Pre-Training Strategy** that enhances the performance of all ABSA tasks when paired with our template-based generation strategy. We resort to contrastive learning to help the model comprehend the contrasting polarities between an aspect with positive and negative sentiment. We leverage the recently popular prompting techniques to lucidly extract **Aspect-Specific Sentence Embeddings** from a review sentence with multiple triplets to perform contrastive learning. We experimentally show that this elegant pre-training stage provides substantial gains.

3. Scope and Objectives

The *aim* of this thesis is to develop an efficient framework that can solve all the sub-tasks in Aspect Based Sentiment Analysis through a unified and multi-task framework. Moreover, we propose a pre-training strategy based on prompt-based Contrastive Learning that is expected to enhance the aspect-level sentiment understanding of the model. Our proposed prompt-based unified generative framework when paired with our pre-training setup outperforms the state-of-the-art on several of the ABSA tasks. We present our initial results in the next section.

Our main contributions in this work are as follows:

- We propose a novel unified generative scheme that can handle all of the nine ABSA sub-tasks. More specifically we propose a single model for all the ABSA tasks using task-specific prompts and mixture-of-tasks training. This is the first work to propose a single model to address all the ABSA tasks.
- We propose a novel strategy to pre-train an encoder-decoder framework for ABSA tasks. More specifically, given a sentence, we derive (possibly multiple)

aspect-based prompts with corresponding sentiments masked as shown in Table 1. The model is then pre-trained by applying supervised contrastive learning (SCL) on the decoder-generated aspect-level sentiment embeddings of the masked tokens as shown in Fig 5.We demonstrate that such as approach results in a better downstream performance of ABSA tasks such as ASTE, and Aspect Category Opinion Sentiment (ACOS) quad prediction, than performing SCL on sentence-level sentiment embeddings to pre-train the model, as in existing works. Also, different from prior works, we do not use any additional data to perform contrastive pre-training.

- We propose more generic prompt-based templates than the ones used in PARAPHRASE (refer to Table 2) to fine-tune our T5 encoder-decoder framework for the ASTE task.
- Our claims are based on extensive experimentation over SemEval-14 datasets and comparing our approach with popular benchmarks. We achieve the new state-of-the-art on several ABSA tasks. We also report ablation studies on the ASTE task to help better understand the impact of the components in our framework. Our code is released here for the benefit of research [1]

4. Literature Review

4.1. Transformers

Transformers was first proposed by (Vaswani et al. 2017) and it has since transformed the field of NLP. The proposed ideas replaced the ever-popular LSTMs by allowing parallel processing of texts. They claimed that it was no longer needed to pass text sequentially into a model, and replaced it with the parallel encoding of texts. With this, it was possible to train the model on a very large volume of text much faster than

[1] https://github.com/nitkannen/MultiABSA

before. They additionally paired it with a novel idea called Attention that transformed the world of NLP. In simple terms, attention allowed variable weighting of contextual words within a sentence. Below, we look at the basic architecture of the Transformer and refer interested readers to the original paper for more details.

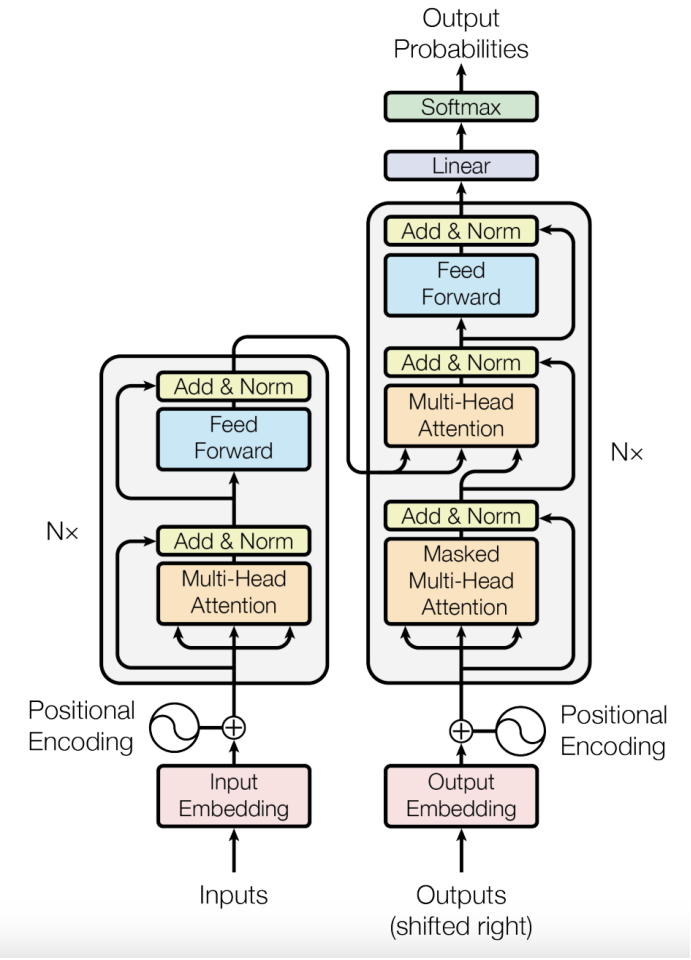


Figure 3. Transformer architecture

4.1.1 Architecture

Encoder: The encoder is composed of a stack of N = 6 identical layers. Each layer has two sub-layers. The first is a multi-head self-attention mechanism, and the second is a simple, position-wise fully connected feed-forward network. A residual connection is employed around each of the two sub-layers, followed by layer normalization.

Decoder: The decoder is also composed of a stack of N = 6 identical layers. In addition to the two sub-layers in each encoder layer, the decoder inserts a third sub-layer, which performs multi-head attention over the output of the encoder stack. Similar to the encoder, they employ residual connections around each of the sub-layers, followed by layer normalization. They also modify the self-attention sub-layer in the decoder stack to prevent positions from attending to subsequent positions. This masking, combined with the fact that the output embeddings are offset by one position, ensures that the predictions for the position i can depend only on the known outputs at positions less than i.

4.1.2 Attention

An attention function can be described as mapping a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors. The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key. The expression is given as:

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

4.1.3 Language Models

A language model in NLP is a probabilistic statistical model that determines the probability of a given sequence of words occurring in a sentence based on the previous words. It helps to predict which word is more likely to appear next in the sentence. A language model is trained on a large corpus of text, typically millions of tokens. A very widely used language model BERT (Devlin et al. 2019) is trained using Masked Language Modelling (MLM), where random tokens are masked out and the model is

trained to fill in the masks. This way the model is expected to learn the context of the language. In our work, we use T5 (Raffel et al. 2020) as our base model, another popular language encoder-decoder language model that is suitable for sequence-to-sequence tasks.

4.1.3.1 T5 (Encoder-Decoder Transformer model)

T5 (Raffel et al. 2020) is a transformer-based language model that was introduced by Google AI in 2019. It is a versatile model that can perform a wide range of natural language processing tasks, including text summarization, question answering, and machine translation. The name T5 stands for Text-to-Text Transfer Transformer. T5 was trained using a massive dataset of text sources including web pages and books. The pre-training process involved using a task-agnostic objective function called "C4" that encouraged the model to learn a generalized representation of language. The input and output of the model are both sequences, irrespective of the task. This is achieved by prefixing the inputs with appropriate prompts. In our case, our base model is initialised with the T5 checkpoint.

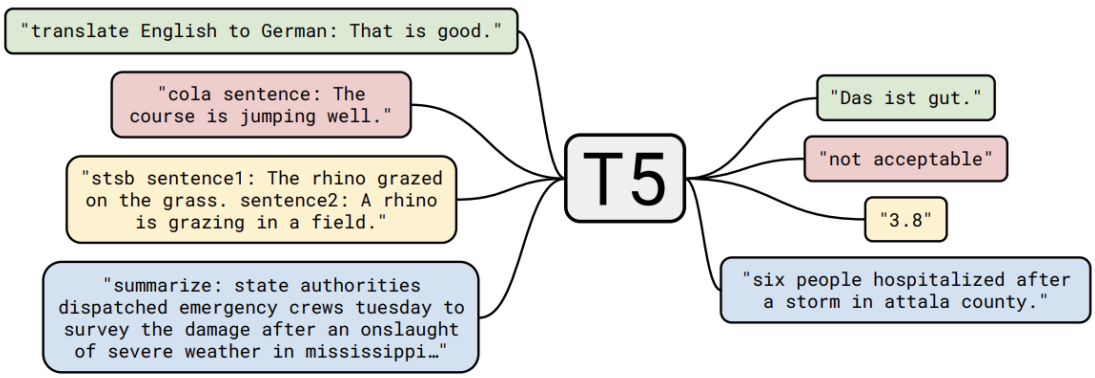


Figure 4. Diagram representing the text-to-text framework used to train T5

4.2. Aspect-Based Sentiment Analysis

As discussed earlier, Aspect Based Sentiment Analysis is used to refer to a broad umbrella of nine tasks. AE, OE, AOE, AESC, AO, ASTE, TASD and ACOS are extraction tasks, while ALSC is a classification task. Naturally, ALSC and ASTE are the most studied subtasks in the ABSA literature. Some of the recent ABSA works include (Kamila et al. 2022), (Liang et al. 2022), (Chen et al. 2022), (Z. Li et al. 2021). A survey of the ABSA landscape including the different approaches adopted is detailed in (Zhang et al. 2022).

4.2.1. Modelling Paradigms

Before describing specific ABSA tasks and their solutions, we first introduce several mainstream natural language processing (NLP) modelling paradigms that are commonly employed for ABSA tasks, including Sequence-level Classification, Token-level Classification, Machine Reading Comprehension (MRC), and Sequence-to-Sequence modelling (Seq2Seq).

4.2.1.1. Sequence-level Classification

For the sequence-level classification, a model typically first feeds the input text X into an encoder Enc(·) to extract the task-specific features, followed by a classifier CLS(·) to predict the label Y:

$$Y = CLS(Enc(X)),$$

where Y can be represented as one-hot or multi-hot vectors. Typically, in Natural Language Processing, the encoder is a Transformer model described in section 4.1.

4.2.1.2. Token-level Classification

In contrast to the sequence-level classification that assigns the label to the whole input text, token-level classification (also referred to as sequence labelling or sequence tagging) assigns a label to each token in the input text. It also first encodes the input text into contextualized features with an encoder Enc(·) and subsequently predicts a label y for each token x.

4.2.1.3 Machine Reading Comprehension (MRC)

The MRC paradigm (Zeng et al. 2020) extracts continuous text spans from the input text X conditioned on a given query Xq. Therefore, ABSA methods with the MRC paradigm need to construct a task-specific query for the corresponding task, i.e., a query denoting what is the desired information. For example, Xq can be constructed as "What are the aspect terms?" in the AE task. The original text, as well as the constructed query can then be used as the input to a MRC model to extract the text spans of aspect terms.

4.2.1.4. Sequence-to-Sequence (Seq2Seq)

The sequence-to-sequence (Seq2Seq) framework takes an input sequence X = {x1, ..., xn} as input and aims to generate an output sequence Y = {y1, ..., ym}. A classical NLP application with such a paradigm is the machine translation task (S. Yang, Wang, and Chu 2020). It is also used for solving ABSA tasks, e.g., directly generating the label sequence or desired sentiment elements given the input sentence. Taking the AE task as an example, X can be "The fish dish is fresh", and Y can be "fish dish" in the natural

language form. It typically adopts an encoder-decoder model such as Transformer. Recent works in the ABSA domain have resorted to Seq2Seq approaches for modelling the problem owing to several reasons. Tagging-based approaches discussed above do not allow the prediction of overlapping triplets or pairs. Moreover, tagging-based approaches do not capture the inherent relationships between aspect and opinion terms. (Zhang, Li, et al. 2021) is one of the first works to look into this direction in the context of ABSA.

4.2.2. ABSA Baselines

Structured extraction tasks like ASTE, TASD and ACOS are the most challenging tasks of the lot and are of great interest to the research community owing to the fact that it necessitates the extraction of complete triplet information from the text. ASTE literature can be broadly categorized into tagging-based and sequence to sequence-based:

Tagging Based: (Peng et al. 2020) propose a two-stage pipeline tagging framework using the BIEOS[2] scheme. (Tay, Luu, and Hui 2017a) propose a multitasking framework to jointly detect aspects, opinions and sentiments. (Tay, Luu, and Hui 2017b) propose a novel position-aware tagging scheme extending the BIEOS tags. (Wu et al. 2020) propose a novel grid-tagging scheme to address the limitations of conventional tagging

Seq2SeqBased: (Mukherjee et al. 2021) is one of the very first tagging-free approaches that utilized pointer networks for decoding. (Yan et al. 2021) propose a unified generative framework using BART for all of the 7 subtasks in ABSA. (Zhang, Li, et al. 2021) and (Zhang, Deng, et al. 2021) are template-based generation approaches closer to our work.

Likewise, for TASD, there have been several attempts in the community with innovative approaches (Lv et al., n.d.) and (Tianhao Gao et al. 2022). A very recent and interesting

[2] BIOES: begin, inside, outside, end and single

direction is proposed by (Mao et al. 2022) where they look at generating sentiment tuples as paths of a tree. ACOS involves the category term along with the sentiment terms involved in ASTE. Although the modelling techniques of ASTE can be extended as it is to predict the category term along with the triplets, there have been works specifically addressing the ACOS task (Zhang, Deng, et al. 2021).

4.3. Contrastive Learning

Contrastive Learning has seen a huge resurgence in recent years in the research community. (Khosla et al. 2021) proposed the very first formulation of supervised contrastive learning and showed how it is competitive to and at times even outperforms the conventional cross-entropy loss function. Since then contrastive learning has become a popular term in the AI community and has also found its way into NLP Research. A variety of successful approaches have used contrastive learning for Text Generation (Lee, Lee, and Hwang 2021), Machine Translation (Pan et al. 2021), and Summarization (S. Xu et al. 2022). Contrastive learning has been used for simple classification ABSA tasks (Z. Li et al. 2021), however, it has not been experimented on more complex tasks like triplet extraction prior to this work. In this work, we show how contrastive learning can be leveraged for more complex structured prediction tasks like ASTE, TASD and ACOS. Such an approach can be extended to similar tasks in other domains as well.

4.4. Prompt-Based Learning

Prompting (Schick and Schütze 2021) is yet another recent technique that has gained popularity. Prior to this, the task was treated as a black box that a model has to learn using the given data samples. Prompting is a simple technique where the task or the target labels are explained to the model in natural language terms (Sanh et al. 2022).

More specifically, consider a simple sentiment classification task trained using an encoder. The model is trained to predict 0 (negative) or 1 (positive) by minimizing the cross-entropy loss function. The model (here, an LLM) is not aware of the task it is being trained to solve. On the other hand, in the prompting paradigm, you prompt the model with an appropriate question that it learns to answer in the Natural Language (NL) form. For example, the question "Is the sentiment positive or negative?", "Is it good or bad?" and, "Was the sentiment good?", are some of the plausible prompts that can be used for the task of sentiment classification. The answer is given by the model in Natural language (NL) form depending on the question prompt. More details on prompt-based learning can be found in (Liu et al. 2023). A variety of successful few-shot learners have used prompts-based learning (Shin et al. 2020), (Tianyu Gao, Fisch, and Chen 2021). Prompting techniques are yet to be experimented with in the ABSA domain.

5. Background

5.1 Contrastive Learning

Supervised Contrastive Learning was first proposed by (Khosla et al. 2021) and has since been an active topic of research. In essence, contrastive learning tries to maximize the distance between the positive and negative classes. It tries to pull the two classes far apart so it becomes easier for the model to linearly separate or distinguish them. In sentiment analysis, contrastive learning may be used to highlight the opposing sentiment polarities between positive and negative sentences, thus pulling them apart in the embedding space in order to make them easily separable. It is supervised because of the labels.

Specifically for $\{x_i, x_j\}$ within a batch B of N samples, we first obtain the sentence representation Z using which we define the contrastive loss on batch B as

$$\textit{Contrastive Loss} = -\log \frac{exp\,(sim\,(Z_i, Z_j)/\tau)}{\sum_{k \in B,\, 1\,[i\,!=k]}^{N} exp\,(sim\,(Z_i, Z_k)/\tau)} \quad \text{- eqn 1}$$

Note that contrastive learning is performed at the sentence level using the sentence representation and the corresponding label. It is not trivial to use this in our case as the sentiment labels are fine-grained (aspect specific) and not at the sentence level as shown in Fig. 5.

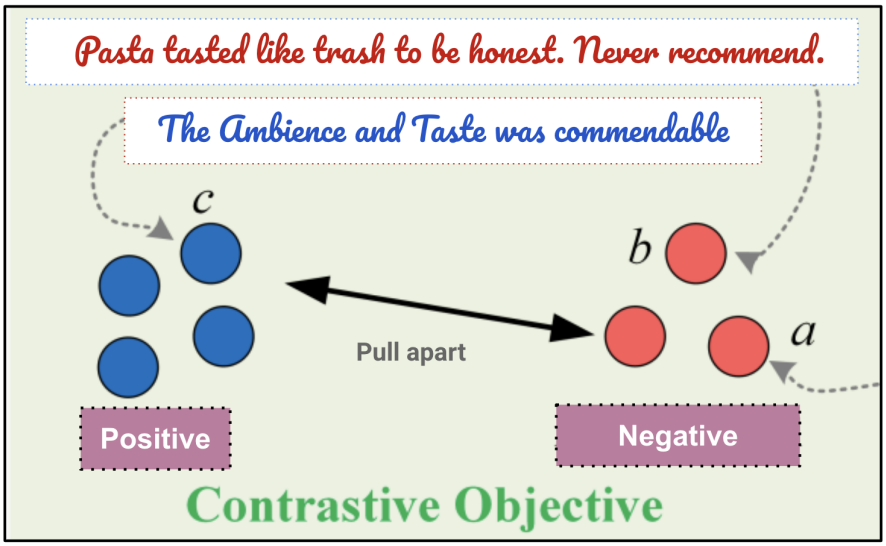


Figure 5: A representation of the embedding space after contrastive learning

6. Methodology

In this section we describe the proposed **CONTRABSA** framework in detail. We use a pre-trained encoder-decoder T5 (Raffel et al. 2020) model as the backbone of

CONTRABSA. Note that T5's pre-training consists of text-to-text tasks like summarization, question answering etc. We first describe the prompt-based contrastive pre-training strategy, followed by the template-generation templates and finally discuss the mixture-of-tasks approach to train a unified and multi-task ABSA model. Finally, we describe the training strategy and the training objective that is optimized.

6.1. Base Model

We use the T5 (Raffel et al. 2020) model as the backbone of CONTRABSA. As explained earlier, T5 is a language model trained for a variety of tasks like Question Answering, Text Summarization etc., using a Sequence-to-Sequence approach. In our case, we begin our finetuning process with existing T5 weights that are publicly released.

The input sequences are processed by an encoder, which generates a fixed-size vector representation (sometimes called a context vector) that captures the meaning of the input sequence. The size of the context vector is a hyperparameter that can be adjusted depending on the complexity of the task. The output sequences are processed by a decoder, which takes in the context vector and generates the output sequence one token at a time using an autoregressive approach, as described in my previous answer. The size of the decoder's hidden state is also a hyperparameter that can be adjusted depending on the complexity of the task. During decoding, the model generates a distribution over all possible tokens at each time step. This distribution is typically represented as a vector with a length equal to the size of the vocabulary. The model then selects the most likely token based on this distribution, as determined by a softmax function.

To handle sequences of different lengths, padding is used. Padding involves adding special tokens to the beginning or end of a sequence to make it a fixed length. For example, if the maximum sequence length is set to 100, a shorter input sequence might

be padded with special tokens at the end to make it 100 tokens long. During training and inference, the model takes in a batch of input sequences and a batch of output sequences. The input sequences are tokenized into sequences of integers, where each integer represents a unique token in a vocabulary. The output sequences can also be tokenized in the same way.

The T5 tokenizer uses byte pair encoding (BPE), which is a type of subword tokenization that can handle out-of-vocabulary (OOV) words by breaking them down into smaller subword units. The T5 tokenizer works by first splitting the input text into words or subwords. Each word or subword is then encoded as a sequence of integers using BPE. The BPE encoding process starts with a vocabulary of single characters and pairs of characters that occur frequently in the training data. During training, the most frequent pairs of characters are merged into new subword units, which are added to the vocabulary. This process continues until a maximum vocabulary size is reached. During decoding, the T5 tokenizer takes in the integer-encoded input sequence and generates the corresponding text output sequence. The output sequence can also be tokenized using the same tokenizer, allowing the model to handle a wide range of text-to-text tasks.

One notable feature of the T5 tokenizer is that it can handle multiple text-to-text tasks using the same input-output format. For example, the input might be a question and the output might be an answer, but the same input-output format can be used for tasks such as summarization, translation, or text completion. This allows T5 to be fine-tuned on a variety of NLP tasks using the same input-output format, making it a powerful and flexible tool for NLP. We leverage this feature by adding special tokens to the T5 vocabulary which is detailed in the coming sections. The above form the motivation for our choice of pre-trained language model.

6.2. Prompt-Based Contrastive Pre-training

Table 1. Pre-training prompts used for prompt-based contrastive learning

Sentence	Pre-training Prompt (<aspect> *aspect* [MASK])
The food was good.	<aspect> **food** **[MASK]**
Both sound as well as display quality are great.	<aspect> **sound** [MASK] <aspect> **display quality** [MASK]
While the sushi was tasty, the ambience sucked.	<aspect> **sushi** [MASK] **ambience** [MASK]

As discussed earlier, supervised contrastive learning attempts to pull the sentence representation of opposite classes farther apart and attempts to push the same classes closer to each other in the embedding space. Note that this is possible at a sentence level if the complete sentence has a single label. However, in tasks like ASTE, there is no single label for the complete review sentence, in fact, the sentiment label is present at the aspect level and is contrasting most of the time. Hence, the sentence representation that is needed to perform sentiment-based contrastive learning for ASTE has to be aspect specific in nature.

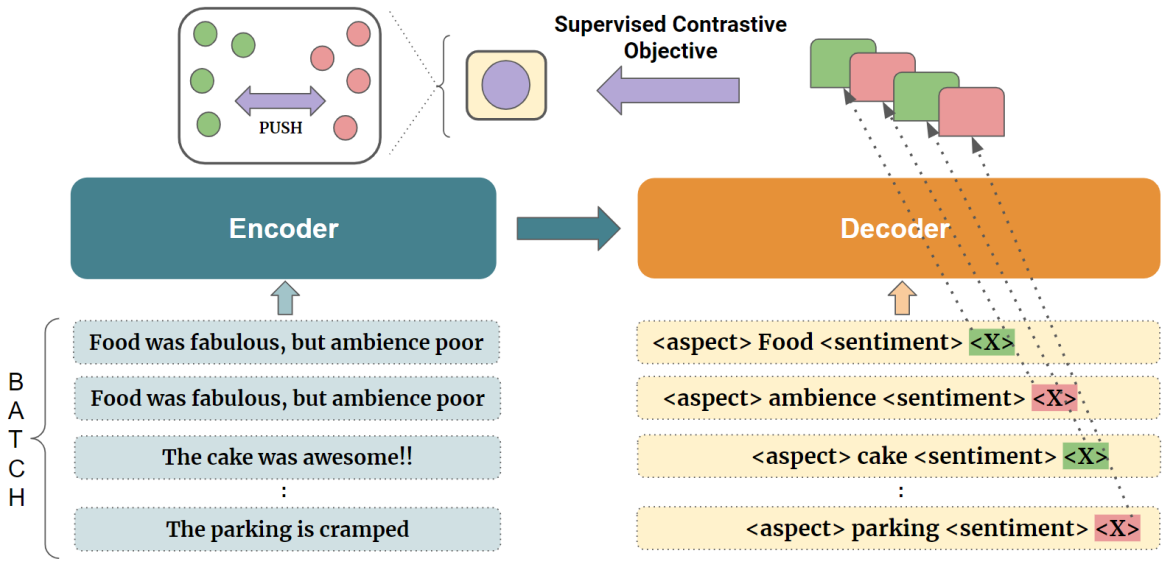


Figure 6. A representation of the proposed prompt-based contrastive pre-training framework.

We leverage the recent developments in prompting technique to propose a novel, simple and elegant strategy to obtain aspect-level sentence representation. Concretely, we pass the original review sentence on the encoder side and subsequently feed a prompt that is aspect-specific to the decoder side. The model now has to consider the input review sentence and the fed-in aspect-specific prompt in order to generate a representation that would be used for contrastive learning. Note that now the combination of input + prompt will have a single sentiment label. The prompt is followed by a [mask] token whose embeddings are taken as sentence representation. This representation makes Z_i in eqn 1. It is interesting to note that we would get multiple examples from a single sentence. More precisely, the number of samples for contrastive learning would now be the same as the number of aspects in the review sentences that do not have contrasting opinion polarities. The decoder side is prompted with the aspects while masking the sentiment label. The [mask] token embedding thus captures the required sentiment and is taken as the representation. Table 1 shows the examples of sentences and the corresponding masked prompts used for pre-training.

Note that the prompt is used to obtain the sentence-level representation which is now captured by the [mask] token. Please refer to Fig. 6 for a visual representation of the process.

6.3. Template-Based Generation

Inspired by the recent success of generative approaches to model ASTE and Information Extraction (Huguet Cabot and Navigli 2021) and (Zhang, Deng, et al. 2021),
we model all the sub-tasks in ABSA using a template-based generation strategy. Concretely, given the input sentence and the corresponding task-prompt, the decoder is expected to autoregressively generate the elements (aspects in case of AE, triplets in case of ASTE etc.,). In order to unify the model input/output space, we constrain the deoder to output the elements following a specific template. PARAPHRASE (Zhang, Deng, et al. 2021) is a previous work that used Natural language (NL) paraphrases to tackle the ASTE problem. More specifically, they trained the decoder to produce sentences as shown in below Table 3.

We argue that the templates proposed by PARAPHRASE are sub-optimal owing to several reasons. i) The templates will have to be tailored and changed for the different ABSA tasks, ii) The templates do not generalize to all samples within the dataset. For example, iii) Fixed NL templates are not language-agnostic, i.e., a multilingual extension of the work would require hand-crafting sensible prompts in the target languages.

Table 2. Templates used by PARAPHRASE (Zhang, Deng, et al. 2021) for fine-tuning

Sentence	**PARAPHRASE template** (It is great/ok/bad because ASPECT is OPINION)
The food was good.	It is great because food is good
I was so disappointed with the chef.	It is bad because **chef is so disappointed**

We propose a novel template to counter the above shortcomings. Table 3. shows the template we propose for a few of the ABSA tasks. We use special tokens <triplet>, <opinion>, <sentiment> and <aspect> as placeholders for corresponding elements. We use the same decoding algorithm as described in (Huguet Cabot and Navigli 2021) for ASTE and extend it intuitively for the other ABSA tasks. We additionally introduce 2 complementary tasks, namely the Opinion Term Detector (OTD) and the Triplet Count Estimator (TCE) as shown in Fig. 6 that we describe in the next section.

Table 3. Showing the proposed template used for the different ABSA subtasks. The reference input sentence used here is *The cake was too creamy but the coffee was nice.*

Task	Proposed Template (Ours)
Aspect Extraction (AE)	<aspect> **cake** <aspect> **coffee**
Opinion Extraction (OE)	<opinion> **too creamy** <opinion> **nice**
Aspect and Opinion Pair Extraction (AO)	<pair> **cake** <opinion> **too creamy** <pair> **coffee** <opinion> **nice**
Aspect Term Extraction and Sentiment Classification (AESC)	<aspect> **cake** <sentiment> **negative** <aspect> **coffee** <sentiment> **positive**
Aspect Sentiment Triplet Extraction (ASTE)	<triplet> **cake** <opinion> **too creamy** <sentiment> **negative** <aspect> **coffee** <opinion> **nice** <sentiment> **positive**

6.4. Auxiliary Tasks (MTL)

After being pre-trained, the encoder-decoder framework now needs to be fine-tuned for the ASTE task. For this, corresponding to each sentence x being passed as input to the

encoder, first we construct the target sequence y to be generated by the decoder. y is constructed to follow the proposed target template introduced in the previous section.

Let e denote the encoder-generated contextualized representation of x. At the i^{th} time step, the decoder output y_i = D(e, $y_{<i}$) is computed based on e and the previous outputs $y_{<i}$. Probability distribution for the next token is obtained as:

$$p_\theta(y_{i+1}|e, y_{<i+1}) = softmax(W^T y_i)$$

Here, θ is initialized with parameter weights obtained after pre-training the model using contrastive learning. W maps y_i to a logit vector which is then used to calculate the probability distribution over the whole vocabulary set. It is to be noted here that the tokens <aspect>, <opinion>, and <sentiment> are added to the vocabulary at the time of training, and their embeddings are learnt from scratch. Finally, the model parameters are fine-tuned on the input-target pairs by minimizing the negative log-likelihood denoted as follows:

$$\mathcal{L}_{ED} = -log\ p_\theta(y|e) = -\sum_{i=1}^{n} log\ p_\theta(y_i|e, y_{<i})$$

As shown in Fig. 7 , we include the following two auxiliary objectives to further improve the ASTE performance, as detailed below:

6.4.1. Opinion Term Detection (OTD)

The motivation behind including this module comes from (Mrini et al. 2022) where the authors introduce a similar module called entity mention detection as an auxiliary task to improve the performance of the main task of encoder-decoder autoregressive entity linking. We hypothesize that the opinion term detection (OTD) module will help to

better detect the opinion spans boundaries which in turn affects the sentiment prediction. As depicted in Figure 7, we formulate OTD as a sequence-tagging task as per the BIO scheme.

Formally, for each token $tok_i \in x$ the opinion tagger takes as input the contextualized token embedding generated by the encoder, and performs a 3-way classification task with the classes being B - beginning of the span, I - inside the span, O - outside the span. The module is trained by minimizing the Cross-Entropy loss L_{OTD} between the true and predicted labels. Our ablation results further justify the importance of this module.

6.4.2. Triplet Count Estimation (TCE)

While fine-tuning the encoder-decoder framework for the ASTE task, the target sequences are expected to explicitly guide the decoder on how many triplets to generate. However, to augment this process, we introduce the module of triplet count estimation(TCE) to implicitly guide the decoder on when to stop. It is a simple regressor, consisting of a fully connected layer, that takes the encoder-generated sentence embedding e as input and is trained to predict the number of triplets associated with x by minimizing the Mean Squared Error loss L_{TCE}. Again, our ablation results reported in analysis experimentally justify the advantage of this module.

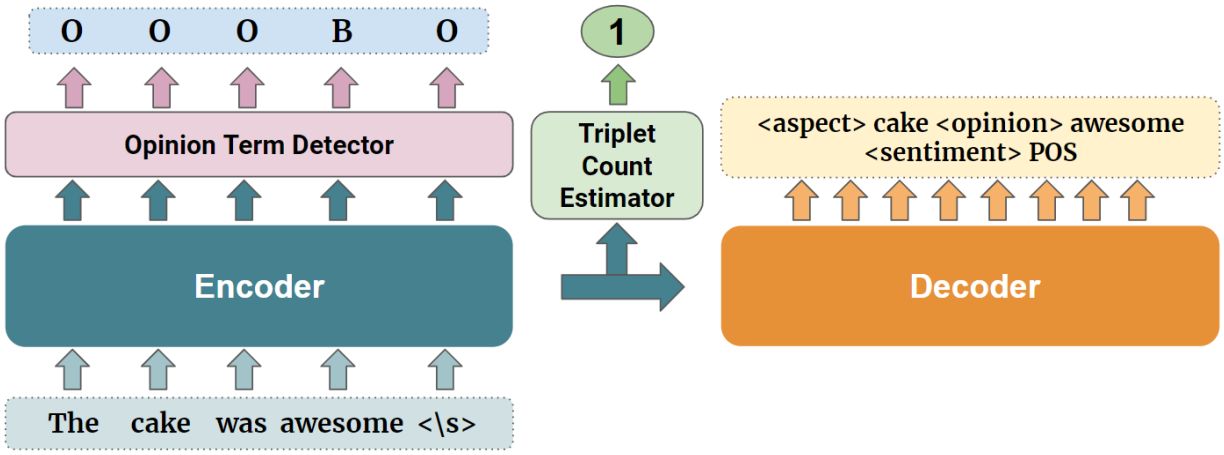


Figure 7. A representation of the proposed multi-task finetuning framework. Note the 2 complimentary tasks TCE and OTD.

6.4.3. Joint training

Our base model, is trained by optimizing the encoder-decoder loss L_{ED} only. The full model, is jointly trained in a multi-task setup by minimizing the combined loss:

$$\mathcal{L} = \mathcal{L}_{ED} + \alpha \cdot \mathcal{L}_{OTD} + \beta \cdot \mathcal{L}_{TCE}$$

α and β are the weight coefficients assigned to the OTD loss L_{OTD}, and TCE loss L_{TCE} respectively.

6.5. Mixture-of-Tasks ABSA Training

In order to train the model for all the ABSA tasks simultaneously, we resort to a mixture of tasks approach (Su et al. 2022) enabled by defining task-specific prompts. More specifically, we prepend task specific special tokens such as <ae> , <pair> and <triplet> to the model inputs that imply to the model which specific ABSA task it is required to perform for the given input as shown in Fig. 8. Note that this way modelling implicitly makes it multi-task – trained to perform multiple tasks. Multi-task learning is a popular

method of choice in the NLP literature when the tasks in consideration are highly related. The close interconnectedness and relationship between the tasks is said to mutually benefit each other, thus resulting in increased performance. In our case, such a modelling approach ensures we have a single model capable of performing all nine ABSA tasks – which is one of our overarching motivations.

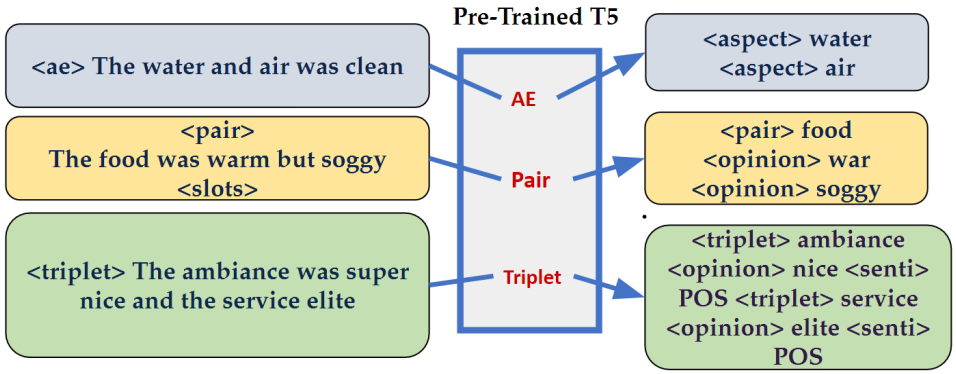


Figure 8: A visual representation of the mixture-of-tasks training using task prompts. For easier visualisation, we only show 3 of the 9 ABSA tasks in this figure.

7. Experimental Setup

Pre-Training: For this, we combine the train data from all four ASTE-DATA-V2 datasets (refer to Table 4) to prepare our pre-training datasets. This results in a total of 5,039 train data points from 3,634 sentences. Please note that we do not need test data in the pre-training phase. Also, different from (Z. Li et al. 2021)we do not use any external data for performing supervised contrastive pre-training. Pre-trained[3] was used to initialize the model weights. We (pre)train the T5 encoder-decoder framework for 14 epochs using AdamW optimizer (Loshchilov and Hutter 2019) with a learning rate of 2e-7 and a batch size of 16. The temperature parameter Γ was set to 0.07.

Fine-Tuning: Our Base model contains around 222 million trainable parameters.

[3] https://huggingface.co/t5-base

For each of the datasets, we respectively fine-tune the pre-trained model weights for the downstream ASTE task using AdamW optimizer with a learning rate of 1e-4 for 14Res and 16Res, and 3e-4 for 15Res and Lap14. A batch size of 16 was used for all datasets. Following (Mrini et al. 2022), we optimize the auxiliary task weights, α (OTD), and β (TCE) for each dataset.

7.1 Baselines

We compare the proposed framework with a variety of successful approaches proposed for ABSA. More specifically, we compare our work with several ASTE, AO, AESC, TASD and ACOS baselines. Some of the most relevant baselines are listed below:

- **PASTE + BERT** (Mukherjee et al. 2021) propose the first tagging-free encoder-decoder approach for ASTE using pointer networks. (ASTE)
- **Unified ABSA** (Yan et al. 2021) propose a unified generative framework for ABSA using BART as the backbone. Note that this differs from our work as they propose a unified framework which has to be trained individually for the different ABSA tasks, while we propose a single model that can perform all ABSA tasks. (ASTE, AO)
- **GAS** (Zhang, Li, et al. 2021) propose a simple and effective language generation technique for ASTE. The triplets are generated sequentially without placeholders. (ASTE, AO).
- **PARAPHRASE** (Zhang, Deng, et al. 2021) propose a novel perspective to ASTE by modelling it as a paraphrasing problem. We refer readers to the original paper for details. (ASTE).
- **LEGO-ABSA** (Tianhao Gao et al. 2022) propose a unified generative multi-task framework that solves multiple ABSA tasks by controlling task prompts. This also allows transferring to harder tasks by gathering task prompts like Lego bricks.

- **EHG-Para** (Lv et al., n.d.) uses an Efficient Hybrid Transformer to generate the locations and semantic information of ABSA targets in parallel.
- **SentiPrompt** (C. Li et al. 2021) uses sentiment knowledge-enhanced prompts to tune the language model in the unified framework. They report performance on the AESC task.
- **Seq2Path** (Mao et al. 2022) proposes a very interesting and novel approach to representing sentiment tuples as paths of a tree. They report performance on ASTE, TASD and ACOS.

7.2 Datasets

For the English language we use the ASTE-Data-V2 released by (L. Xu et al. 2020). The triplets are annotated from the original datasets released in SemEval 14 (Pontiki et al. 2014). The dataset consists of reviews from 2 domains: laptops and restaurants. The restaurant domain has 3 variants which we refer to as 14res, 15res and 16res. The laptop dataset is lap14. The data contains a train set, a dev set and a test set. All results reported here are by training on train split, model selection and hyperparameter tuning on dev, and finally evaluated on the test split. The datasets statistics for the ASTE task is reported in the Table 4. We use the recently released extension of ASTE datasets for ACOS (Zhang, Deng, et al. 2021) and TASD (Wan et al. 2020).

Table 4. Dataset statistics for the ASTE task.

Datasets		#S	POS	NEU	NEG
Lap14	Train	906	817	126	517
	Dev	219	169	36	141
	Test	328	364	63	116
14Res	Train	1266	1692	166	480
	Dev	310	404	54	119
	Test	492	773	66	155
15Res	Train	605	783	25	205
	Dev	148	185	11	53
	Test	322	317	25	143
16Res	Train	857	1015	50	329
	Dev	210	252	11	76
	Test	326	407	29	78

7.3 Experiments

We refer to our pre-trained model as **CONTRABSA**. We refer to the fine-tuning setup with the auxiliary tasks initiated from the pre-trained checkpoints as **CONTRABSA-MTL**. Note that for single tasks like AE and OE, we leave out the MTL tasks. Finally, we train the model for several ABSA tasks at once and call our final model **CONTRABSA-MTL-MoT**.

We conduct extensive experimentation using our proposed model comparing it with several baselines and report the results of the same. Our experiments are as follows:

- **Comparison with Supervised Contrastive learning approaches:** We compare our prompt-based pre-training strategy with existing pre-training approaches used in the ABSA domain (Z. Li et al. 2021). For a fair comparison, we pair the same finetuning strategy (our proposed) with the different pre-training methods.
- **Comparison with different Fine-tuning templates**: We compare our proposed template with the existing PARAPHRASE template.

- **Proposed Model experiments (CONTRABSA-MTL-MoT)**: experiments on SemEval 14 dataset trained for AE, OE, AO, ASTE, TASD and ACOS tasks. We compare the performance of our approach with baselines for the different tasks.
- **Qualitative Analysis**: We conduct an intensive analysis of our model to understand how our proposed components improve the model's performance.

7.4 Evaluation Metrics

Like all previous works on ABSA, we report the precision, recall and micro-f1 as the evaluation metrics. Note that a predicted pair is correct only if both components match exactly and a triplet is considered correct only if all three components match with that of the gold labels.

8. Results

Table 5. Results obtained using the proposed fine-tuning strategy to compare with the best existing template and the best existing pre-training strategy.

Model	14res	15res	16res	lap14
PARAPHRASE	0.715	0621	0.719	0.605
ASTE-Base	0.720	0.634	0.722	0.608
-W/SCL-Sent	0.722	0.645	0.724	0.611
CONTRASTE-MTL	0.728	0.648	0.730	0.614

Table 6. Results obtained by our proposed model on the mono extraction tasks - aspect extraction (AE) and opinion extraction (OE)

Task	Model	14res			15res			16res			lap14		
		P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
AE	CONTRABSA	0.891	0.808	0.847	0.912	0.84	0.875	0.841	0.87	0.855	0.895	0.884	0.889
	CONTRABSA - MoT	0.914	0.819	0.864	0.916	0.9	0.908	0.869	0.91	0.889	0.904	0.912	0.908
OE	CONTRABSA	0.868	0.94	0.903	0.78	0.824	0.801	0.867	0.90	0.883	0.855	0.938	0.895
	CONTRABSA -MoT	0.88	0.945	0.911	0.78	0.84	0.81	0.89	0.90	0.90	0.86	0.942	0.899

Table 7. Results obtained by our proposed model on pair and triplet extraction tasks - aspect and opinion extraction (AO) and aspect sentiment triplet extraction (ASTE).

Task	Model	14res			15res			16res			lap14		
		P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
AO (Pair)	GAS	**-**	**-**	0.687	-	-	0.650	-	-	0.705	-	-	0.626
	Unified ABSA	**-**	**-**	0.768	-	**-**	0.67	**-**	**-**	0.703	**-**	**-**	0.661
	CONTRABSA - MoT	0.674	0.615	0.643	0.719	0.705	0.712	0.684	0.725	0.704	0.727	0.67	0.697
ASTE (Triplet)	PASTE + BERT	0.648	0.638	0.643	0.583	0.567	0.575	0.655	0.644	0.65	0.55	0.516	0.532
	Unified ABSA	0.655	0.65	0.652	0.591	0.593	0.592	0.666	0.686	0.676	0.614	0.561	0.587
	GAS	0.65	0.695	0.672	0.561	0.618	0.588	0.661	0.687	0.674	0.571	0.54	0.551
	PARAPHRASE	0.725	0.715	0.72	0.614	0.643	0.628	0.702	0.735	0.718	0.618	0.618	0.613
	CONTRABSA	0.722	0.732	**0.727**	0.646	0.670	0.658	0.696	0.763	0.728	0.636	0.617	0.622
	CONTRABSA - MoT	0.721	0.732	0.726	0.644	0.681	**0.662**	0.7	0.77	**0.734**	0.636	0.618	**0.623**

Table 8. Results obtained by our model on the aspect sentiment classification (AESC), target aspect sentiment detection (TASD) and aspect category opinion sentiment (ACOS).

Task	Model	14res	15res	16res	lap14
		F	F	F	F
AESC	Span-BART	68.17	78.47	69.95	75.69
	GAS-R	65.87	79.06	68.82	75.73
	SentiPrompt	**70.79**	81.09	**74.2**	79.81
	CONTRABSA	70	**82.6**	72.1	**81**
	CONTRABSA-MoT	71.3	**82.6**	72.9	**83.1**
TASD	GAS-R	-	60.06	67.7	-
	EHG-Para	-	62.83	**72.09**	-
	CONTRABSA	-	**65.4**	68.6	-
	CONTRABSA-MoT	-	**66.3**	68.7	
ACOS	PARAPHRASE	46.8	57.8	59.2	42.9
	CONTRABSA	**47.8**	**59.8**	**60.5**	**44.6**
	CONTRABSA-MoT	**47.9**	**60.2**	**61.2**	**45.7**

Table 9 Ablation study on the ASTE task to highlight the component contributions

Ablation	14res			15res		
	P	R	F1	P	R	F1
Ours (CONTRABSA-MTL-MoT)	0.721	0.732	0.726	0.644	0.681	0.662
Ours -w/Mixture-of-tasks	0.722	0.732	0.727	0.646	0.670	0.658
Ours -w/ Multi-Task Learning	0.713	0.724	0.718	0.632	0.668	0.65
Ours -w/Contrastive Pre-T	0.69	0.697	0.698	0.613	0.666	0.638

9. Discussion

9.1 Main Results:

In Table 5, we wanted to compare our pre-training strategy with the one used in existing ABSA works. While (Z. Li et al. 2021) use SCL for pre-training, it is performed for a simpler task of ALSC. However, different from us, it applies SCL to sentence-level sentiment representations of sentences. In order to replicate their methodology, we pre-train our encoder-decoder framework by applying SCL on mean-pooled representations of sentences from the final layer of the encoder. For this, we collected a total of 3358 data points} (sentences containing triplets with the same sentiment polarity) from 3,634 sentences combining all four train datasets. Model weights were initialized with pre-trained t5-base, and the framework was (pre)trained for 14 epochs using AdamW optimizer with a learning rate of 2e-5 and a batch size of 16

We fine-tune ASTE-base from this pre-trained checkpoint respectively for each of the datasets and report our results in Table 5 (row ASTE-Base w/ SCL-Sent.).
We observe that CONTRASTE-Base outperforms ASTE-Base w/ SCL-Sent. with an overall 0.6% improvement in F1 scores. This establishes the better suitability of performing supervised contrastive pre-training on aspect-centric sentiment embeddings, since in ABSA, the sentiments are defined at an aspect level and not at the sentence level.

Table 6 and Table 7 report the main results obtained by our proposed model for 4 ABSA tasks in comparison with baselines for ASTE and AO. From results in Table 7, it is evident that the contributions of our proposed pre-training as well as training framework are salient for structured prediction tasks (triplet and pair). In ASTE, our

proposed methods outperform the strongest baselines (PARAPHRASE) by 0.7, 3.4, 1.6 and 1 F1 points on 14res, 15res, 16res and lap14 respectively. Similar gains are observed in the AO task where our proposed framework sets new state-of-the-art on 15res and lap14. One interesting point to note in Table 7 is how our model compares with Unified ABSA, which is the only existing work addressing a unified framework for ABSA. However, Unified ABSA involves individually training and testing the model on the target tasks, while our approach trains the model for all the tasks - thus making it a multi-task model. It is noteworthy that our way of multi-task modeling (MoT) when paired with the proposed contrastive pre-training significantly outperforms Unified ABSA in ASTE across datasets as well as all but one dataset in AO. AE and OE performance of our model reported in Table 6 is higher than the pair and triplet extraction scores as expected. Scores of AE and OE are quite comparable.

In Table 8 we present the results of AESC, TASD and ACOS tasks. We obtain an improvement of 1, 0.5 and 1.8 F1 points over the best-performing baselines on the 3 tasks respectively. It can be noted that the Mixture-of-Task training improves over the single-task trained CONTRABSA on all the cases for the 3 tasks.

9.2 Qualitative Analysis:

We perform a comprehensive qualitative study to better understand the individual impact of our proposed components.

9.2.1 Embedding Space Study

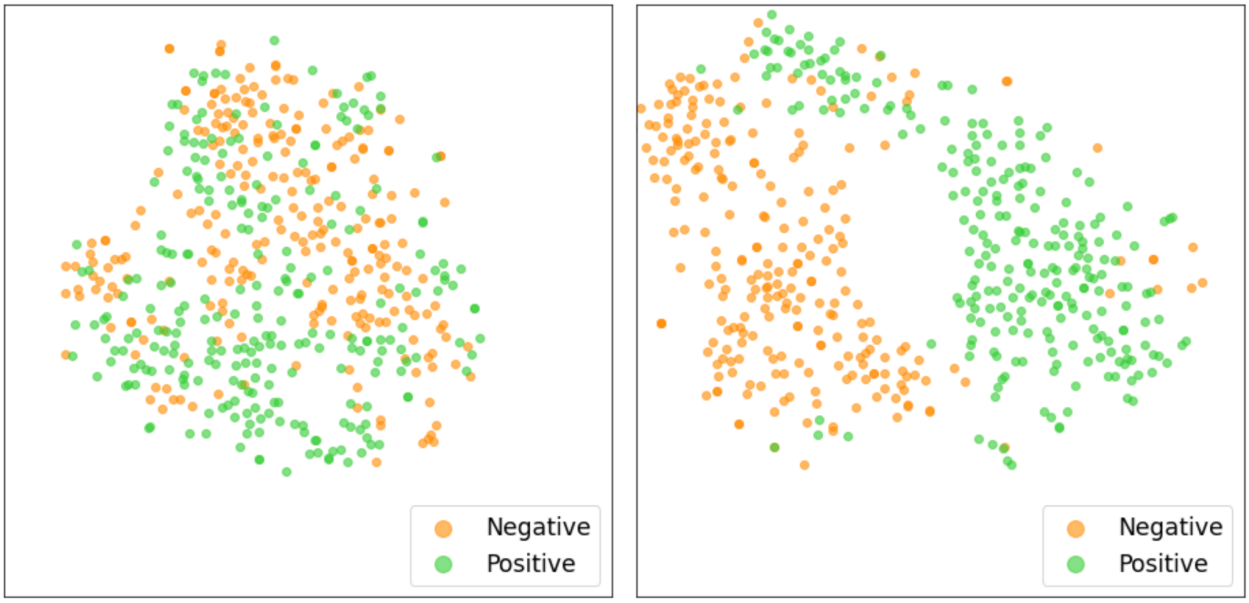
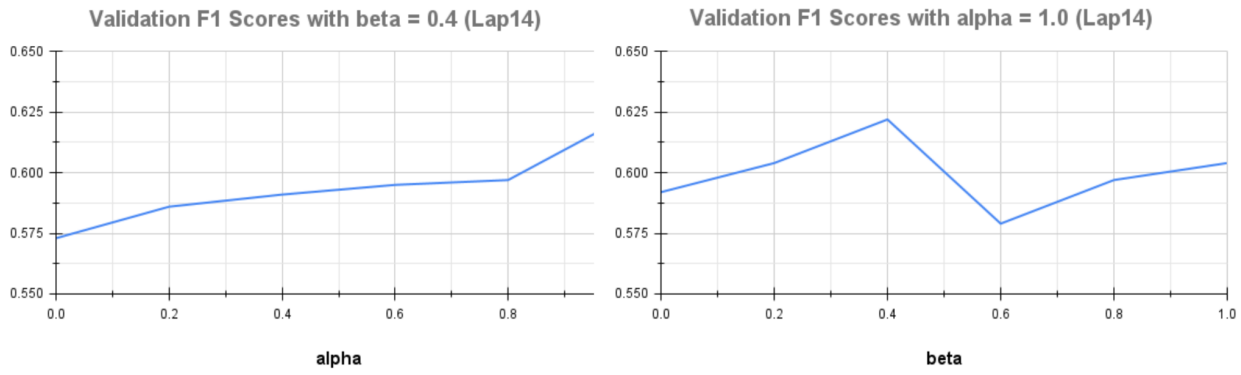


Figure 9. t-SNE visualization of decoder-generated [MASK] token embeddings from aspect-base prompts derived from the 15Res val set. Before pre-training (left), After pre-training (right).

In Figure 9, We observe that the positive, and negative sentiment embeddings are better clustered and more neatly separated from each other after pre-training. Thus, our SCL-based pre-training objective helps in improving the performance on ABSA tasks.

9.2.2 Ablation Study

We plug out components from our proposed framework to study its contribution for the ASTE task. We perform this study on ASTE which is one of the tasks that maximally benefits from the proposed schemes. From Table 9, it can be seen that Contrastive-Pretraining is a significant contributor to the results and removing it results in a 0.24 and 0.35 F1 drop on 15res and 14res respectively. Surprisingly, removing the mixture-of-tasks training procedure hardly affects performance on 14res and causes only a small drop on 15res. As the main motivation behind mixture-of-tasks approach is to train a single model capable of all the ABSA tasks, its individual contribution is reasonable. It can be seen that there is a significant drop if we remove the MTL tasks which are specifically designed to improve ASTE performance.



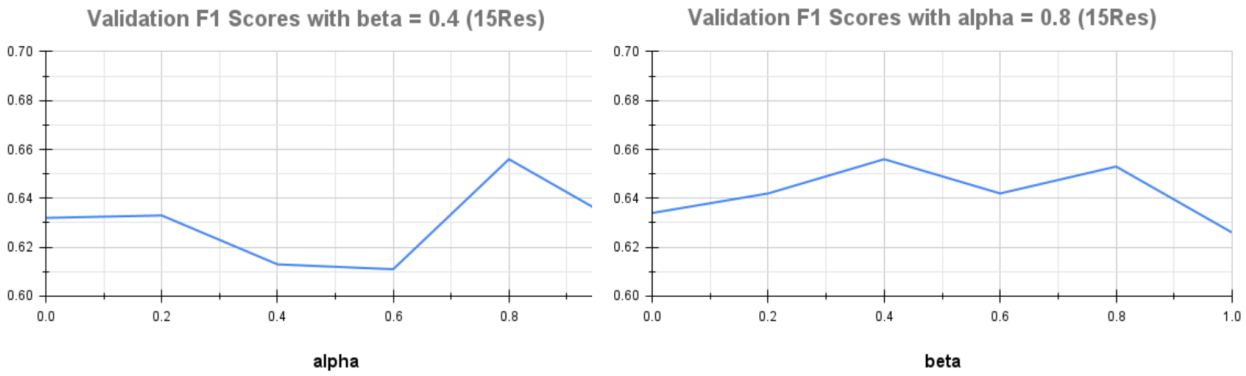


Figure 10: Task weight tuning on the dev set for Opinion Term Detection (OTD) and Triplet Count Estimation(TCE). We first optimize for α (a), and then for (b) β.

9.2.3 Impact of weighting terms in MTL

We perform this task specifically for ASTE. We optimize for values. We start optimizing for α with β set to 0.4 We then optimize β given the optimal α values. One can visibly observe that the ASTE performance varies with changing task weights in Fig. 10. For each dataset, we obtain a different set of optimal α and β values based on the highest ASTE scores on the respective val sets. Each of our models was trained for 20 epochs and the model instance corresponding to the best val F1 score was used to evaluate the test set. We report the median scores over five runs of the experiments. Each pre-training epoch took 10 minutes and each fine-tuning epoch took 1 minute on 15res and 2 minutes on the other three datasets. All our experiments were run on Tesla P100-PCIE 16GB GPU.

10. Conclusion

In this work, we address a prominent gap in the ABSA literature by a) Proposing a unified generative framework that enables training a single model for all ABSA tasks using task-specific prompts and a mixture of tasks training, b) Proposing a novel contrastive pre-training mechanism (CONTRABSA) enabled through aspect prompts to enhance the aspect level sentiment understanding of the model. To the best of our knowledge, this is the first work that proposes a pre-training scheme that benefits structured prediction tasks in ABSA like ASTE, TASD and ACOS. We additionally propose novel templates for Seq2Seq finetuning that outperform existing templates. We pair our final finetuning strategy with 2 auxiliary tasks, namely Opinion Term Detection (OTD) and Triplet Count Estimation (TCE) which when paired with CONTRABSA improves results, particularly for ASTE. Finally, we train a single model for all ABSA tasks using Mixture-of-Tasks (MoT) training from the pre-trained checkpoints, thus enabling a single model to cater to all ABSA tasks. We conduct extensive experimentation to show that our proposed methodology outperforms state-of-the-art approaches on several ABSA tasks over the SemEval 14 datasets and particularly outperforms ASTE baselines significantly. There are several possible extensions of the above work which we look to target. We would attempt to scale up the pre-training data by weak supervision over large-scale unsupervised review data available on the web. Data-efficient strategies could be employed to see the efficacy of pretraining. Finally, we could attempt to extend this work beyond English, into a multilingual model by leveraging recent ABSA datasets in low-resource languages. Our work is under review for publication at a top-tier conference.

Dissemination of Work

"*CONTRASTE: Supervised Contrastive Pre-training With Aspect-based Prompts For Aspect Sentiment Triplet Extraction*", **Nithish Kannen**, Rajdeep Mukherjee, Pawan Goyal. Under Review at **ACL 2023**

Acknowledgement

I would like to express my sincere gratitude to my supervisors, Dr Pawan Goyal and Dr Anirban Mukherjee for their guidance and support, without which this research project would not have been possible. I would also like to thank Mr. Rajdeep Mukherjee, Research Scholar and Bishal Santra, Research Scholar, for his constant help, and support. Furthermore, I would also like to thank the Department of Electrical Engineering, IIT Kharagpur for the facilities provided for the smooth conduction of this project. I would also like to extend my special thanks to the Department of Electrical Engineering for providing me with the opportunity to work on this project.

References

Bai, Xuefeng, Pengbo Liu, and Yue Zhang. 2021. “Investigating Typed Syntactic Dependencies for Targeted Sentiment Classification Using Graph Attention Neural Network.” *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 29: 503–14. https://doi.org/10.1109/TASLP.2020.3042009.

Chen, Chenhua, Zhiyang Teng, Zhongqing Wang, and Yue Zhang. 2022. “Discrete Opinion Tree Induction for Aspect-Based Sentiment Analysis.” In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2051–64. Dublin, Ireland: Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.acl-long.145.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. “BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding.” arXiv. https://doi.org/10.48550/arXiv.1810.04805.

Gao, Tianhao, Jun Fang, Hanyu Liu, Zhiyuan Liu, Chao Liu, Pengzhang Liu, Yongjun Bao, and Weipeng Yan. 2022. “LEGO-ABSA: A Prompt-Based Task Assemblable Unified Generative Framework for Multi-Task Aspect-Based Sentiment Analysis.” In *Proceedings of the 29th International Conference on Computational Linguistics*, 7002–12. Gyeongju, Republic of Korea: International Committee on Computational Linguistics. https://aclanthology.org/2022.coling-1.610.

Gao, Tianyu, Adam Fisch, and Danqi Chen. 2021. “Making Pre-Trained Language Models Better Few-Shot Learners.” *ArXiv:2012.15723 [Cs]*, June. http://arxiv.org/abs/2012.15723.

Huguet Cabot, Pere-Lluís, and Roberto Navigli. 2021. “REBEL: Relation Extraction By End-to-End Language Generation.” In *Findings of the Association for Computational Linguistics: EMNLP 2021*, 2370–81. Punta Cana, Dominican Republic: Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.findings-emnlp.204.

Kamila, Sabyasachi, Walid Magdy, Sourav Dutta, and MingXue Wang. 2022. “AX-MABSA: A Framework for Extremely Weakly Supervised Multi-Label Aspect Based Sentiment Analysis.” arXiv. http://arxiv.org/abs/2211.03837.

Khosla, Prannay, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2021. “Supervised Contrastive Learning.” *ArXiv:2004.11362 [Cs, Stat]*, March. http://arxiv.org/abs/2004.11362.

Lee, Seanie, Dong Bok Lee, and Sung Ju Hwang. 2021. “Contrastive Learning with Adversarial Perturbations for Conditional Text Generation.” *ArXiv:2012.07280 [Cs]*, March. http://arxiv.org/abs/2012.07280.

Li, Chengxi, Feiyu Gao, Jiajun Bu, Lu Xu, Xiang Chen, Yu Gu, Zirui Shao, et al. 2021. “SentiPrompt: Sentiment Knowledge Enhanced Prompt-Tuning for Aspect-Based Sentiment Analysis.” arXiv. https://doi.org/10.48550/arXiv.2109.08306.

Li, Zhengyan, Yicheng Zou, Chong Zhang, Qi Zhang, and Zhongyu Wei. 2021. “Learning Implicit Sentiment in Aspect-Based Sentiment Analysis with Supervised Contrastive Pre-Training.” In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 246–56. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.emnlp-main.22.

Liang, Shuo, Wei Wei, Xian-Ling Mao, Fei Wang, and Zhiyong He. 2022. “BiSyn-GAT+: Bi-Syntax Aware Graph Attention Network for Aspect-Based Sentiment Analysis.” In

Findings of the Association for Computational Linguistics: ACL 2022, 1835–48. Dublin, Ireland: Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.findings-acl.144.

Liu, Pengfei, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. “Pre-Train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing.” *ACM Computing Surveys* 55 (9): 195:1-195:35. https://doi.org/10.1145/3560815.

Loshchilov, Ilya, and Frank Hutter. 2019. “Decoupled Weight Decay Regularization.” arXiv. https://doi.org/10.48550/arXiv.1711.05101.

Lv, Haoran, Junyi Liu, Henan Wang, Yaoming Wang, Jixiang Luo, and Yaxiao Liu. n.d. “Efficient Hybrid Generation Framework for Aspect-Based Sentiment Analysis.”

Mao, Yue, Yi Shen, Jingchao Yang, Xiaoying Zhu, and Longjun Cai. 2022. “Seq2Path: Generating Sentiment Tuples as Paths of a Tree.” In *Findings of the Association for Computational Linguistics: ACL 2022*, 2215–25. Dublin, Ireland: Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.findings-acl.174.

Mrini, Khalil, Shaoliang Nie, Jiatao Gu, Sinong Wang, Maziar Sanjabi, and Hamed Firooz. 2022. “Detection, Disambiguation, Re-Ranking: Autoregressive Entity Linking as a Multi-Task Problem.” In *Findings of the Association for Computational Linguistics: ACL 2022*, 1972–83. Dublin, Ireland: Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.findings-acl.156.

Mukherjee, Rajdeep, Tapas Nayak, Yash Butala, Sourangshu Bhattacharya, and Pawan Goyal. 2021. “PASTE: A Tagging-Free Decoding Framework Using Pointer Networks for Aspect Sentiment Triplet Extraction.” *ArXiv:2110.04794 [Cs]*, October. http://arxiv.org/abs/2110.04794.

Pan, Xiao, Mingxuan Wang, Liwei Wu, and Lei Li. 2021. “Contrastive Learning for Many-to-Many Multilingual Neural Machine Translation.” *ArXiv:2105.09501 [Cs]*, July. http://arxiv.org/abs/2105.09501.

Peng, Haiyun, 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.” *Proceedings of the AAAI Conference on Artificial Intelligence* 34 (05): 8600–8607. https://doi.org/10.1609/aaai.v34i05.6383.

Pontiki, Maria, 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)*, 27–35. Dublin, Ireland: Association for Computational Linguistics. https://doi.org/10.3115/v1/S14-2004.

Raffel, Colin, 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.” *ArXiv:1910.10683 [Cs, Stat]*, July. http://arxiv.org/abs/1910.10683.

Sanh, Victor, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, et al. 2022. “Multitask Prompted Training Enables Zero-Shot Task Generalization.” *ArXiv:2110.08207 [Cs]*, March. http://arxiv.org/abs/2110.08207.

Schick, Timo, and Hinrich Schütze. 2021. “Exploiting Cloze Questions for Few Shot Text Classification and Natural Language Inference.” *ArXiv:2001.07676 [Cs]*, January. http://arxiv.org/abs/2001.07676.

Shin, Taylor, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. “AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts.” *ArXiv:2010.15980 [Cs]*, November. http://arxiv.org/abs/2010.15980.

Su, Yixuan, Lei Shu, Elman Mansimov, Arshit Gupta, Deng Cai, Yi-An Lai, and Yi Zhang. 2022. “Multi-Task Pre-Training for Plug-and-Play Task-Oriented Dialogue System.” arXiv.

https://doi.org/10.48550/arXiv.2109.14739.
Tay, Yi, Anh Tuan Luu, and Siu Cheung Hui. 2017a. “Learning to Attend via Word-Aspect Associative Fusion for Aspect-Based Sentiment Analysis.” *ArXiv:1712.05403 [Cs]*, December. http://arxiv.org/abs/1712.05403.
———. 2017b. “Learning to Attend via Word-Aspect Associative Fusion for Aspect-Based Sentiment Analysis.” *ArXiv:1712.05403 [Cs]*, December. http://arxiv.org/abs/1712.05403.
Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. “Attention Is All You Need.” arXiv. http://arxiv.org/abs/1706.03762.
Wan, Hai, Yufei Yang, Jianfeng Du, Yanan Liu, Kunxun Qi, and Jeff Z. Pan. 2020. “Target-Aspect-Sentiment Joint Detection for Aspect-Based Sentiment Analysis.” *Proceedings of the AAAI Conference on Artificial Intelligence* 34 (05): 9122–29. https://doi.org/10.1609/aaai.v34i05.6447.
Wu, Zhen, Chengcan Ying, Fei Zhao, Zhifang Fan, Xinyu Dai, and Rui Xia. 2020. “Grid Tagging Scheme for Aspect-Oriented Fine-Grained Opinion Extraction.” In *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2576–85. Online: Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.findings-emnlp.234.
Xu, Lu, Hao Li, Wei Lu, and Lidong Bing. 2020. “Position-Aware Tagging for Aspect Sentiment Triplet Extraction.” In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2339–49. Online: Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.emnlp-main.183.
Xu, Shusheng, Xingxing Zhang, Yi Wu, and Furu Wei. 2022. “Sequence Level Contrastive Learning for Text Summarization.” *ArXiv:2109.03481 [Cs]*, January. http://arxiv.org/abs/2109.03481.
Xue, Wei, and Tao Li. 2018. “Aspect Based Sentiment Analysis with Gated Convolutional Networks.” In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2514–23. Melbourne, Australia: Association for Computational Linguistics. https://doi.org/10.18653/v1/P18-1234.
Yan, Hang, Junqi Dai, Tuo Ji, Xipeng Qiu, and Zheng Zhang. 2021. “A Unified Generative Framework for Aspect-Based Sentiment Analysis.” In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2416–29. Online: Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.acl-long.188.
Yang, Heng, Biqing Zeng, Mayi Xu, and Tianxing Wang. 2021. “Back to Reality: Leveraging Pattern-Driven Modeling to Enable Affordable Sentiment Dependency Learning.” *ArXiv:2110.08604 [Cs]*, October. http://arxiv.org/abs/2110.08604.
Yang, Shuoheng, Yuxin Wang, and Xiaowen Chu. 2020. “A Survey of Deep Learning Techniques for Neural Machine Translation.” arXiv. https://doi.org/10.48550/arXiv.2002.07526.
Zeng, Changchang, Shaobo Li, Qin Li, Jie Hu, and Jianjun Hu. 2020. “A Survey on Machine Reading Comprehension: Tasks, Evaluation Metrics and Benchmark Datasets.” arXiv. https://doi.org/10.48550/arXiv.2006.11880.
Zhang, Wenxuan, Yang Deng, Xin Li, Yifei Yuan, Lidong Bing, and Wai Lam. 2021. “Aspect Sentiment Quad Prediction as Paraphrase Generation.” *ArXiv:2110.00796 [Cs]*, October. http://arxiv.org/abs/2110.00796.
Zhang, Wenxuan, Xin Li, Yang Deng, Lidong Bing, and Wai Lam. 2021. “Towards Generative Aspect-Based Sentiment Analysis.” In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, 504–10. Online: Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.acl-short.64.

———. 2022. “A Survey on Aspect-Based Sentiment Analysis: Tasks, Methods, and Challenges.” arXiv. http://arxiv.org/abs/2203.01054.

Zhong, Wanjun, Yifan Gao, Ning Ding, Yujia Qin, Zhiyuan Liu, Ming Zhou, Jiahai Wang, Jian Yin, and Nan Duan. 2022. “ProQA: Structural Prompt-Based Pre-Training for Unified Question Answering.” In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 4230–43. Seattle, United States: Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.naacl-main.313.

Appendix

1. Applications of Prompting

1.1. Benchmarking Dialog Evaluation

1.1.1 Introduction

In this section, we look at a particular application of prompt learning in the context of dialogue evaluation. The recent surge in the use of Dialogue agents like Alexa and ChatGPT makes this domain of research even more profound. While a popular way of evaluating dialogue involved human-in-loop through manual annotation, it is very expensive and not very scalable for large-scale Dialog Evaluation. Dialog evaluation is the task of assessing the quality of responses generated by dialog models in terms of several properties. However, one significant impediment to open-domain dialog generation research is the lack of meaningful automatic metrics for open-domain dialog evaluation. Standard language generation metrics have been shown to be ineffective for dialog evaluation, a large part of which is because conversations can be followed by multiple valid responses. Standard automatic metrics (e.g. BLEU), which use references for evaluation, cannot deal with this quality, known as the one-to-many response problem. Many recently introduced automatic metrics for dialog evaluation have attained increasingly stronger correlations with human judgment. Since human dialogue evaluation typically measures multiple fine-grained properties (e.g. appropriate, interesting, consistent), automatic evaluation metrics should be expected to

do the same. As an attempt to come up with a strong metric that correlates highly with human judgement, we attempt to leverage Instruction Fine-Tuned language models (Chung et al. 2022) for zero-shot dialog evaluation. More specifically, we attempt to benchmark the efficacy of different language models on Dialog Evaluation in a zero-shot fashion. For the same, we carefully design prompts that are passed as inputs to the language model to subsequently obtain the model scores. In the next sections, we discuss this process in more detail.

The major contributions of this chapter can be summarized as follows:

- Design prompts for dialog evaluation through Instruction-Fine tuned language models. Assess the sensitivity of the model outputs to prompts by comparing the performance of different prompts.
- Benchmark the performance of different off-the-shelf Instruction Fine-tuned language models available in the market in a zero-shot fashion.
- Alleviate an initial hurdle predominately seen in smaller models which is the outputting of out-of-vocabulary words as model responses. Such cases will have to be discarded otherwise because the verbalizer does not map these words to numeric values for the final correlation calculation with human judgement.
- We release our codes for public use at: https://github.com/nitkannen/DialogEval

1.1.2 Literature Review

Automatic Dialog evaluation is a highly researched field in the NLP community with several researchers attempting to contribute.The widely used automatic metrics, such as BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005) and ROUGE (Lin, 2004), use statistical rules to measure the degree of lexical word overlap between

generated responses and reference responses. However, these metrics have been demonstrated to correlate poorly with human judgments due to the absence of semantic information (Liu et al., 2016; Novikova et al., 2017). Therefore, the subsequent metrics are considered to incorporate the semantic information. For instance, BERTScore (Zhang et al., 2020) turns to measure the soft semantic word-overlap rather than the hard lexical word-overlap like BLEU. Moreover, learnable metrics encoding the semantic information have been attracting interest recently, which are trained in a supervised manner with large-scale human-annotated data, such as ADEM (Lowe et al., 2017), or trained in an unsupervised manner with automatically constructed data, such as RUBER (Tao et al., 2018) and BERT-RUBER (Ghazarian et al., 2019). Furthermore, the recently proposed coherence metric, GRADE (Huang et al., 2020), introduces the graph information of dialogue topic transitions and achieves the current state-of-the-art results. Note that these learnable metrics are trained in a two-level training objective to separate the coherent dialogues from the incoherent ones, while our QuantiDCE models the task in a multi-level setting which is closer to the actual human rating.

1.1.3 Tasks and Datasets

We use the Amazon Topical Chat dataset (Gopalakrishnan et al. 2019) that was re-released after manual annotation by (Mehri and Eskenazi 2020). The manual annotation involved 3 annotators that provided a score for a particular quality of the model response. The different qualities involved in the dataset are defined below:

- *Understandable* (0 - 1): Is the response understandable given the previous context?
- *Natural* (1 - 3): Does the response seem to be something natural?
- *Maintains Context* (1 - 3): Does the response serve as a valid continuation of the preceding conversation?
- *Interesting* (1 - 3): Is the response dull or interesting?

• *Uses Knowledge* (0 - 1): Given the fact that the response is conditioned, how well does the response use that fact?
• *Overall Quality* (1 - 5): Given your answers above, what is your overall impression of the quality of this utterance

The Topical-Chat dataset is a large collection of human-human knowledge-grounded open-domain conversations that consisting of 11,319 dialogues and 248,014 utterances. We use the dataset in its entirety to obtain the correlation numbers with the annotators.

1.1.4. Methodology

1.1.4.1 Prompting

Prompting (Schick and Schütze 2021) is a recent technique that has gained popularity. Prior to this, the task was treated as a black box that a model has to learn using the given data samples. Prompting is a simple technique where the task or the target labels are explained to the model in natural language terms (Sanh et al. 2022). More specifically, consider a simple sentiment classification task trained using an encoder. The model is trained to predict 0 (negative) or 1 (positive) by minimizing the cross-entropy loss function. The model (here, an LLM) is not aware of the task it is being trained to solve. On the other hand, in the prompting paradigm, you prompt the model with an appropriate question that it learns to answer in the Natural Language (NL) form. For example, the question “Is the sentiment positive or negative?”, “Is it good or bad?” and, “Was the sentiment good?”, are some of the plausible prompts that can be used for the task of sentiment classification. The answer is given by the model in Natural language (NL) form depending on the question prompt. More details on prompt-based learning can be found in (Liu et al. 2023).

1.1.4.2 Our Approach

Since we task the model to output evaluation scores, we verbalize each of the numerical values corresponding to the different qualities. More specifically, for the quality understandable that takes 2 possible values (0 and 1).) can be mapped to the word “no” and 1 can be mapped to the word “yes”. As shown in Fig. 1 the input prompt we use consists of the context, the fact involved and the dialog history all concatenated. Finally, the prompt ends with a carefully designed question that instructs the model to output a word out of the provided options. Note that the phrasing of the different questions depends on the quality metric (understandable/maintains context etc.,) that is being evaluated.

Prompt example:

Context:
Agent: i don't think i have heard of them . i have heard that one of the directors recently passed away , and his last film was nominated for an academy award
User: yeah , sadly , disney (which owns the american rights to the films) does n't tend to promote them very much . i think they're worried they'll cut into their " home grown " market . anyway , dunno if you even like animated movies , but they're worth checking out .
Agent: i don't watch them very often . apparently there was a showing of the recent film in a park in d.c . that's one u.s . city i haven't been to
User: sadly , i haven't been to dc either , although i 've always wanted to visit there . apparently there's a lot of interesting going down this summer . they 're having a crab feast at the navy - marine corps stadium . they 'll have 100 gallons of crab soup ! can you imagine that much soup ?
Generated response:
Agent: I really like playing tennis.
Question about the generated response:
1. Understandable (Yes/Somewhat/No): Is the response understandable given the previous context?
Answers:

Figure 1. An example of a prompt that is provided as input to the Instruction Finetuned model. The model is expected to respond with one of the available options.

1.1.5. Experiments

Table 1. Results obtained using zero-shot FlanT5-small.
Prompt used: "Understandable (no/yes): Is the response understandable given the previous context?"

Correlation	Understandable	Natural	Maintains Context	Engaging	Uses Knowledge	Overall
Annot1-Model	0.04	nan	nan	nan	0.66	nan
Annot2-Model	-0.08	nan	nan	nan	0.408	nan
Annnot3-Model	-0.034	nan	nan	nan	0.66	nan
Mean-Model	0.05	nan	nan	nan	0.64	nan

Table 2. Results obtained using zero-shot FlanT5-base.
Prompt used: "Understandable (no/yes): Is the response understandable given the previous context?"

Correlation	Understandable	Natural	Maintains Context	Engaging	Uses Knowledge	Overall
Annot1-Model	0.17	0.14	nan	0.11	nan	nan
Annot2-Model	0.12	0.188	nan	0.08	nan	nan
Annnot3-Model	0.144	0.17	nan	0.14	nan	nan
Mean-Model	0.133	0.167	nan	0.11	nan	nan

It is interesting to note that many of the above quality evaluations could not be processed because the model produces i) invalid responses that could not be converted into a numerical value for correlation computation, ii) produces the same response

which results in a *nan* correlation value as seen in Table 1. We propose 2 methods as an initial attempt to alleviate the above issue.

1. **Constrained Decoding**

It is a technique used in Language models to guide the generation of text by an LLM towards a specific set of constraints or requirements. In our case, we constrain the model to only predict tokens that are in the available options for the particular quality. Concretely, we hope to obtain the probability P of the language model in predicting a token t_i in the model vocabulary. We compare the probabilities of the model producing each of the available options in the prompt vocabulary and pick the response with the highest probability.

2. **Prompt Modification**

Another possible attempt to alleviate the OOV case is trying out different prompts. More specifically, we observed that not only the model scores are highly sensitive to the input prompt, but the model's tendencies to stick to its instruction also largely depended on the prompt we used. For the time being, we manually handcraft these prompts to pick the most suitable prompt for our use case. In the future, we wish to come up with efficient techniques that enable this process of prompt selection.

Table 3. Results obtained using zero-shot FlanT5-base.
Prompt used: "Understandable: Is the response understandable given the previous context? pick one out of (no/yes)"

Correlation	Understandable	Natural	Maintains Context	Engaging	Uses Knowledge	Overall
Annot1-Model	0.34	0.13	0.21	0.18	0.22	0.04
Annot2-Model	0.27	0.21	0.07	0.31	0.24	0.011

Annot3-Model	0.28	0.14	0.14	0.23	0.16	0.08
Mean-Model	0.3	0.16	0.14	0.26	0.19	0.5

Table 4. Results obtained using zero-shot T0pp.
Prompt used: Prompt used: "Understandable: Is the response understandable given the previous context? pick one out of (no/yes)"

Correlation	Understandable	Natural	Maintains Context	Engaging	Uses Knowledge	Overall
Annot1-Model	0.14	0.08	0.2	0.16	0.05	0.01
Annot2-Model	0.23	0.01	0.12	0.18	0.09	-0.04
Annot3-Model	0.12	0.02	0.1	0.06	0.11	0.04
Mean-Model	0.16	0.04	0.14	0.12	0.08	0

Table 5. Results obtained using zero-shot TK-Instruct.
Prompt used: Prompt used: "Understandable: Is the response understandable given the previous context? pick one out of (no/yes)"

Correlation	Understandable	Natural	Maintains Context	Engaging	Uses Knowledge	Overall
Annot1-Model	0.41	0.22	0.27	0.32	0.24	0.12
Annot2-Model	0.44	0.22	0.13	0.3	0.22	0.19
Annot3-Model	0.39	0.19	0.21	0.34	0.29	0.08
Mean-Model	0.42	0.21	0.22	0.32	0.25	0.14

Table 6. Results obtained using zero-shot InstructGPT.
Prompt used: Prompt used: "Understandable: Is the response understandable given the previous context? pick one out of (no/yes)"

Correlation	Understandable	Natural	Maintains Context	Engaging	Uses Knowledge	Overall

Annot1-Model	0.34	0.28	0.18	0.2	0.14	0.08
Annot2-Model	0.49	0.21	0.17	0.16	0.08	0.01
Annot3-Model	0.42	0.14	0.16	0.22	0.09	0.08
Mean-Model	0.42	0.19	0.17	0.18	0.11	0.05