# Studying Limitations of Generative Transformer based models for Aspect Based Sentiment Analysis

by

Dhruv Mullick

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computing Science
University of Alberta

# Abstract

Companies can only progress if they understand what their customers feel about their products and services. With companies having an online presence, and with the availability of third-party online reviewing platforms like Yelp, it becomes critical to scour through online reviews. Analysing millions of online reviews across various platforms is not a trivial task. Aspect-based Sentiment Analysis (ABSA) is an NLP task useful for automating such an analysis. ABSA solutions have historically used discriminative models, but there have been recent advances in the field which use generative transformer models (like T5 and BART). Generative ABSA models treat the ABSA task as a text generation problem. We study the latest generative ABSA models and discuss some of their limitations. We find that state-of-the-art generative ABSA models perform well for the standard ABSA settings. However, they face problems in certain real-life scenarios like handling cross-lingual settings and with reviews containing coreference resolution. We propose solutions for these limitations, justifying why they work.

# Preface

This thesis is based upon work that was submitted to The 17th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2023), where the work awaits review with coauthors Dr. Bilal Ghanem and Dr. Alona Fyshe.

# Acknowledgements

It was my intention to do quality research before pursuing a lifetime of industry employment. I believe I have fulfilled my desire thanks to the support I have received from the people in my life.

I would like to thank my supervisor, Dr. Alona Fyshe for accepting me as her student and providing me with constant advice throughout my time at the University of Alberta. There were many times when I felt disheartened about my research, but her guidance helped me pull through.

Next, a thank-you to my lab-mates and colleagues from the University. They helped me refine both my research and presentation skills by providing critical feedback.

Finally, I am grateful to my friends and family, who have always believed in me, motivating and comforting me throughout my life.

# Contents

# List of Tables

# List of Figures

# Abbreviations

**ABSA** Aspect Based Sentiment Analysis.

**ALSC** Aspect Level Sentiment Classification.

**CR** Coreference Resolution.

**DPR** Definite Pronoun Resolution.

**Int.** Intermediate.

**LLM** Large Language Model.

**LM** Language Model.

**MAMS** Multi-Aspect Multi-Sentiment Dataset.

**QA** Question Answering.

**QQP** Quora Question Prediction.

**Rest16** SemEval Restaurant 2016 Dataset.

**SQuAD** Stanford Question Answering Dataset.

**SST** Stanford Sentiment Treebank.

**T5** Text-To-Text Transfer Transformer.

# Chapter 1

# Introduction

Understanding customers' opinions towards products is one of the main priorities for companies to improve on the products' acceptance by people. Online reviews make it easy for customers to share their feelings about products and services in a quick and efficient way. But for business owners, this can mean a deluge of comments with a variety of concerns. Companies with millions of customers receive massive amounts of online reviews that can not be analyzed manually, thus needing automation. This automation can be done using Aspect-based Sentiment Analysis (ABSA) which helps to explain customers' opinions towards products and services.

For example, a laptop manufacturer looking to analyze customers' opinions towards a newly released laptop can use ABSA on reviews existing on retail platforms like Amazon and Walmart. With ABSA, the laptop manufacturer can know the customer's sentiment towards the laptop's various features like its "display" and "battery". Formally, the "display" and "battery" are called the aspect terms (or aspects) for which sentiment is detected using ABSA.

ABSA typically involves detecting the different aspects that exist in the review, and classifying their sentiment polarities. However, ABSA also has a subtask called Aspect Level Sentiment Classification (ALSC) [1] to analyse a model's sentiment classification ability without evaluating its aspect extraction ability. ALSC allows detecting the sentiment associated with a specific aspect in the review. So in the earlier example, the

laptop manufacturer could use ALSC to analyze reviews specifically for the sentiment of "display" aspect of the laptop.

With the advent of Transformer based models [2], ABSA solutions have achieved better performance. ABSA solutions were earlier based on discriminative models, but recently a few solutions based on generative models have been proposed. Discriminative ABSA solutions commonly use a two-step process: 1) extraction - detect aspects using methods like sequence labeling; 2) classification - classify sentiments for the detected aspects. On the other hand, generative ABSA solutions generate aspects and sentiment polarities together without separate extraction and classification steps. Generative ABSA solutions are simpler and said to perform better than their discriminative ABSA counterparts [1, 3]. However, these generative ABSA solutions were not evaluated comprehensively in prior work, including in some specialized settings.

In our work, we study generative ABSA solutions in the following specialized settings:

- ABSA that requires Coreference Resolution (CR) - the ability to identify words referring to the same entity (Chapter 3). For example, CR is needed to predict the sentiment for "food" in the sentence - "I had food, it was bad." Here, "food" has a negative sentiment only because it is referring to the same entity as "it" which in turn is associated with a negative sentiment.

- Cross-domain (different domain used at train and test time), cross-lingual (different language used at train and test time), and cross-domain/lingual settings (both domain and language different at train and test time) (Chapter 4).

In this thesis, we pose a two-part **research question**:

1. Are there settings where generative ABSA transformer models have poor performance?

2. (If yes,) Are there any techniques to mitigate this poor performance?

## 1.1 Objectives

The contributions of our work are as follows:

- We find several specialized settings where generative ABSA solutions have poor performance.
- For the setting where accurately predicting the sentiment associated with an aspect requires CR:
    1. We demonstrate and quantify the poor performance of generative ABSA (or more specifically - ALSC) solutions when working with reviews requiring CR ability.
    2. We show that intermediate training with either intermediate task: Quora Question Pairs [4] (a task to detect semantically equivalent questions) or Commongen [5] (a generative commonsense task) improves performance on reviews requiring CR ability.
    3. We release a dataset[1] to benchmark future ABSA methods on reviews requiring CR ability.
    4. We present a framework for evaluating and improving a transformer model's performance on CR cases, which can be used for NLP target tasks other than ABSA as well.
- For the setting where train and test datasets have a different domain and/or language:
    1. We show that discriminative ABSA solutions work as well or better than generative solutions in many cases.

[1] Accessible via the Dataverse link: (https://doi.org/10.5683/SP3/HSKJEY)

2. We demonstrate that generative ABSA solutions make significant errors on cross-lingual and cross-domain/lingual settings wherein they incorrectly generate aspects not existing in the input sentence.
3. We propose a mitigation technique - a simple constrained decoding mechanism, called masking, which improves performance significantly (by up to 20%) in many cases.

Our work paints a picture of generative ABSA performance and provides a few targeted ways to improve the performance.

## 1.2 Outline

In Chapter 2 of this thesis, we explain the fundamental concepts associated with our work. We explain the ABSA task and its ALSC subtask, which we consider in our research problem. Language modeling and transformer based language models are discussed in detail. These transformer models form the backbone of modern ABSA solutions, including that of the generative and discriminative ABSA solutions which we consider in our research. Lastly, we describe the significance testing we perform to ensure the conclusions of our research are not based on chance results.

Chapter 3 firstly motivates the problem faced by generative ABSA (or more specifically ALSC) solutions in handling reviews requiring CR ability. We describe the coreference resolution (CR) concept, and the existing standard ABSA datasets available to us. We explain how to use standard ABSA datasets to build a dataset suitable for benchmarking performance on CR requiring reviews. We discuss our choice of intermediate tasks used for the intermediate training in our experiments. We then describe three experiments we performed, along with their results. In our first experiment, we quantify the problem faced by generative ABSA solutions in handling reviews requiring CR ability. In the second experiment, we explore intermediate training using various intermediate tasks, to mitigate the drop in performance while

handling CR requiring reviews. In the last experiment, we show the intermediate-task trained model's performance on a CR dataset to justify its improved performance on CR requiring reviews. Finally, we provide the related work done for ABSA in the context of CR, and discuss the conclusions of our study.

In Chapter 4, we begin by giving a background of the evaluation done on generative ABSA solutions in prior work. We motivate the need to explore generative ABSA solutions in non-standard settings like cross-domain and cross-lingual. We then explain the datasets and the models used in our experiments. One of the considered models relies on a constrained decoding mechanism called logit masking. This constrained decoding based model is meant to mitigate some of the problems faced by generative models in non-standard settings. The experiments and results for model evaluations in different settings are then presented. We do an error analysis and discuss the reasons behind the under-performance of certain models in different conditions. Finally, we present the conclusions of our study.

Chapters 3 and 4 are based on independent short papers submitted to EACL 2023 (European Chapter of the Association for Computational Linguistics).

Finally, in Chapter 5, we present the conclusions of our thesis on the limitations of generative ABSA solutions and discuss the directions for future work.

# Chapter 2

# Background

In this chapter, we give a background of the research done as part of this thesis. We first introduce the Aspect Based Sentiment Analysis (ABSA) problem and explain the language modeling task. We then elaborate on Transformer based models which are the present state-of-the-art models for language modeling, explaining the fine-tuning approach used to impart knowledge into Transformers.

## 2.1 Aspect Based Sentiment Analysis

Sentiment analysis is about predicting a sentiment for a given text [6]. This task can be lacking at times since one sentence can possess multiple sentiments. For example, there is no single sentiment for the sentence - "The service was good at the restaurant, but the food was not". Thus, there are some fine-grained variations of sentiment analysis like Aspect Based Sentiment Analysis (ABSA) [7].

ABSA allows for the prediction of aspects and their associated sentiment polarities in a given sentence. An example of ABSA would be - "The service was good at the restaurant, but the food was not" which has two aspect terms ("service" and "food"), associated with sentiments "positive" and "negative", respectively.

ABSA typically involves two sub-problems: 1) extracting aspects; 2) classifying the aspects' sentiments. Instead of focusing on both these sub-problems, one can choose to analyze (or use) only the sentiment classification ability as well. This sentiment

classification ability can be analysed using the Aspect Level Sentiment Classification (ALSC) task. ALSC predicts the sentiment of a specific aspect term in the review. For instance, using ALSC, a restaurant owner can analyze reviews for the sentiment associated with "food." In the earlier example review - "The service was good at the restaurant, but the food was not", ALSC would detect the sentiment for "food" as "negative". ALSC is a sub-type of the ABSA problem [1], such that it only predicts the sentiment for the given aspect, without the need for extracting aspect terms.

State-of-the-art ABSA (and ALSC) solutions use large language models like transformer models in various settings [1, 3, 8–10]. Such models are further explained in Section 2.3).

## 2.2 Language Modeling

Language Modeling is a critical problem in NLP, involving generating the probability distribution of the next word given a context. The probability of the next word $w_i$ is given by Equation 2.1 where $w_{i-1}, w_{i-2}, ..., w_0$ are the preceding words, i.e. the context of $w_i$.

$$P(w_i|w_{i-1}, w_{i-2}, ..., w_0) \tag{2.1}$$

For example, a language model (LM) might have an input "I couldn't sleep because the neighbour's dog was ____". An LM would learn a probability distribution for the word which would fit in the blank. This could realistically have the highest probability for "barking". Figure 2.1 shows how the LM would model a probability distribution for the next word.

With sufficient training, LMs can be used for various NLP tasks like ABSA, question answering and natural language inference [3, 11]. An ALSC task (sub-type of ABSA) can be posed to the LM as a language modeling problem, in the form of the input: "Get sentiment: [sentence]. [aspect]". The LM can be trained to predict the

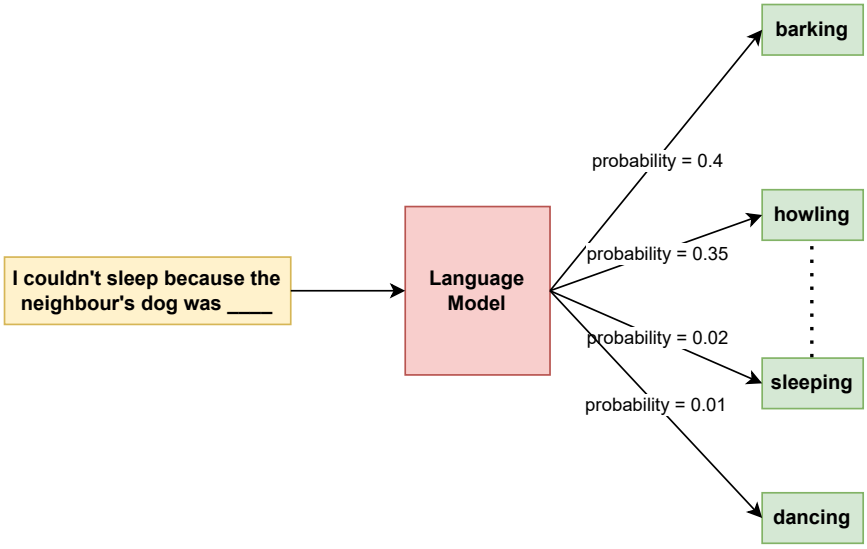



Figure 2.1: Example of Language Modeling. A probability distribution is generated over the words in the vocabulary

sentiment of the aspect as the next word - “positive”, “negative” or “neutral”. For example, the LM would be trained to predict the next word as “positive” when it encounters the input: “Get sentiment: The service was good, but the food was not. service”. In a similar way, a language model can be used to perform various NLP tasks.

Language modeling can be done using various approaches such as Recurrent Neural Networks and Transformer models. Transformer based models are the current state-of-the-art for language modeling [2, 11].

## 2.3 Transformers

Large language models known as Transformers [2] were released in 2017, and have ever since been part of state-of-the-art solutions in artificial intelligence fields like natural language processing and computer vision [12]. Transformers are forms of deep neural network models and effectively model long-range dependencies [2, 12]. The original transformer model proposed by Vaswani *et al.* [2] is an encoder-decoder architecture where the encoder generates a representation (encoding) for the input

sentence, whereas the decoder generates the next token in the sequence by using the encoding and the already generated sequence.

This original transformer model is also known as the vanilla transformer model. Various architectural variations of this vanilla model have since been proposed [12]:

- Encoder only: These are typically used for generating a representation of the input sentence and then doing classification or sequence labeling tasks. Popular encoder only models include BERT [11] and RoBERTa [13].

- Decoder only: Such models are usually used for auto-regressive sequence generation tasks like language modeling. GPT-3 [14] is one such decoder only model which has shown exemplary results on NLP tasks.

- Both encoder and decoder: These models are popularly used for sequence-to-sequence generation tasks like neural machine translation where the generated tokens depend on both the original input as well as the already generated tokens. T5 [15] and BART [16] are some popular encoder-decoder transformers.

Transformer models are pre-trained with very large datasets (of the order of billion examples) on objectives such as language modeling [11, 15].

### 2.3.1 Transformer Model Types

There are discriminative and generative paradigms for transformer based models [17].

#### Discriminative Transformer Models

Discriminative transformer models are pre-trained with discriminative tasks such as Next Sentence Prediction (NSP) and Masked Language Modeling [11]. Some popular discriminative models are BERT [11] and RoBERTa [13].

BERT is an encoder-only transformer architecture which provides an encoding (representation) of the input text. The encoding is then used as input to a classifier to do predictions at the sentence or at the token level. The classifier can be as simple

as a single linear layer with softmax head [11]. RoBERTa is a variant of BERT which is pre-trained without the NSP pre-training objective of BERT. Moreover, RoBERTa is trained with larger datasets and obtains a performance better than BERT on various NLP tasks.

**Generative Transformer Models**

Generative transformer models are pre-trained with generative tasks like language modeling. Some popular generative models are GPT, T5 and BART. We use T5 and BART in this thesis.

- **T5** [15] (Text-To-Text Transfer Transformer) is a sequence-to-sequence pre-trained transformer based model. It has an architecture similar to the original vanilla transformer model [2]. T5 has been pre-trained on several pre-training objectives, on a massive dataset with hundreds of gigabytes of data. The authors of T5 have conducted extensive experiments on various training datasets and strategies and have found a model which gives state-of-the-art performance on several NLP benchmarks. In our experiments explained later, we use a Huggingface implementation of T5[1].

- **BART** [16] is also a modern sequence-to-sequence pre-trained transformer based model. It is trained as a de-noising autoencoder - learning to reconstruct the original text which was corrupted by adding noise. BART's architecture comprises several encoder and decoder layers [1]. BART's encoder layer is said to be a generalization of the BERT architecture which solely consists of an encoder component. On the other hand, the decoder layer being an auto-regressive decoder, is seen as a generalization of the GPT architecture, which only consists of a decoder component. BART can be used in several downstream tasks like a) classification tasks by using the decoder representation of final token;

[1]https://huggingface.co/t5-large

b) sequence generation tasks like machine translation by using tokens output from the auto-regressive decoder. In our experiments explained later, we use a Huggingface implementation of BART[2].

### 2.3.2 Transformer Fine-Tuning

For using transformers in various tasks, instead of training from scratch every time, a popular approach is to use pre-trained transformers and then fine-tune on the appropriate downstream dataset. This approach leads to state-of-the-art performance on several tasks [11].

Devlin *et al.* [11] show how BERT can be used for NLP tasks like the GLUE benchmark comprising of nine sub-tasks. Devlin *et al.* [11] demonstrate how a pre-trained BERT model can be used along with a simple classification layer. The model is fine tuned end-to-end on the required GLUE sub-task. Fine tuning updates BERT's internal weights as well as weights of the newly added classification layer. This simple yet effective fine tuning methodology gives state-of-the-art results on the GLUE sub-task. In a similar way, other transformer based models can be fine tuned for various NLP tasks. By fine-tuning on a downstream task, we impart the model an ability to perform that task.

It is also possible for a model to have an ability to perform multiple tasks at once. As seen in Figure 2.2 from Raffel *et al.* [15], the T5 model has been trained to perform on a mixture of NLP tasks simultaneously. A model can be trained to perform several tasks at once by using input prefixes, called "input prompts" [15, 18]. For instance, for a translation task, we can have an input prompt: "translate English to German". This prompt can let the model know that we want it to translate the subsequent English input text to German language. Input prompts can be used to extract task specific knowledge from the pre-trained language model [19].

Apart from simply fine-tuning a model on a target task, it can be useful to do

[2]https://huggingface.co/facebook/bart-base

*intermediate training*, which can improve the model's performance and stability on the target task [20, 21]. Intermediate training involves fine-tuning on an intermediate task prior to fine-tuning on the target task. This can be visually seen in Figure 2.3.

In Chapter 3, while using an intermediate training approach, we use different prompts for the intermediate and target tasks. This allows the model to differentiate between the intermediate task objective and the target task objective. For example, when using intermediate training with the CosmosQA task for the final ALSC target task, we would first train the model using the CosmosQA dataset with an input prompt for CosmosQA, and then train the model on the ALSC dataset using an input prompt for ALSC. Note that we do not use such input prompts in the experiments in Chapter 4 because only a single task is involved.

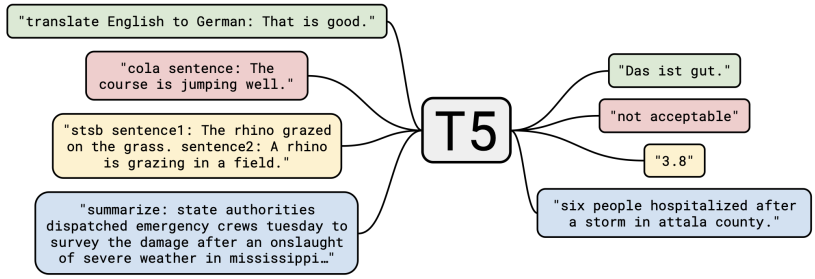



Figure 2.2: T5 is trained to perform well on various NLP tasks simultaneously. Figure from Raffel *et al.* [15]

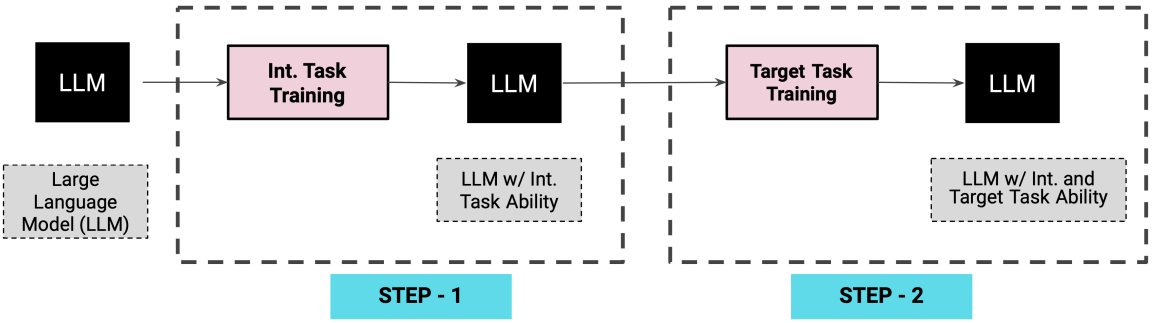



Figure 2.3: The intermediate training process applied to a large language model (LLM). Step 1 involves fine-tuning the LLM on an intermediate (Int.) task, followed by Step 2 which involves fine-tuning on the target task.

### 2.3.3 Transformers in Aspect Based Sentiment Analysis

Several state-of-the-art ABSA (and ALSC) solutions use transformer models as part of their solutions [1, 3, 8–10, 22]. These solutions can be divided into discriminative and generative as per their methodologies.

#### Discriminative Transformer Models for ABSA

Discriminative models have been well studied, with several variations being released for the same [8, 9, 22]. Discriminative models, which use decision boundaries to make predictions, commonly use sequence labeling techniques to detect aspects in a given review (extraction) and then use another step to classify those aspects (classification). A few discriminative models also perform extraction and classification at once [8, 9]. For example, for the sentence "The service was good, but the food was not", the discriminative model in the first step would extract the aspects "service" and "food". In the second step, the model would classify the polarities of "service" and "food" in the given context (sentence) as being positive and negative.

One prominent discriminative approach is SPAN-BERT [8]. Other ABSA approaches treated ABSA as a sequence labeling task, marking every word as a beginning word, ending word, or being inside a target phrase [9, 23, 24]. SPAN-BERT on the hand proposes a span-based extract-then-classify framework. Thus reducing the search space and possibility of sentiment inconsistency. It obtains one of the highest accuracy among discriminative models.

#### Generative Transformer Models for ABSA

Compared to discriminative models, generative models are new, with few studies having been done on them [1, 3]. Generative ABSA models learn probability distributions of the next word, and generate aspects and sentiment polarities together without separate extraction and classification steps. The literature shows that these models perform better than discriminative models, at least in the mono-lingual in-

domain English setting. A possible reason for the performance gain is that generative models have the advantage that they understand the semantics of the label, unlike discriminative models [3]. For example, for the sentence "The service was good, but the food was not", the generative model would autoregressively output "service positive <sep> food negative." Here, "<sep>" is used to demarcate a separation between multiple aspect-polarity pairs.

Prominent prior work for generative ABSA models, rely on T5 and BART models [1, 3]. The transformer models are fed reviews as input, and the corresponding aspect-polarity pairs are output from the model autoregressively. This surprisingly simple generative technique yields state-of-the-art results.

## 2.4 Statistical Significance

Statistical tests are used to test the significance of experimental results [25]. These are important to see if the results we observe can be attributed to randomness. Our experiments involve comparing two algorithms together and hence we rely on the Yuen-Welch test [26] for statistical significance.

### 2.4.1 Yuen-Welch Test

The Yuen-Welch test is also known as the trimmed t-test. It tests for the null hypothesis that 2 independent sets of samples have identical expected values. It is used to check if any difference observed between two distributions is statistically significant. It is similar to the t-test, however the Yuen-Welch's test does not assume normality or equal variance of the distributions being compared [27].

The Yuen-Welch test is suitable for experiments in Chapter 3 as distributions being compared do not have equal variance and are not normally distributed. The Yuen-Welch test is recommended if the data is contaminated with outliers [26]. During our experiments in Chapter 4, we find there to be outliers due to degenerate runs [28], and hence find the Yuen-Welch test to be suitable for those experiments. We use the

recommended symmetric trimming factor of 20% [29].

In our experiments, we use the Yuen-Welch test implementation available on Python's SciPy library[3].

### 2.4.2 Bonferroni Correction

Bonferroni correction is used to account for the multiple comparisons when several dependent or independent statistical tests are being run on the data simultaneously [30, 31]. The critical p-value used for rejecting the null hypothesis is reduced to prevent false positives results (Type 1 errors).

## 2.5 Conclusions

In this chapter, we have explained the basics of ABSA, Language modeling and Transformer models. In the next chapter, we will show how generative language models face a problem while working with text requiring a coreference resolution ability.

[3]https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.ttest_ind.html

# Chapter 3

# Coreference Resolution in ABSA

## 3.1 Overview

Customer feedback is invaluable to companies as they refine their products. Monitoring customer feedback can be automated with Aspect Level Sentiment Classification (ALSC) which allows us to analyze specific aspects of the products in reviews. Large Language Models (LLMs) are the heart of many state-of-the-art ALSC solutions, but they perform poorly in some scenarios requiring Coreference Resolution (CR). In this work, we propose a framework to improve an LLM's performance on CR-containing reviews by fine tuning on highly inferential intermediate tasks. We show that the performance improvement is likely attributed to the improved model CR ability. We also release a new dataset[1] that focuses on CR in ALSC.

## 3.2 Introduction

Large Language Models (LLMs) are part of state-of-the-art ALSC solutions [3, 32]. However, reviews often use pronouns, which can make coreference resolution (CR) in LLMs necessary to infer the sentiment associated with the aspect. Hence, LLMs used for ALSC need strong CR ability, and can fail otherwise. For instance, the sentence - "*He ate food at the restaurant, it was deserted.*" requires the LLM to understand that the definite pronoun "it" refers to the "restaurant" (antecedent), because of the

[1]Accessible via the Dataverse link: (https://doi.org/10.5683/SP3/HSKJEY)

Table 3.1: Cases where the T5 ALSC model fails due to its poor coreference resolution ability.

| **Sentence** | **Aspect** | **Sentiment Polarity** | |
|---|---|---|---|
| | | **Predicted** | **Gold** |
| He ate food at the restaurant, it was deserted | restaurant | neutral | negative |
| | food | negative | neutral |
| He ate food at the restaurant, it was dark | restaurant | neutral | negative |
| | food | negative | neutral |

context ("deserted"). Table 3.1 shows four examples where the state-of-the-art T5 ALSC model [3] fails due to its poor CR ability. We find that nearly 15% of this T5 model's errors are on cases requiring CR ability.

LLMs are also known to have performance and stability issues [21]. To remedy these, instead of directly training on the task of interest (target task), it can be beneficial to do intermediate training [20]. Intermediate training can be done by first training the model on an intermediate task before training it on the target task. Certain intermediate tasks can contribute to both improved performance and stability of the target task [21]. In our experiments, we explore various intermediate tasks mentioned in Section 3.5. Using intermediate training, our work shows a way to improve an LLM's performance on English ALSC reviews requiring CR.

In our work, we: **a)** show that an LLM trained for ALSC makes more errors when evaluated only on reviews requiring CR ability, compared to when handling typical ALSC reviews (8.7% mean F1); **b)** demonstrate that our framework for handling CR-containing reviews can improve ALSC model's CR ability (16% mean F1); **c)** show that this improved CR ability can improve ALSC performance for reviews requiring CR ability (5% mean F1). **d)** release annotated variants of existing datasets which can be used to benchmark a model's ALSC performance on CR cases.

A limitation of our work is that we only consider reviews containing labeled ground truth aspect terms. This follows prior work on ABSA [33, 34]. We also limit our

analysis to CR-containing reviews which have a definite pronoun[2] as the anaphor. There are other cases[3] of Coreference Resolution where anaphors may not be used, but these cases are less frequent in the considered datasets and hence we ignore them in our research problem. This restriction is hence applied for simplicity of model training (preventing imbalance in dataset).

## 3.3 Coreference Resolution

Coreference Resolution (CR) is the problem of resolving multiple references of the same entity [35]. For instance, the sentence - "*He ate food at the restaurant, it was deserted.*" has the words "it" and "restaurant" referring to the same entity. Here, "it" is an anaphor, referring to the "restaurant" which is the antecedent.

## 3.4 Data

### 3.4.1 Original ALSC Datasets

We consider English ALSC datasets: SemEval Restaurant 2016 (Rest16) [36] and MAMS [32], both of which contain reviews from a similar restaurant domain. Inspired by Yan *et al.* [1], ALSC reviews are processed into an input format suitable for our LLM - "[sentence]. aspect: [aspect]". The ground truth output is "positive", "negative" or "neutral". For example, "*$20 for good sushi cannot be beaten. aspect: sushi*" has the ground truth as "positive".

Following existing work [33, 34] we disregard reviews that do not contain a labeled ground truth aspect term. We also remove the reviews with aspects labeled as having "conflict" sentiment polarity to prevent a class imbalance problem due to the low count of the "conflict" class (label). The "conflict" label for sentiment polarity is used in ABSA datasets to mark a conflicting sentiment for an aspect term.

[2]Definite pronouns refer to specific entities. Example: "he", "she", "it", "who", etc. https://en.wikipedia.org/wiki/English_pronouns

[3]https://en.wikipedia.org/wiki/Coreference

### 3.4.2 CR Cases

We identify reviews in the Rest16 and MAMS datasets that contain definite pronouns, and henceforth call these sentences *Pronoun cases*.

Limiting ourselves to the ALSC task described above, we say that a review is a *CR case* if detecting the aspect's sentiment requires finding the sentiment associated with a definite pronoun referring to the same entity as the aspect. Specifically, *the aspect should be an antecedent of a definite pronoun which is associated with a sentiment polarity*. For example, "He ate food at the restaurant, it was deserted." with aspect: "restaurant" is a CR case. Here, "restaurant" is the antecedent of "it" which is associated with "deserted" and has negative connotations. On the other hand, "He ate food at the restaurant, it was too spicy." with aspect: "restaurant" is a Non-CR case because the aspect is not an antecedent of the pronoun ("it").

CR cases require coreference resolution ability to detect the sentiment associated with the aspect because the aspect is not directly associated with a sentiment. Rather, coreference resolution must first be applied to detect the definite pronoun which is associated with the aspect. The sentiment associated with the aspect would be the sentiment associated with the identified definite pronoun (as they are referring to same entity).

**Identifying CR Cases**: We first identify Pronoun cases from the Rest16 and MAMS datasets by performing a regular expression search for definite pronouns. We then manually go through each of the Pronoun cases to identify the required CR cases.

### 3.4.3 ALSC-CR Dataset

Our dataset is composed of the original ALSC datasets (Rest16 and MAMS). The testing, however, is done only using CR cases, and we use a combination of Pronoun and Non-Pronoun cases for validation and train sets. Table 3.2 presents the dataset composition. Since the ALSC-CR test set only consists of CR cases, a better perfor-

Table 3.2: ALSC-CR composition. Note that CR cases are types of Pronoun cases.

| **Partition** | **Size** | **Dataset** | | **Data Type** | | |
|---|---|---|---|---|---|---|
| | | | | ***Pronoun Cases*** | | ***Non-Pronoun Cases*** |
| | | **MAMS** | **Rest16** | **CR Cases** | **Non-CR Pronoun Cases** | |
| Train | 12,434 | ✓ | ✓ | ✓ | ✓ | ✓ |
| Validation | 889 | ✓ | ✓ | ✓ | ✓ | ✓ |
| Test | 346 | ✓ | ✓ | ✓ | ✗ | ✗ |

mance on the test set will indicate a superior ability to handle CR cases in ALSC. This is the model ability we wish to quantitatively evaluate in our experiments. We do not have Non-CR cases in the test set because Non-CR cases do not test for model's CR ability, and we are only interested in measuring the model's CR ability in our research problem.

The train, validation and test sets are of similar, but not identical, distributions. Due to the limited number of CR cases, it is not possible to have train and validation sets composed entirely of CR cases. The aspect polarity distribution in the ALSC-CR dataset can be seen in Table 3.3. Note that it is possible to have multiple pronouns in each of the CR cases. The sentiment distribution of ALSC-CR test set is shown in Table 3.4.

For constructing ALSC-CR, we used standard ALSC datasets (MAMS and Rest16). MAMS's original train set along with data from Rest16 train set is used for training. For validation, we used the original validation sets from MAMS and Rest16, in addition to Pronoun cases from MAMS test and Rest16. The composition of the validation dataset is such that we use minimal Pronoun cases for validation while having sufficient CR cases for testing. Details of the composition of ALSC-CR are shown in Table 3.5.

Table 3.3: Sentiment polarity distribution in ALSC-CR dataset. Percentage shown corresponds to the percentage of that sentiment polarity of the total size, in the given partition.

| **Partition** | **Polarity** | | | **Total Size** |
|---|---|---|---|---|
| | **Positive** | **Negative** | **Neutral** | |
| Train | 4,279 (34.41%) | 3,065 (24.65%) | 5,090 (40.93%) | 12434 |
| Validation | 337 (37.90%) | 222 (24.97%) | 330 (37.12%) | 889 |
| Test | 178 (51.44%) | 122 (35.26%) | 46 (13.29%) | 346 |

Table 3.4: Pronoun distribution in ALSC-CR test set, which has only CR cases

| **Pronoun** | **Count** |
|---|---|
| it | 132 |
| which | 59 |
| they | 54 |
| he | 24 |
| who | 19 |
| she | 17 |
| their | 14 |
| them | 12 |
| its | 10 |
| his | 10 |
| there | 10 |
| him | 5 |
| her | 5 |
| hers | 0 |

## 3.5 Intermediate Tasks

Since fine-tuning with intermediate tasks can contribute to both improved performance and stability of the target task, we experimented with an intermediate training approach for improving an LLM's performance on cases requiring CR ability [21]. These experiments are explained in Section 3.6.2. We specifically use highly infer-

Table 3.5: Detailed ALSC-CR dataset composition.

| **Partition** | **Size** | **Composition** |
|---|---|---|
| Train | 12,434 | MAMS Train (#count = 11,186) + Rest16 Train (Non Pronoun) (#count = 1,248) |
| Val | 889 | 15% of (MAMS Test (Pronoun) + Rest16 Train/Val/Test (Pronoun)) + 50% of (MAMS Val + Rest Val (Non Pronoun)) [Here, MAMS #count = 746, Rest16 #count = 143] |
| Test | 346 | MAMS Test (CR) (#count = 124) + Rest16 Train/Val/Test (CR cases) (#count = 222) |

ential tasks for intermediate training in the experiments as they generally provide higher improvements for various NLP target tasks [20].

We select two commonsense tasks - Commongen [5] and CosmosQA [37], as commonsense reasoning helps with CR [38]. SQuAD [39] is selected because it is a non-commonsense question answering (QA) task. Its performance is contrasted with CosmosQA, checking if it is the QA or the commonsense ability which improves CR. Quora Question Pairs [4] (QQP) is selected as it benefits performance on the Stanford Sentiment Treebank (SST) task which is similar to ALSC [40]. Even if intermediate tasks are not designed for CR, they can impart CR ability to the model. For the QA example - "Context: Alice can't come. She is old"; "Question: Who is old?", answer is "Alice". Answering this requires CR and teaches the model CR ability.

**Commongen** is a generative commonsense task involving the generation of a plausible sentence given a list of concepts (train size = 67,389). It tests: 1) relational reasoning which is the ability to construct grammatical sentences adhering to commonsense; 2) compositional generalisation which is reasoning with unseen concept combinations. For example, we can have: input - "concepts = [dog, frisbee, catch, throw]"; output - "A dog leaps to catch a thrown frisbee."

**CosmosQA** is a QA task where answering questions requires commonsense (train size = 25,262). For each question, there are four options, and the model should output the correct option number.

**SQuAD** is an extractive QA task where the correct answer to the question is

present exactly in the passage (train size = 87,599).

**QQP** task involves checking if two Quora questions are semantically equivalent. We cap the train size at 50,000 to match the other datasets.

For our experiments, we only use one intermediate task for each intermediate training step. However, we note that another possible approach is to use a mixture of intermediate tasks for the intermediate training step. Such an approach could be promising because different intermediate tasks could help with different problems faced by the model. However, we do not explore such a mixture-based approach due to the complexities involved in selecting the appropriate intermediate tasks in the right proportions [15].

## 3.6 Experiments and Results

We ran experiments for three purposes: **a)** to quantify a drop in ALSC performance for reviews requiring CR ability; **b)** to demonstrate that we can alleviate this performance drop by intermediate-task fine-tuning; **c)** to provide additional evidence that change in performance on CR cases is due to improved CR ability.

Inspired by state-of-the-art performance in Zhang *et al.* [3], we used the T5[4] LLM [15]. Our baseline model is a T5 trained on ALSC-CR, but not fine-tuned on intermediate tasks. All experiments were run with at least 10 random seeds. Yuen-Welch test was used for testing statistical significance, along with Bonferroni correction to account for multiple comparisons.

The T5 model was trained in various settings using training prompts / input prefixes shown in Table 3.6. The wording of prompts has limited impact on the outcome so we did not experiment with the wording [15]. Rather, we relied on prior work for task prompts [5, 15, 41]. For ALSC and Definite Pronoun Resolution (DPR) [42] (in Section 3.6.3), we created prompts as we did not find examples in the literature.

[4]T5-large from https://huggingface.co/t5-large

Table 3.6: Details of T5 training prompts used for intermediate and target tasks.

| **Task** | **Training Prompt** |
|---|---|
| ALSC-CR | get sentiment: [sentence, aspect] |
| ALSC-Regular | get sentiment: [sentence, aspect] |
| DPR | Get antecedent: [sentence] |
| Commongen | generate a sentence with: [concepts] |
| CosmosQA | question: [question] answer_0: [ans_0] answer_1: [ans_1] answer_2: [ans_2] answer_3: [ans_3] context: [context] |
| SQuAD | question: [question] context: [context] |
| QQP | qqp question1: [question_1] question2: [question_2] |

### 3.6.1 Model Performance on ALSC Without Intermediate Training

To check LLM performance on CR cases, we evaluated the T5 model on regular ALSC data (ALSC-Regular), which does not consist solely of CR cases. ALSC-Regular and ALSC-CR are equal sized and have an identical proportion of Rest16 and MAMS. We also evaluated the T5 model on ALSC-CR, to get the model's performance solely on CR cases.

ALSC-Regular and ALSC-CR are created from the same standard ALSC datasets (Rest16 and MAMS). Both datasets have the same distribution apart from the presence of CR in all examples of ALSC-CR's test set. So the only difference between between the datasets is the necessary presence of CR in all ALSC-CR samples, but only an optional presence of CR in ALSC-Regular samples.

By comparing T5 model's performance on the two ALSC datasets, we show that unspecialized LLMs face a significant performance problem while handling reviews requiring CR ability. Results are shown in Table 3.7, where evaluation on ALSC-CR shows a drop in performance of 8.7% mean F1, as well as an increase of 0.6 F1 standard deviation indicating a poorer model convergence.

Table 3.7: T5 model evaluated on ALSC datasets. Best score bolded. Performances on the datasets are statistically significantly different (p-value=$9.03e-05$).

| **Dataset** | **Mean F1 (± Std. Dev)** |
|---|---|
| ALSC-Regular | **79.71** (± 1.99) |
| ALSC-CR | 71.07 (± 2.60) |

### 3.6.2 Fine Tuning With Intermediate Tasks

As a solution to poor performance on ALSC-CR (Section 3.6.1), we experimented with various intermediate tasks mentioned in Section 3.5.

We trained T5 model on the intermediate task first to incorporate intermediate task knowledge (Step 1). This model is then trained and evaluated on ALSC-CR, our target task (Step 2). Both these steps are shown in Figure 3.1. We experimented with different intermediate-task dataset sizes (fraction) as the size has little correlation with the target task performance [40]. We used the dataset fractions - 0.1, 0.2, 0.5 and 1.0, for simplicity (roughly doubling the fraction from 0.1 to 1.0). However, a more comprehensive experiment would involve evaluations using other dataset fractions as well, which could be randomly sampled from 0 to 1.0.

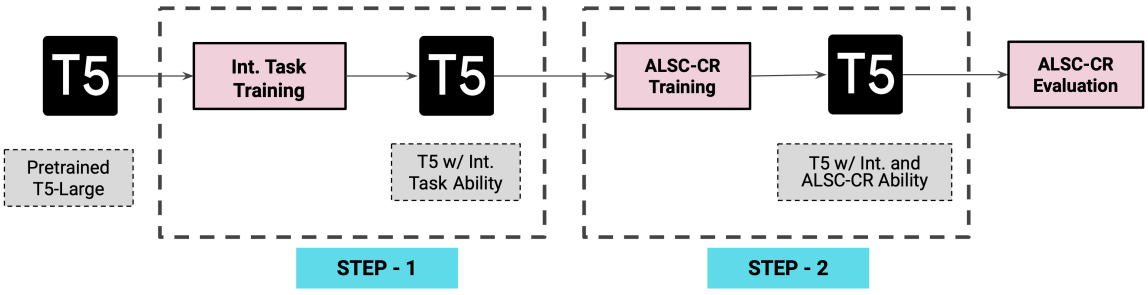



Figure 3.1: Fine tuning a T5-large model with intermediate (Int.) tasks prior to training and evaluation on the target ALSC-CR task.

The model's performance on ALSC-CR with different intermediate tasks is compared to baseline model's ALSC-CR performance to see if intermediate tasks were beneficial. The results are shown in Table 3.8. We find that the lower ALSC-CR performance (compared to ALSC-Regular) can be alleviated by intermediate training

Table 3.8: Mean F1 (± Std. Dev) performance on ALSC-CR on different fractions of intermediate-task dataset. * denotes statistically significant difference from baseline. Table's best scores bolded, $2^{nd}$ best underlined.

| **Intermediate Task** | **Intermediate-Task Dataset Fraction** | | | |
|---|---|---|---|---|
| | **0.1** | **0.2** | **0.5** | **1.0** |
| Commongen | <u>75.72</u> (± 1.14) * | 72.46 (± 2.21) | 71.04 (± 3.50) | 71.45 (± 1.91) |
| CosmosQA | 71.79 (± 1.55) | 71.45 (± 3.02) | 72.60 (± 1.85) | 73.12 (± 2.15) |
| SQuAD | 72.02 (± 1.88) | 72.60 (± 2.07) | 71.47 (± 3.24) | 72.08 (± 2.25) |
| QQP | 72.49 (± 2.79) | 71.85 (± 2.98) | **76.10** (± 1.26) * | 71.30 (± 2.19) |
| N/A (Baseline) | 71.07 (± 2.60) | | | |

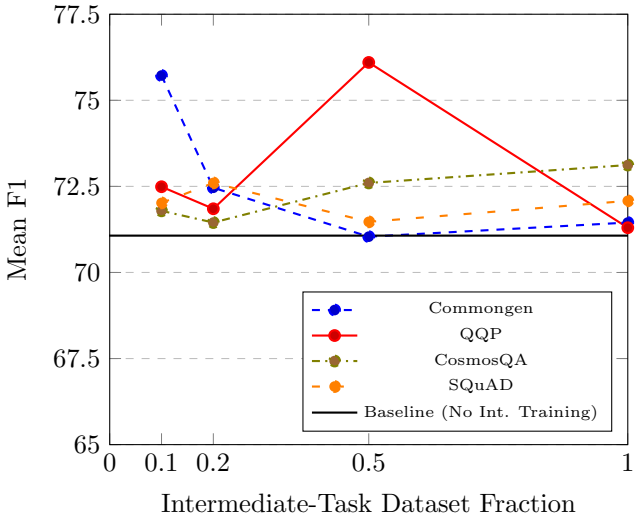



Figure 3.2: Performance of ALSC models with intermediate (Int.) training on ALSC-CR dataset.

with Commongen and QQP, which lead to statistically significant improvements of 5% mean F1. Intermediate training with CosmosQA and SQuAD does not lead to statistically significant improvement in any case.

Prior work [20] showed a general improvement in a model's target task performance when fine-tuned with highly inferential tasks. Apart from being highly inferential, because Commongen is a generative commonsense task, it is ideal for imparting commonsense knowledge to a generative LLM like T5. On the other hand, CosmosQA being a discriminative task is unlikely to impart as much commonsense knowledge

into a generative system [5]. As being highly inferential is helpful for target tasks, the SQuAD extractive QA task, would not result in as significant an improvement. When used for intermediate training, QQP shows a high improvement in the SST target task [40] which involves similar sentiment analysis, explaining QQP's improved performance on ALSC-CR.

Though intermediate training using DPR task might seem promising, it is a much smaller dataset (train size = 1500) than other tasks. For completeness we did train using DPR but found that the mean F1 = 72.77 was not statistically significantly different from the baseline.

Similar to Wang *et al.* [40], we do not find a correlation between intermediate-task dataset fraction and target performance. This can be seen in Figure 3.2 where the target performance does not follow an increasing or decreasing trend with respect to the intermediate-task dataset fraction. This lack of correlation can be attributed to small dataset size might not teach the task sufficiently [15]. On the other hand, large intermediate-task datasets can cause catastrophic forgetting of the LLM's original objective [40]. This original objective is generally beneficial for target tasks. There is hence a need to strike a balance between the amount of intermediate-task ability we wish to impart and the amount of original objective catastrophic forgetting that can be endured. Despite the lack of correlation between intermediate-task dataset fraction and target performance, we have demonstrated a framework for improving any target task's performance on CR cases.

We further provide a pronoun error analysis in Section 3.7 to better understand the improvements seen due to fine-tuning with intermediate tasks.

### 3.6.3 Evaluating Coreference Ability

Performing well on ALSC-CR requires strong CR ability, as CR associates the aspect with its sentiment. To verify that the improvement in Section 3.6.2 is attributable to the ALSC model's improved CR ability, we estimate the CR ability by evaluating

on DPR. For every intermediate task, we evaluate the model corresponding to the dataset fraction with the best mean F1 (in Section 3.6.2 experiments). For example, we evaluate the 0.1 fraction for Commongen and the 1.0 fraction for CosmosQA. Since we have an ALSC model for each random seed used for training (Section 3.6.2), we run DPR evaluation on the ALSC random seed model with the highest ALSC-CR val set performance. The steps for evaluating on DPR after training on an intermediate task are shown in Figure 3.3. Step 1 and Step 2 lead to the intermediate-task trained model from Section 3.6.2. Step 3 performs the required DPR evaluation.

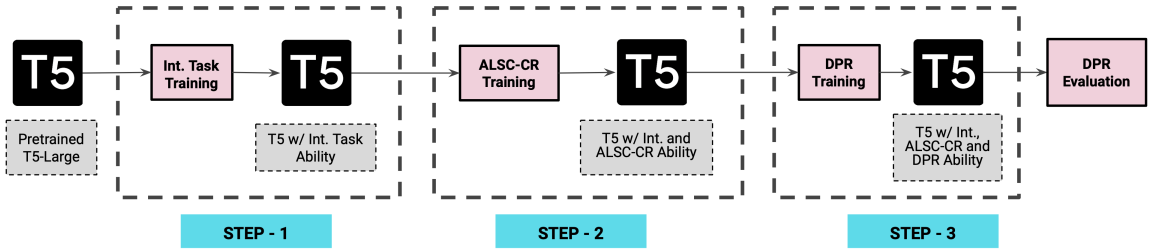



Figure 3.3: Evaluating a T5-large model with the DPR task to check for Coreference Resolution ability. Step 1 involves fine-tuning the T5 model on an intermediate (Int.) task. Then, in Step 2, the model is trained on the required ALSC-CR task. Here, we have obtained the model from Sec 3.6.2. Now, in Step 3, the model is trained and evaluated on the DPR task, to test for CR ability.

The DPR task involves predicting the antecedent of the given pronoun. This is precisely the ability required for good performance on ALSC-CR (which contains only definite pronoun cases), making DPR ideal to measure the CR ability of our models. Other CR datasets like OntoNotes [43] are not as suitable as DPR because DPR only focuses on definite pronouns, which is the ability we are interested in. Similarly, DPR is also the only CR dataset suitable for intermediate training, but the small size makes this infeasible as discussed in Section 3.6.2.

We use a DPR variant for generative models where input is of the form: "Humans were afraid of robots as *they* were strong.", and the objective is to predict what the highlighted pronoun (*they*) is referring to [15].

Evaluating ALSC models on DPR (Table 3.9) confirms that the ALSC-CR per-

Table 3.9: CR ability of top performing models (Sec 3.6.2) measured using DPR. Statistically significant improvement(*) and deterioration($^\dagger$) from baseline marked. Best bolded, $2^{nd}$ best underlined.

| **Intermediate Task** | **Intermediate-Task Dataset Fraction** | **Mean F1 (± Std. Dev)** |
|---|---|---|
| N/A (Baseline) | 0 | 59.28 (± 8.82) |
| Commongen | 0.1 | <u>75.77</u> (± 1.68)* |
| CosmosQA | 1.0 | 54.55 (± 7.19)$^\dagger$ |
| SQuAD | 0.2 | 62.91 (± 6.77) |
| QQP | 0.5 | **76.36** (± 2.16)* |

formance gains may be attributable to the improved CR ability of the model due to intermediate training. Experiments show that Commongen and QQP fine-tuned models show a drastically improved (and statistically significant) CR ability of up to 16%. This explains their improved ALSC-CR performance. Using CosmosQA, we see a statistically significant 5% deterioration in CR ability which does not lead to statistically significant changes in ALSC-CR performance.

## 3.7 Error Analysis by Pronoun

We analyzed the errors and improvements seen for individual pronouns (in reviews) when ALSC-CR is evaluated with different ALSC models. Since a few pronouns have very low counts as per Table 3.4, we only analyzed pronouns with a count greater than 15. For all pronouns analyzed, we found improvements in prediction accuracy for the models fine-tuned with intermediate tasks, compared to the baseline model which has no intermediate training. Results are shown in Table 3.10.

## 3.8 Related Work

The importance of CR has been noted in prior ABSA work. Ding and Liu [44] use aspect sentiments for performing CR, demonstrating a correlation between CR and sentiment classification. De Clercq and Hoste [45] use CR to detect aspects from re-

Table 3.10: Error Analysis of ALSC models by pronoun distribution. Model Accuracy% presented by Pronoun. Highest scores bolded. $2^{nd}$ highest underlined. Pronouns with count less than 15 (as per Table 3.4) are not analyzed.

| **Pronoun** | **Baseline** | **Commongen 0.1** | **QQP 0.5** |
|---|---|---|---|
| it | 65.91 | <u>68.18</u> | **71.21** |
| which | 74.58 | **83.05** | <u>77.97</u> |
| they | 72.22 | **79.63** | <u>77.78</u> |
| he | 70.83 | **75.0** | <u>70.83</u> |
| who | 84.21 | **94.74** | **94.74** |
| she | <u>88.24</u> | **94.12** | <u>88.24</u> |

lated reviews, for the reviews lacking explicit aspects. Instead, we consider an LLM's intra-sentence CR ability, considering only reviews with explicit aspects as having an aspect is critical to ALSC. Mai and Zhang [46] use CR in aspect extraction, but only for identifying duplicate references among proposed aspects.Varghese and Jayasree [47] use CR to solve their dependency parser component's inability to correctly associate opinion words with pronouns. In our work, we consider the CR problem in end-to-end state-of-the-art ALSC LLM models. Chen *et al.* [48] improve BERT LLM's CR ability for opinion-mining, using a method relying on external knowledge bases.

## 3.9 Conclusions

Real world reviews vary widely and can frequently contain pronouns. While building an ALSC model to handle all kinds of reviews, it is crucial to know how its performance is on reviews requiring CR ability. In case of inadequate performance, mitigation steps can be taken. Since our research problem only involves model performance on CR-requiring reviews (CR cases), we do not evaluate the effect of the mitigation step on the performance on Non-CR cases. However, since CR cases are tougher to perform on, it is reasonable to expect that improving performance on CR

cases will not lead to a deterioration on the simpler Non-CR cases.

Although LLMs generally perform well on ALSC, our experiments provide evidence that LLMs can have poor performance on ALSC reviews requiring CR ability. We show that this problem can be alleviated by fine-tuning with certain intermediate tasks before fine-tuning on the target tasks. Our framework for evaluating and improving an LLM's performance on CR cases can be applied for other target tasks as well. We note that the intermediate tasks and the associated intermediate-task dataset fractions identified for the ALSC-CR target task may not be suitable for other target tasks. This may happen if the target task objectives or dataset distributions are different from that of ALSC-CR. Due to factors such as catastrophic forgetting of the original objective, it is non-trivial to hypothesize about the intermediate task and dataset fraction which would lead to performance improvements for target tasks other than ALSC-CR. However, our framework details a sound empirical approach to identifying intermediate tasks and dataset fractions, for any target task to handle CR cases. Such a framework is critical for developing any model deployed in the real world. In the future, we will explore if intermediate training can reduce the target task training that is needed for CR cases.

## 3.10 Chapter Appendix

### 3.10.1 Hyperparameters

Learning rates for both intermediate-task training and ALSC training steps are picked from $\{5e-4, 1e-4, 5e-5\}$ and $\{1e-3, 5e-4, 1e-4\}$ respectively, after running for three random seeds and selecting the rates giving max F1 score for their respective validation dataset. For intermediate-task training, the learning rates for all intermediate tasks were found to be $1e-4$, except for SQuAD with Intermediate-Task Dataset Fraction as 1.0 for which we found learning rate as $5e-5$. For ALSC target task training, the learning rate was found to be $5e-4$ in all cases except when using Commongen task for fine tuning with Intermediate-Task Dataset Fraction as 0.1 for which we found learning rate as $1e-4$.

Batch size for training is taken as 16 to maximize GPU utilization. We train for 30 epochs to allow for convergence, while using an early stopping mechanism.

### 3.10.2 Training Details

For fine tuning the T5-large model, we use 1 NVIDIA V100 GPU, 6 CPU cores with 4 GB memory per core. We run training jobs with a 71 hour time limit.

# Chapter 4

# Cross Lingual and Cross Domain Experiments in ABSA

## 4.1 Overview

Aspect-based Sentiment Analysis (ABSA) helps to explain customers' opinions towards products and services. In the past, ABSA models were discriminative, but recently generative models have been used to generate aspects and polarities directly from reviews. Previous results showed that generative models outperform discriminative models on several English ABSA datasets. Here, we evaluate and contrast two state-of-the-art discriminative and generative models in several settings involving different language and/or domain for training and testing. This is done to understand generalizability in settings other than English mono-lingual in-domain (English language and the same domain used for both training and testing). Our evaluation shows that discriminative models can still outperform generative models in a few settings. Further, we present a problem faced by generative models in cross-lingual settings, and demonstrate a mitigation strategy that combines the best of discriminative and generative models.

## 4.2 Introduction

A few natural languages receive more research effort compared to other languages (e.g. English vs. Swahili). Although the community has remarkably accelerated the

improvement of English NLP techniques, techniques for other languages lag behind. Working on a lower resource language is a challenging task, where few datasets, lexicons, and models exist. Thus, utilizing cross-lingual approaches is important to migrate model ability across languages.

ABSA involves predicting aspect terms and their associated sentiment polarities [7]. Training such an ABSA model requires a suitable amount of data. Hence, in low resource settings, it can be difficult to use ABSA to analyze reviews. A solution is to use models which were trained in some other setting [49]. For example, to perform ABSA in Swahili, use a model trained on ABSA in English. For such settings, we conduct a comparative study of discriminative and generative ABSA models.

Prior work shows that generative models achieve better performance than discriminative models in the English in-domain setting (English language and the same domain used for both training and testing) [1, 3]. However, none have explored performance in the cross-lingual setting (train language different from test language) or the cross-domain setting (train domain different from test domain). These settings are important to cases involving a low resource domain or language. For example, the cross-lingual setting can be relevant for evaluating an ABSA model on reviews in Swahili (low resource language). For such an evaluation, we can use the cross-lingual setting of training in a different language (like English) prior to the evaluation in Swahili.

In our work: **a)** We evaluate the performance of the two model types in various settings by comparing state-of-the-art representatives, and demonstrate that discriminative models can still perform better than generative models in a few cases; **b)** We find that generative models face problems with aspect extraction in cross-lingual settings. As a mitigation, we propose a masking approach that gives large improvements (up to 20%), by combining a generative model's semantic understanding of labels, along with a discriminative model's constraint of only generating words from the input sentence.

A limitation of our work is that we only consider reviews containing labeled ground truth aspect terms. This follows prior work on ABSA [33, 34].

## 4.3 Data

In our experiments, we considered various languages and domains for evaluating models. For languages, we used SemEval datasets - Restaurant (Rest16) [36] in English, Spanish and Russian. For domains, we used Rest16 and Laptop (Lap14) from SemEval [50]. Additionally, we used the MAMS dataset [32] of the restaurant domain. In MAMS, each sentence contains at least two aspects with different polarities, making the dataset more challenging than the SemEval datasets.

For SemEval datasets, since the validation sets are not given, we sampled 10% of the training dataset to use for validation. The datasets we considered vary in terms of the type of content and the training set size. Table 4.1 presents the datasets' statistics after cleaning and sampling. For reference, we also provide Table 4.2 which has the datasets' statistics before cleaning and sampling. Note that originally, SemEval datasets only contain testing and training datasets (no validation dataset).

Following existing work [33, 34] we disregard reviews that do not contain a labeled ground truth aspect term. We also remove the reviews with aspects labeled as having "conflict" sentiment polarity to prevent a class imbalance problem due to the low count of the "conflict" class (label). The "conflict" label for sentiment polarity is used in ABSA datasets to mark a conflicting sentiment for an aspect term.

## 4.4 Models

We perform experiments on two generative and one discriminative ABSA model. We contrast the generative and discriminative ABSA paradigms (explained in Chapter 2.3.3), by considering a representative model for each, which shows state-of-the-art performance. We also consider a generative model constrained to generate aspects

Table 4.1: Filtered (cleaned) datasets' statistics - Count of aspects with sentiment polarities for the cleaned datasets. Multiple aspects can exist in single record

| Datasets | Data Split | #Pos | #Neg | #Neu |
|---|---|---|---|---|
| **Rest16**$_{En}$ | Train | 1028 | 380 | 54 |
| | Val | 130 | 32 | 6 |
| | Test | 427 | 119 | 28 |
| **Rest16**$_{Es}$ | Train | 972 | 338 | 72 |
| | Val | 101 | 46 | 5 |
| | Test | 420 | 142 | 29 |
| **Rest16**$_{Ru}$ | Train | 2044 | 411 | 188 |
| | Val | 223 | 56 | 23 |
| | Test | 608 | 193 | 85 |
| **Lap14**$_{En}$ | Train | 895 | 799 | 414 |
| | Val | 99 | 71 | 50 |
| | Test | 341 | 128 | 169 |
| **MAMS**$_{En}$ | Train | 3382 | 2769 | 5042 |
| | Val | 403 | 325 | 605 |
| | Test | 400 | 330 | 607 |

from the input sentence.

**1) SPAN-MBERT Discriminative model**: we considered the SPAN-BERT model [8] which is a state-of-the-art ABSA model that uses a BERT transformer. It has a good performance on mono-lingual in-domain datasets, and has been used as a baseline for generative models [1, 3]. The model extracts continuous spans of text for multiple target aspect terms and classifies their polarities using contextualized span representations. To use this methodology in cross-lingual settings, we replace SPAN-BERT model's encoder with a multilingual BERT model[1]. We call this ABSA methodology - "SPAN-MBERT".

[1]https://github.com/google-research/bert/blob/master/multilingual.md

Table 4.2: Unfiltered datasets' statistics - Count of aspects with sentiment polarities for the unfiltered datasets. Multiple aspects can exist in single record. A single aspect can have multiple sentiment polarities associated.

| **Datasets** | **Data Split** | **#Pos** | **#Neg** | **#Neu** |
|---|---|---|---|---|
| **Rest16**$_{En}$ | Train | 1657 | 749 | 101 |
| | Test | 611 | 204 | 44 |
| **Rest16**$_{Es}$ | Train | 1925 | 674 | 120 |
| | Test | 750 | 274 | 48 |
| **Rest16**$_{Ru}$ | Train | 3103 | 709 | 276 |
| | Test | 870 | 321 | 103 |
| **Lap14**$_{En}$ | Train | 1637 | 1084 | 188 |
| | Test | 481 | 274 | 46 |
| **MAMS**$_{En}$ | Train | 3380 | 2764 | 5042 |
| | Val | 403 | 325 | 604 |
| | Test | 400 | 329 | 607 |

**2) mBART Non-Masked Generative model**: we used an encoder-decoder BART-based approach [1]. This model takes a review as input and generates aspects and their polarities. The aspect-polarity terms have the format: "service positive <sep> food negative", indicating presence of two aspect terms ("service" and "food"), with associated polarities ("positive" and "negative"). A separator token "<sep>" is used to demarcate a separation between the multiple aspect-polarity pairs in a review. To use this approach in cross-lingual settings, we use a multilingual BART model[2]. We call this methodology - "mBART Non-Masked" ("Non-Masked" term explained in Section 4.5).

**3) mBART Masked Generative model**: this model is a modification of the mBART Non-Masked model. It uses masking to constrain the model to generate aspects from the input sentence itself (details in Section 4.5).

[2]huggingface implementation https://huggingface.co/facebook/mbart-large-cc25

The SPAN-MBERT model and both mBART models use a similar mBERT-like encoder, implying that performance differences between them is only due to their discriminative and generative components and not because of the input sentence's encoding.

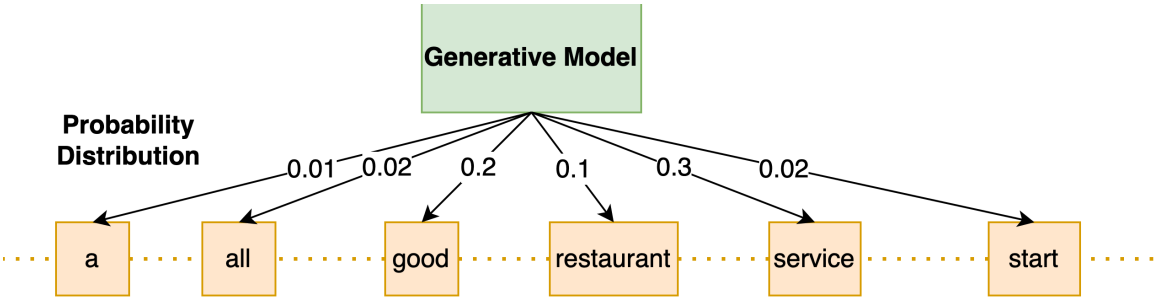
## 4.5 Logit Masking and the mBART Masked Model

The ABSA task involves finding aspect terms which are phrases within the input sentence. Discriminative models generally label each word in the input sentence as an aspect or non-aspect [9, 23]. An example of the discriminative model's working can be seen in Figure 4.1. Generative models, however, are not restricted to generating words from the input sentence. They can generate words from the entire vocabulary and are hence at risk of generating out-of-sentence words ("hallucinating") [51]. An example of this problem can be seen in Figure 4.2a where the generative model, models the probability distribution for the aspect, however, it assigns non-zero probabilities to words not existing in the input sentence.

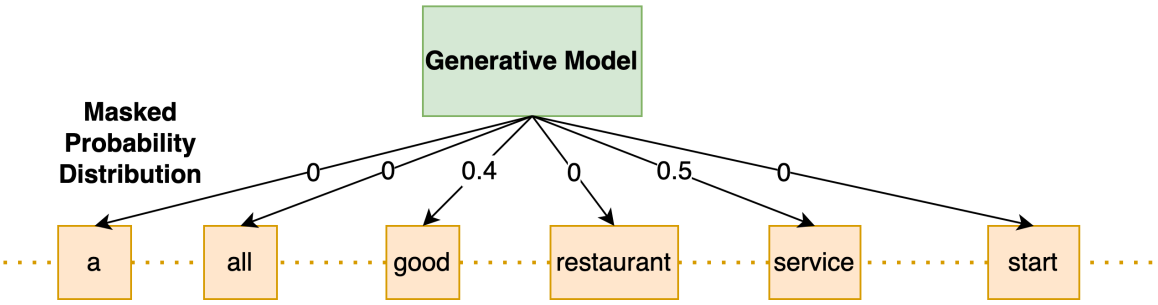
This hallucination problem for the generative mBART Non-Masked model is confirmed in the error analysis (Section 4.8). A solution to this problem is to constrain the generative model's decoding step so that the model can only generate from a fixed set of tokens that exist in the input sentence [52, 53].

We implemented this solution for mBART by "masking" tokens not in the input sentence. Masking changes a token's softmax inputs to -Infinity before calculating softmax probabilities. This ensures that the generative model can only select a token that appears in the input sentence. We call this the "mBART Masked" model. This model combines the best of other models - a generative model's semantic understanding of labels [3], and a discriminative model's constraint of only generating words from the input sentence. An example of this masking solution can be seen in Figure 4.2b where the generative model models the probability distribution for the aspect while assigning a zero probability to words not existing in the input sentence.


Figure 4.1: A discriminative ABSA model identifying aspects from within the example sentence - "The service was good". A checkmark indicates that the word is an aspect, and a cross indicates that the word is not an aspect.




(a) Probability distribution without masking.




(b) Probability distribution with masking.

Figure 4.2: Example probability distribution of the aspect term when a generative ABSA model is generating the aspect in the sentence - "The service was good".

## 4.6 Text Normalization Process

Prior to evaluation, the model outputs and the gold data are normalized. We remove punctuation marks such as ",", ".", "" from the sentences, lower-case and lemmatise the words, and remove common stop words. This is because the generative model often generates a different variant of a term, e.g. plural or singular. This idea of normalizing the generated output is similar to Zhang *et al.* [3], where Levenshtein distance is used to align the generated aspect words with the closest words existing in the original sentence. Compared to this, our normalization process followed by an

exact matching is stricter. Levenshtein distance may align the model's predictions with unrelated words in the original sentence. For example, if a generated word - "salmon", has the least distance with the word "not" out of all the words in the original sentence, then "salmon" can get aligned to "not", as is mentioned by Zhang *et al.* [3], which is a loose matching.

## 4.7 Experiments and Results

For the three models mentioned in Section 4.4, we ran experiments under different settings. We compare **1)** the mBART Non-Masked model against the SPAN-MBERT model to check if generative models indeed outperform discriminative models in various settings; **2)** the mBART Non-Masked model against the mBART Masked model to see if constrained decoding is beneficial.

We ran all experiments with 10 random seeds for robustness. Yuen-Welch test was used for testing statistical significance, along with Bonferroni correction to account for multiple comparisons. We find that the standard deviation in results is high because of degenerate runs. Transformer-based models are known to produce degenerate runs when fine-tuned on small datasets [28].

Once model outputs and gold data are normalized as per Section 4.6, the predicted aspect-polarity terms and the corresponding gold aspect-polarity terms are compared using an exact match. We consider a hit only if both aspect term and the polarity term match. We use the standard evaluation metrics for calculating ABSA scores, which are Micro- Precision, Recall and F1. We use the evaluation code released by Li *et al.* [23][3].

### 4.7.1 Monolingual and In-Domain

In the mono-lingual in-domain setting, we evaluated models with train and test data from the same domain and language. As shown in Table 4.3, the mBART Non-

[3]http://github.com/lixin4ever/E2E-TBSA

| **Domain**$_{Lang}$ | **SPAN-MBERT** | **mBART Non-Masked** | **mBART Masked** |
|---|---|---|---|
| **Rest16**$_{En}$ | 60.96 (± 2.15)* | **74.17** (± 2.13) | <u>74.02</u> (± 2.17) |
| **Rest16**$_{Es}$ | 64.72 (± 1.19)* | **69.83** (± 1.28) | <u>69.50</u> (± 1.34) |
| **Lap14**$_{En}$ | 57.20 (± 1.51)* | <u>66.35</u> (± 2.70) | **66.65** (± 2.21) |
| **MAMS**$_{En}$ | **66.00** (± 0.42)* | <u>61.14</u> (± 1.20) | 60.97 (± 1.18) |
| **Rest16**$_{Ru}$ | 54.60 (± 2.27)* | **68.55** (± 1.28) | <u>68.38</u> (± 1.13) |

Table 4.3: Mono-lingual and in-domain F1 scores. * SPAN-MBERT statistically significantly different from mBART Non-Masked. † mBART Masked statistically significantly different from mBART Non-Masked. For every setting, the highest model score is bolded and the $2^{nd}$ highest model score is underlined.

Masked model performs better than the SPAN-MBERT model in all cases except the MAMS case. The mBART Non-Masked model and the mBART Masked models have no statistically significant difference in performance.

### 4.7.2 Cross-Lingual

In the cross-lingual setting, we considered datasets from the same domain but different languages. The models were trained on a dataset from one language and were evaluated on a dataset from another language. For example, the model could be trained on Rest16$_{En}$ (English) and tested on Rest16$_{Es}$ (Spanish).

Table 4.4 presents the cross-lingual results. Except in a case involving testing in Spanish, the generative model (mBART Non-Masked) has a better performance than the discriminative model. Moreover, when testing in Spanish, the mBART Masked model provides a significant improvement (up to 16%) in performance over the Non-Masked model.

### 4.7.3 Cross-Domain

In the cross-domain setting, we considered datasets from the same language but from different domains. The models were trained on a dataset from one domain and were evaluated on a dataset from another domain, but in the same language. For example,

| Train → Test | SPAN-MBERT | mBART Non-Masked | mBART Masked |
|---|---|---|---|
| *Es* → *En* | 48.87 (± 2.22)* | <u>55.88</u> (± 14.96) | **61.90** (± 12.72) |
| *Ru* → *En* | 32.89 (± 6.16) * | <u>64.77</u> (± 3.62) | **67.75** (± 3.91) |
| *En* → *Ru* | 39.86 (± 1.89) * | <u>50.66</u> (± 8.12) | **54.02** (± 10.84) |
| *Es* → *Ru* | 37.44 (± 1.76) * | <u>50.76</u> (± 10.19) | **52.41** (± 8.66) |
| *En* → *Es* | <u>54.42</u> (± 2.44)* | 42.79 (± 4.43) | **58.41** (± 2.84) † |
| *Ru* → *Es* | 28.20 (± 4.87) * | <u>55.03</u> (± 5.68) | **63.13** (± 1.89) † |

Table 4.4: Cross-lingual F1 scores using Rest16 in several languages. * SPAN-MBERT statistically significantly different from mBART Non-Masked. † mBART Masked statistically significantly different from mBART Non-Masked. For every setting, the highest model score is bolded and the $2^{nd}$ highest model score is underlined.

| Train → Test | SPAN-MBERT | mBART Non-Masked | mBART Masked |
|---|---|---|---|
| **Rest16**$_{En}$ → **Lap14**$_{En}$ | 31.32 (± 1.74)* | <u>41.04</u> (± 3.61) | **41.58** (± 3.62) |
| **MAMS**$_{En}$ → **Lap14**$_{En}$ | <u>31.57</u> (± 2.71) | 31.23 (± 3.03) | **31.97** (± 2.95) |
| **Lap14**$_{En}$ → **Rest16**$_{En}$ | 42.06 (± 2.71)* | <u>55.28</u> (± 5.01) | **57.49** (± 3.07) |
| **MAMS**$_{En}$ → **Rest16**$_{En}$ | **56.04** (± 1.30)* | <u>50.06</u> (± 2.12) | 49.52 (± 1.98) |
| **Rest16**$_{En}$ → **MAMS**$_{En}$ | 32.32 (± 2.00)* | **36.10** (± 0.93) | <u>35.96</u> (± 0.89) |
| **Lap14**$_{En}$ → **MAMS**$_{En}$ | 23.57 (± 2.19) | <u>29.07</u> (± 3.30) | **30.26** (± 2.13) |

Table 4.5: Cross-domain F1 scores. Bolded results are the best per model and test language. * SPAN-MBERT statistically significantly different from mBART Non-Masked. † mBART Masked statistically significantly different from mBART Non-Masked. For every setting, the highest model score is bolded and the $2^{nd}$ highest model score is underlined.

the model could be trained on Rest16$_{En}$ and tested on Lap14$_{En}$.

Table 4.5 presents the cross-domain results. The generative mBART Non-Masked model performs better than the SPAN-MBERT model, except in cases involving MAMS$_{En}$ in either the test or train domain. In those cases, three times out of four, the SPAN-MBERT model does equal to better than the mBART Non-Masked model. In none of the settings does the mBART Masked model provide a statistically significant improvement over the mBART Non-Masked model.

| Train → Test | SPAN-MBERT | mBART Non-Masked | mBART Masked |
|---|---|---|---|
| **Rest16**$_{Es}$ → **Lap14**$_{En}$ | 28.52 (± 2.72)* | <u>33.84</u> (± 10.89) | **35.91** (± 9.73) |
| **Rest16**$_{Ru}$ → **Lap14**$_{En}$ | 16.80 (± 1.84)* | <u>39.56</u> (± 2.31) | **40.32** (± 2.71) |
| **Lap14**$_{En}$ → **Rest16**$_{Es}$ | <u>42.26</u> (± 4.22) | 36.06 (± 9.09) | **47.42** (± 5.14) |
| **MAMS**$_{En}$ → **Rest16**$_{Es}$ | **47.33** (± 1.26)* | 20.01 (± 2.54) | <u>39.20</u> (± 3.32) † |
| **Lap14**$_{En}$ → **Rest16**$_{Ru}$ | 31.58 (± 7.76)* | <u>42.27</u> (± 8.02) | **42.57** (± 11.01) |
| **MAMS**$_{En}$ → **Rest16**$_{Ru}$ | 30.75 (± 3.57) | <u>34.16</u> (± 3.72) | **39.42** (± 2.41) |
| **Rest16**$_{Es}$ → **MAMS**$_{En}$ | <u>28.78</u> (± 1.39) | 25.37 (± 7.25) | **29.05** (± 6.77) † |
| **Rest16**$_{Ru}$ → **MAMS**$_{En}$ | 14.81 (± 3.23)* | <u>31.50</u> (± 1.51) | **32.35** (± 1.33) |

Table 4.6: Cross-domain and cross-lingual F1 scores. * SPAN-MBERT statistically significantly different from mBART Non-Masked. † mBART Masked statistically significantly different from mBART Non-Masked. For every setting, the highest model score is bolded and the $2^{nd}$ highest model score is underlined.

### 4.7.4 Cross-Lingual and Cross-Domain

In the cross-lingual and cross-domain experiments, we evaluated models in an extreme setting which combines the previous cross-lingual and cross-domain settings. The models were trained on a dataset from a domain in a language and were then evaluated on a dataset from another domain and another language. For example, the model could be trained on Rest16$_{Es}$ and tested on Lap14$_{En}$.

Table 4.6 shows the evaluation results. In 50% of the cases, the mBART Non-Masked model does better than the SPAN-MBERT model. In the rest, the SPAN-MBERT performs better or equal to the mBART Non-Masked model. Only in the cases involving both MAMS$_{En}$ and Rest16$_{Es}$ for training/testing or testing/training, does the mBART Masked model have a statistically significant improvement over the Non-Masked model (of up to 20%).

## 4.8 Error Analysis

We conducted an error analysis on the outputs of the models to better understand the cases where they fail.

For the discriminative model (SPAN-MBERT), we found that in a large number of the error cases, the model did not predict any aspect term at all. This implies that the SPAN-MBERT model was not able to confidently identify any possible aspect term spans, as it uses thresholds (representing confidence) for prediction scores. For example, in the following sentence the model fails to predict an aspect term: "Not the biggest portions but adequate.". We also found several cases where the model correctly identifies the aspect term but misclassifies the sentiment, such as for the sentence "i am never disappointed with there [sic] food."; it gives "food" a negative sentiment instead of a positive. Here, the underlying language model (mBERT) did not understand the word "never", and instead understood the sentiment from "disappointed" which has negative connotations. It has been shown that language models like BERT misunderstand some negations [54]. A significant number of errors are because the predicted and gold aspect spans only have a partial overlap. This can be seen in cases such as "La atención del personal impecable." ("The attention of the impeccable staff.") where the predicted aspect term is "personal" ("staff") instead of "atención del personal" ("attention of the staff").

As in the discriminative model, in both generative models (mBART Non-Masked and mBART Masked) we saw several cases where the predicted aspect span is only partially correct. For instance, in the sentence "Great draft and bottle selection and the pizza rocks.", the predicted entities can include "bottle selection" instead of "draft and bottle selection". We note that such predictions would not have been considered errors if we had used partial matching explained earlier in Section 4.6.

In the mBART Non-Masked model, other notable cases included those where an aspect similar to the true aspect is predicted. For example, for the sentence - "The best calamari in Seattle!", the mBART Non-Masked model generated "salmon" as an aspect term instead of "calamari". This shows that the language model understood the similarity between calamari and salmon, however it did not understand that for the task it was supposed to predict a word from the input sentence itself, and not

make such inferences. Similarly, in cross lingual experiments, we found that the model would predict the aspect term in the training language, instead of the test language. For example, when training on $\text{Rest16}_{En}$ and testing on $\text{Rest16}_{Es}$, the model tends to predict "restaurant" instead of "restaurante", "meal" instead of "comida", "service" instead of "servicio", "place" instead of "sitio", "precio" instead of "price". These are all English translations of the required Spanish aspect terms. This again implies that the model is unable to understand that it is supposed to predict a word from the input sentence itself.

## 4.9 Discussion

In our experiments, we find that in most cases, the mBART Non-Masked generative model performs better or equal to the SPAN-MBERT discriminative model. But, in 36% cases, SPAN-MBERT does better or equal to the mBART Non-Masked model.

In several cases involving $\text{MAMS}_{En}$, SPAN-MBERT performs better than the mBART Non-Masked model. This includes the monolingual in-domain setting for $\text{MAMS}_{En}$, on which generative models were not evaluated in prior work. $\text{MAMS}_{En}$ is a more difficult task, and a generative model might need more data to perform well. This can be attributed to the fact that generative models have a challenging task: learning a joint probability over all words. This is in contrast to discriminative models which need only learn a small number of decision boundaries. This intuition is supported by existing literature [55].

Based on the results and qualitative error analysis, we find that the mBART Non-Masked generative model is unable to understand that it needs to extract words from the input sentence. This leads to errors in cross-lingual and cross-lingual/domain cases. When trained with a language X, and tested with a language Y, the model generates the aspect in language X, instead of language Y.

This problem for cross-lingual and cross-lingual/domain settings especially exists for cases involving English for training/testing and Spanish for testing/training. How-

ever, the problem is less pronounced with cases involving Russian. Since the transformer model breaks words into sub-words before tokenization, English and Spanish words can be encoded the same way due to their common sub-words. Russian, on the other hand, has fewer common sub-words, making it easier for the model to distinguish between Russian and English words. This problem always exists in cases involving MAMS and Rest16 datasets together (in English and Spanish cross-lingual/domain tests). Since both these datasets also belong to the restaurant domain, it is tougher for the model to distinguish between the words from training and testing datasets. On the other hand, this is less of a problem when using Lap14 dataset along with either MAMS or Rest16 (in English and Spanish cross-lingual/domain tests) because the model is able to distinguish between the words from laptop domain and the restaurant domain. Generating out-of-sentence words is not a problem for discriminative models because they are restricted to only the words from the input sentence.

In cross-lingual and cross-domain/lingual cases, the mBART Masked model provides dramatic performance improvements over the mBART Non-Masked model (up to 20%). This is attributable to the masked model not facing the out-of-sentence aspect generation problem discussed earlier. mBART Masked model always does either better or equal to the Non-Masked model, with it being better than mBART Non-Masked in 16% cases.

## 4.10 Conclusions

In this work, we compared two ABSA model types (discriminative and generative) in terms of performance differences by considering a state-of-the-art model for each model type as a representative. We reasoned about these differences in a manner generalizable to discriminative and generative ABSA model types, and not specific to their representative models. Previous studies showed that generative models achieve higher results than the discriminative ones across almost all English ABSA datasets. However, the results in our study demonstrated that generative models can perform

worse than the discriminative ones in many of the proposed scenarios, namely, cross-lingual, cross-domain, and cross-lingual and domain. These results argue against adopting generative models as the defacto standard for all ABSA tasks as discriminative models are more accurate in some settings.

We propose a simple modification to the generative model wherein we constrain the decoding strategy. This constrained methodology often leads to significant improvements (of up to 20%) in the performance of the generative model in cases involving different train and test languages.

In the future, we plan to study the models in other scenarios like when aspects are associated with conflicting sentiment polarities / "conflict" labeled sentiment polarities. This is the condition where aspects have both positive and negative sentiment polarities.

## 4.11 Chapter Appendix

### 4.11.1 Hyperparameters

For mBART based models, the learning rates for all datasets are selected from $\{1e-3, 5e-4, 1e-4, 5e-5, 1e-5\}$ after running for three random seeds and selecting the rates giving highest Mean F1 score for their respective validation dataset.

For the SPAN-MBERT model, the logit thresholds (parameter representing confidence threshold for prediction) for all datasets are selected from $\{15, 14, 13, ..., 3, 2, 1\}$ after running for three random seeds and selecting the threshold value giving highest Mean F1 score for their respective validation dataset.

Batch size for training is taken as 16 to maximize GPU utilization. We train for 30 epochs to allow for convergence, while using an early stopping mechanism.

### 4.11.2 Training Details

For fine tuning the models, we use 1 NVIDIA V100 GPU, 6 CPU cores with 4 GB memory per core. We run training jobs with a 71 hour time limit.

# Chapter 5

# Conclusions & Future Work

Aspect Based Sentiment Analysis (ABSA) is a problem of interest to industry. ABSA can be used to detect the sentiments associated with products and services in an automated manner, thus saving on the human effort of manually scouring through potentially millions of reviews across platforms.

Even though some claim generative ABSA models are the new state-of-the-art, we find that they suffer from various limitations. We have studied generative ABSA models in various settings to find that there are cases where they frequently make errors.

Through empirical experiments, we show that generative ABSA models have a problem handling text requiring Coreference Resolution (CR) ability. We demonstrate that by fine-tuning the model on certain intermediate tasks, the model CR ability can be improved. This improved CR ability leads to a better performance on CR-containing reviews. This framework for evaluating and improving an LLM's performance on CR cases can be used for other NLP tasks apart from ABSA, as well.

By analyzing generative ABSA models in various settings like cross-domain, cross-lingual and cross-domain/lingual, we showed that even though generative ABSA models usually have a good performance, there are many cases where they are outperformed by discriminative ABSA models. Thus, contrary to prior work, discriminative ABSA models continue to be relevant for the ABSA task. We also find that generative

ABSA models frequently generate aspects which do not exist in the input sentence. To solve this, we propose a constrained decoding approach which gives a significant performance boost.

In this work, we explored a few limitations of generative ABSA models. However, since generative ABSA models were developed recently, there are several directions for future research.

For our cross-domain and cross-domain/lingual experiments, we would like to conduct experiments on other domains (apart from restaurant and laptop) and languages (apart from English, Spanish and Russian). This is because a few of our discussions were based on the similarity of certain data types (Rest and MAMS; English and Spanish). With more diverse data, we would be able to make stronger claims.

Like Coreference Resolution (CR), there can be other core NLP tasks that are also problems for generative ABSA models. These might include - sarcasm detection, named entity recognition, and part-of-speech tagging. These core NLP tasks could have an effect on the ABSA performance of generative ABSA models.

In our framework for improving a model's handling of CR-containing reviews, we propose experimenting with various intermediate-task dataset fractions for every intermediate task. However, it would be interesting if a suitable intermediate-task dataset fraction could be determined theoretically. This would save users on computation costs. Moreover, it should be investigated if using a mixture of intermediate tasks can lead to better performance compared to a single intermediate task.

It is important to comprehensively assess the limitations of generative ABSA models before deploying them in the real world.

# Bibliography


[1] H. Yan, J. Dai, T. Ji, X. Qiu, and Z. Zhang, "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)*, Online: Association for Computational Linguistics, Aug. 2021, pp. 2416–2429. DOI: 10.18653/v1/2021.acl-long.188. [Online]. Available: https://aclanthology.org/2021.acl-long.188.

[2] A. Vaswani *et al.*, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[3] W. Zhang, X. Li, Y. Deng, L. Bing, and W. Lam, "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)*, Online: Association for Computational Linguistics, Aug. 2021, pp. 504–510. DOI: 10.18653/v1/2021.acl-short.64. [Online]. Available: https://aclanthology.org/2021.acl-short.64.

[4] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman, "GLUE: A multi-task benchmark and analysis platform for natural language understanding," in *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 353–355. DOI: 10.18653/v1/W18-5446. [Online]. Available: https://aclanthology.org/W18-5446.

[5] B. Y. Lin *et al.*, "CommonGen: A constrained text generation challenge for generative commonsense reasoning," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, Online: Association for Computational Linguistics, Nov. 2020, pp. 1823–1840. DOI: 10.18653/v1/2020.findings-emnlp.165. [Online]. Available: https://aclanthology.org/2020.findings-emnlp.165.

[6] W. Medhat, A. Hassan, and H. Korashy, "Sentiment analysis algorithms and applications: A survey," *Ain Shams engineering journal*, vol. 5, no. 4, pp. 1093–1113, 2014.

[7] B. Liu, *Sentiment Analysis and Opinion Mining*. Morgan & Claypool Publishers, 2012, ISBN: 1608458849.

[8] M. Hu, Y. Peng, Z. Huang, D. Li, and Y. Lv, "Open-domain targeted sentiment analysis via span-based extraction and classification," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 537–546. DOI: 10.18653/v1/P19-1051. [Online]. Available: https://aclanthology.org/P19-1051.

[9] X. Li, L. Bing, W. Zhang, and W. Lam, "Exploiting BERT for end-to-end aspect-based sentiment analysis," in *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 34–41. DOI: 10.18653/v1/D19-5505. [Online]. Available: https://aclanthology.org/D19-5505.

[10] X. Li *et al.*, "Enhancing bert representation with context-aware embedding for aspect-based sentiment analysis," *IEEE Access*, vol. 8, pp. 46 868–46 876, 2020.

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

[12] T. Lin, Y. Wang, X. Liu, and X. Qiu, "A survey of transformers," *arXiv preprint arXiv:2106.04554*, 2021.

[13] Y. Liu *et al.*, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[14] T. Brown *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[15] C. Raffel *et al.*, "Exploring the limits of transfer learning with a unified text-to-text transformer," *arXiv preprint arXiv:1910.10683*, 2019.

[16] M. Lewis *et al.*, "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online: Association for Computational Linguistics, Jul. 2020, pp. 7871–7880. DOI: 10.18653/v1/2020.acl-main.703. [Online]. Available: https://aclanthology.org/2020.acl-main.703.

[17] M. Rezaee, K. Darvish, G. Y. Kebe, and F. Ferraro, *Discriminative and generative transformer-based models for situation entity classification*, 2021. DOI: 10.48550/ARXIV.2109.07434. [Online]. Available: https://arxiv.org/abs/2109.07434.

[18] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[19] C. Li *et al.*, "Sentiprompt: Sentiment knowledge enhanced prompt-tuning for aspect-based sentiment analysis," *arXiv preprint arXiv:2109.08306*, 2021.

[20] Y. Pruksachatkun *et al.*, "Intermediate-task transfer learning with pretrained language models: When and why does it work?" In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online: Association for Computational Linguistics, Jul. 2020, pp. 5231–5247. DOI: 10.18653/v1/2020.acl-main.467. [Online]. Available: https://aclanthology.org/2020.acl-main.467.

[21] J. Phang, T. Févry, and S. R. Bowman, "Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks," *arXiv preprint arXiv:1811.01088*, 2018.

[22] H. Xu, B. Liu, L. Shu, and P. Yu, "BERT post-training for review reading comprehension and aspect-based sentiment analysis," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 2324–2335. DOI: 10.18653/v1/N19-1242. [Online]. Available: https://aclanthology.org/N19-1242.

[23] X. Li, L. Bing, P. Li, and W. Lam, "A unified model for opinion target extraction and target sentiment prediction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019, pp. 6714–6721.

[24] M. Zhang, Y. Zhang, and D.-T. Vo, "Neural networks for open domain targeted sentiment," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal: Association for Computational Linguistics, Sep. 2015, pp. 612–621. DOI: 10.18653/v1/D15-1073. [Online]. Available: https://aclanthology.org/D15-1073.

[25] M. W. Fagerland and L. Sandvik, "Performance of five two-sample location tests for skewed distributions with unequal variances," *Contemporary clinical trials*, vol. 30, no. 5, pp. 490–496, 2009.

[26] K. K. Yuen, "The two-sample trimmed t for unequal population variances," *Biometrika*, vol. 61, no. 1, pp. 165–170, 1974.

[27] J. Pearce and B. Derrick, "Preliminary testing: The devil of statistics?" *Reinvention: An International Journal of Undergraduate Research*, vol. 12, no. 2, 2019.

[28] T. Zhang, F. Wu, A. Katiyar, K. Q. Weinberger, and Y. Artzi, "Revisiting few-sample bert fine-tuning," *arXiv preprint arXiv:2006.05987*, 2020.

[29] V. K. Ng and R. A. Cribbie, "The gamma generalized linear model, log transformation, and the robust yuen-welch test for analyzing group means with skewed and heteroscedastic data," *Communications in Statistics-Simulation and Computation*, vol. 48, no. 8, pp. 2269–2286, 2019.

[30] E. W. Weisstein, "Bonferroni correction," *https://mathworld. wolfram. com/*, 2004.

[31] M. A. Napierala, "What is the bonferroni correction?" *Aaos Now*, pp. 40–41, 2012.

[32] Q. Jiang, L. Chen, R. Xu, X. Ao, and M. Yang, "A challenge dataset and effective models for aspect-based sentiment analysis," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 6280–6285. DOI: 10.18653/v1/D19-1654. [Online]. Available: https://aclanthology.org/D19-1654.

[33] D. Tang, B. Qin, and T. Liu, "Aspect level sentiment classification with deep memory network," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 214–224. DOI: 10.18653/v1/D16-1021. [Online]. Available: https://aclanthology.org/D16-1021.

[34] Y. Tian, G. Chen, and Y. Song, "Aspect-based sentiment analysis with type-aware graph convolutional networks and layer ensemble," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Online: Association for Computational Linguistics, Jun. 2021, pp. 2910–2922. DOI: 10.18653/v1/2021.naacl-main.231. [Online]. Available: https://aclanthology.org/2021.naacl-main.231.

[35] R. Sukthanker, S. Poria, E. Cambria, and R. Thirunavukarasu, "Anaphora and coreference resolution: A review," *Information Fusion*, vol. 59, pp. 139–162, 2020.

[36] M. Pontiki *et al.*, "SemEval-2016 task 5: Aspect based sentiment analysis," in *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, San Diego, California: Association for Computational Linguistics, Jun. 2016, pp. 19–30. DOI: 10.18653/v1/S16-1002. [Online]. Available: https://aclanthology.org/S16-1002.

[37] L. Huang, R. Le Bras, C. Bhagavatula, and Y. Choi, "Cosmos QA: Machine reading comprehension with contextual commonsense reasoning," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 2391–2401. DOI: 10.18653/v1/D19-1243. [Online]. Available: https://aclanthology.org/D19-1243.

[38] Q. Liu, H. Jiang, Z.-H. Ling, X. Zhu, S. Wei, and Y. Hu, "Combing context and commonsense knowledge through neural networks for solving winograd schema problems," in *2017 AAAI Spring Symposium Series*, 2017.

[39] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuAD: 100,000+ questions for machine comprehension of text," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 2383–2392. DOI: 10.18653/v1/D16-1264. [Online]. Available: https://aclanthology.org/D16-1264.

[40] A. Wang *et al.*, "Can you tell me how to get past sesame street? sentence-level pretraining beyond language modeling," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 4465–4476. DOI: 10.18653/v1/P19-1439. [Online]. Available: https://aclanthology.org/P19-1439.

[41] N. Lourie, R. Le Bras, C. Bhagavatula, and Y. Choi, "Unicorn on rainbow: A universal commonsense reasoning model on a new multitask benchmark," *AAAI*, 2021.

[42] A. Rahman and V. Ng, "Resolving complex cases of definite pronouns: The Winograd schema challenge," in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, Jeju Island, Korea: Association for Computational Linguistics, Jul. 2012, pp. 777–789. [Online]. Available: https://aclanthology.org/D12-1071.

[43] E. Hovy, M. Marcus, M. Palmer, L. Ramshaw, and R. Weischedel, "OntoNotes: The 90% solution," in *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, New York City, USA: Association for Computational Linguistics, Jun. 2006, pp. 57–60. [Online]. Available: https://aclanthology.org/N06-2015.

[44] X. Ding and B. Liu, "Resolving object and attribute coreference in opinion mining," in *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, Beijing, China: Coling 2010 Organizing Committee, Aug. 2010, pp. 268–276. [Online]. Available: https://aclanthology.org/C10-1031.

[45] O. De Clercq and V. Hoste, "It's absolutely divine! can fine-grained sentiment analysis benefit from coreference resolution?" In *Proceedings of the Third Workshop on Computational Models of Reference, Anaphora and Coreference*, Barcelona, Spain (online): Association for Computational Linguistics, Dec. 2020, pp. 11–21. [Online]. Available: https://aclanthology.org/2020.crac-1.2.

[46] D. Mai and W. E. Zhang, "Aspect extraction using coreference resolution and unsupervised filtering," in *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing: Student Research Workshop*, Suzhou, China: Association for Computational Linguistics, Dec. 2020, pp. 124–129. [Online]. Available: https://aclanthology.org/2020.aacl-srw.18.

[47] R. Varghese and M Jayasree, "Aspect based sentiment analysis using support vector machine classifier," in *2013 international conference on advances in computing, communications and informatics (ICACCI)*, IEEE, 2013, pp. 1581–1586.

[48] J. Chen, S. Wang, S. Mazumder, and B. Liu, "A knowledge-driven approach to classifying object and attribute coreferences in opinion mining," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, Online: Association for Computational Linguistics, Nov. 2020, pp. 1616–1626. DOI: 10.18653/v1/2020.findings-emnlp.146. [Online]. Available: https://aclanthology.org/2020.findings-emnlp.146.

[49] Z. Liu, G. I. Winata, and P. Fung, "Zero-resource cross-domain named entity recognition," in *Proceedings of the 5th Workshop on Representation Learning for NLP*, Online: Association for Computational Linguistics, Jul. 2020, pp. 1–6. DOI: 10.18653/v1/2020.repl4nlp-1.1. [Online]. Available: https://aclanthology.org/2020.repl4nlp-1.1.

[50] M. Pontiki, D. Galanis, J. Pavlopoulos, H. Papageorgiou, I. Androutsopoulos, and S. Manandhar, "SemEval-2014 task 4: Aspect based sentiment analysis," in *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, Dublin, Ireland: Association for Computational Linguistics, Aug. 2014, pp. 27–35. DOI: 10.3115/v1/S14-2004. [Online]. Available: https://aclanthology.org/S14-2004.

[51] Z. Ji *et al.*, "Survey of hallucination in natural language generation," *arXiv preprint arXiv:2202.03629*, 2022.

[52] K.-H. Huang, I.-H. Hsu, P. Natarajan, K.-W. Chang, and N. Peng, "Multilingual generative language models for zero-shot cross-lingual event argument extraction," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 4633–4646. DOI: 10.18653/v1/2022.acl-long.317. [Online]. Available: https://aclanthology.org/2022.acl-long.317.

[53] N. De Cao *et al.*, "Multilingual autoregressive entity linking," *Transactions of the Association for Computational Linguistics*, vol. 10, pp. 274–290, 2022. DOI: 10.1162/tacl_a_00460. [Online]. Available: https://aclanthology.org/2022.tacl-1.16.

[54] N. Kassner and H. Schütze, "Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online: Association for Computational Linguistics, Jul. 2020, pp. 7811–7818. DOI: 10.18653/v1/2020.acl-main.698. [Online]. Available: https://aclanthology.org/2020.acl-main.698.

[55] A. Y. Ng and M. I. Jordan, "On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes," in *Advances in neural information processing systems*, 2002, pp. 841–848.