



# THE UNIVERSITY *of* EDINBURGH

This thesis has been submitted in fulfilment of the requirements for a postgraduate degree (e. g. PhD, MPhil, DClinPsychol) at the University of Edinburgh. Please note the following terms and conditions of use:

- This work is protected by copyright and other intellectual property rights, which are retained by the thesis author, unless otherwise stated.
- A copy can be downloaded for personal non-commercial research or study, without prior permission or charge.
- This thesis cannot be reproduced or quoted extensively from without first obtaining permission in writing from the author.
- The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the author.
- When referring to this work, full bibliographic details including the author, title, awarding institution and date of the thesis must be given.

---

# Low- and High-Resource Opinion Summarization

---

*Arthur Bražiņskas*



*Doctor of Philosophy*

THE UNIVERSITY OF EDINBURGH

2022

---

# Abstract

---

Customer reviews play a vital role in the online purchasing decisions we make. The reviews express user opinions that are useful for setting realistic expectations and uncovering important details about products. However, some products receive hundreds or even thousands of reviews, making them time-consuming to read. Moreover, many reviews contain uninformative content, such as irrelevant personal experiences. Automatic summarization offers an alternative – short text summaries capturing the essential information expressed in reviews. Automatically produced summaries can reflect overall or particular opinions and be tailored to user preferences. Besides being presented on major e-commerce platforms, home assistants can also vocalize them. This approach can improve user satisfaction by assisting in making faster and better decisions.

Modern summarization approaches are based on neural networks, often requiring thousands of annotated samples for training. However, human-written summaries for products are expensive to produce because annotators need to read many reviews. This has led to annotated data scarcity where only a few datasets are available. Data scarcity is the central theme of our works, and we propose a number of approaches to alleviate the problem. The thesis consists of two parts where we discuss low- and high-resource data settings.

In the first part, we propose self-supervised learning methods applied to customer reviews and few-shot methods for learning from small annotated datasets. Customer reviews without summaries are available in large quantities, contain a breadth of in-domain specifics, and provide a powerful training signal. We show that reviews can be used for learning summarizers via a self-supervised objective. Further, we address two main challenges associated with learning from small annotated datasets. First, large models rapidly overfit on small datasets leading to poor generalization. Second, it is not possible to learn a wide range of in-domain specifics (e.g., product aspects and usage) from a handful of gold samples. This leads to subtle semantic mistakes in generated summaries, such as ‘*great dead on arrival battery*.’ We address the first challenge by explicitly modeling summary properties (e.g., content coverage and sentiment alignment). Furthermore, we leverage small modules – *adapters* – that are more robust to overfitting. As we show, despite their size, these modules can be used to store in-domain knowledge to reduce semantic mistakes. Lastly, we propose a simple method for learning personalized summarizers based on aspects, such as ‘price,’ ‘battery life,’ and ‘resolution.’ This task is harder to learn, and we present a few-shot method for training a query-based summarizer on small annotated datasets.

In the second part, we focus on the high-resource setting and present a large dataset with summaries collected from various online resources. The dataset has more than 33,000 human-written summaries, where each is linked up to thousands of reviews. This, however, makes it challenging to apply an ‘expensive’ deep encoder due to memory and computational costs. To address this problem, we propose selecting small subsets of informative reviews. Only these subsets are encoded by the deep encoder and subsequently summarized. We show that the selector and summarizer can be trained end-to-end via amortized inference and policy gradient methods.

---

# Declaration

---

This thesis has been submitted in fulfilment of the requirements for a postgraduate degree (e.g. PhD, MPhil, DClinPsychol) at the University of Edinburgh. Please note the following terms and conditions of use:

This work is protected by copyright and other intellectual property rights, which are retained by the thesis author, unless otherwise stated. A copy can be downloaded for personal non-commercial research or study, without prior permission or charge. This thesis cannot be reproduced or quoted extensively from without first obtaining permission in writing from the author. The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the author. When referring to this work, full bibliographic details including the author, title, awarding institution and date of the thesis must be given.

---

# Acknowledgements

---

I express endless gratitude to Ivan Titov for being my main supervisor. Our weekly meetings significantly deepened my understanding of machine learning and natural language processing. I was very fortunate to receive so much care and attention. Further, I would like to thank Mirella Lapata – my second supervisor – for her vision in shaping projects, encouragement, and wisdom. Finally, Wilker Aziz for discussions, suggestions, and paper draft reviews.

My journey would not be the same without two key figures. Their contributions made it possible to overcome doubts, objective and subjective obstacles, and reach my goals. The first one is my grandmother – Vera Golubeva. I will always cherish her unconditional emotional support, care, and kindness. Second, Marina Vashchenko for her love, endless faith in me, and help on my journey, especially during low periods.

I thank all my family members and friends for your love and contributions. Yangpeng Zhang for amazing hiking trips, dinners, and technical and non-technical conversations; Prof. Ramon Grima and Dr. Nina Kudryashova for interesting intellectual conversations; Lena Voita for helping me in self-realization. Finally, to my friends with whom I shared many great moments: Nat Tantakasem, William Berg, and Aya Siddig Gamil.

When I was working on my first project, I received a lot of valuable advice and suggestions from Jonathan Mallinson, Serhii Havrylov, Simao Eduardo, and Stefanos Angelidis. I'm grateful for their help in progressing quickly. Furthermore, my internships at Amazon taught me many valuable lessons that I will apply in my future career and life. For these experiences, I would like to thank Markus Dreyer, Mohit Bansal, Ramesh Nallapati, Jonathan Pilaul, and Leonardo Ribeiro.

I was fortunate to share my office with amazing Ph.D. students – Nikos Mavrogeorgis and Mohammad Vaziri. I will never forget our amazing office environment and conversations. Also, I would like to thank my dear friends Romans Seredjuks and Vika Kuznetsova for your hospitality, warmth, and satsang. Finally, I was fortunate to be around many highly talented colleagues: Reinald Amplayo, Yang Liu, Biao Zhang, Karim Manaouil, Mohammed Hosseini, Yumo Xu, Nicola De Cao, Dmitrii Ustiugov, Xinchu Chen, William Berg, and Michael Sejr Schlichtkrull.

---

# Contents

---

<b>Abstract</b>	<b>ii</b>
<b>Declaration</b>	<b>iv</b>
<b>Acknowledgements</b>	<b>v</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Applications . . . . .	2
1.2 Challenges . . . . .	3
1.3 Thesis Statement . . . . .	6
1.4 Thesis Overview . . . . .	7
1.5 Published Works . . . . .	11
<b>2 Background: Machine Learning</b>	<b>12</b>
2.1 Probabilistic Models . . . . .	12
2.2 Inference . . . . .	14
2.3 Model Estimation . . . . .	14
2.4 Auto-encoding . . . . .	16
2.5 Variational Auto-encoding . . . . .	17
2.6 Language Models . . . . .	21
<b>3 Background: Summarization</b>	<b>22</b>
3.1 News Summarization . . . . .	22
3.2 History of Opinion Summarization . . . . .	23
3.3 Automatic Evaluation . . . . .	29
3.4 Human Evaluation . . . . .	30
<b>I Low-Resource Opinion Summarization</b>	<b>32</b>
<b>4 Unsupervised Opinion Summarization as Copycat-Review Generation</b>	<b>33</b>
4.1 Introduction . . . . .	33
4.2 Related Work . . . . .	36
4.3 Model Description . . . . .	36
4.4 Design . . . . .	40
4.5 Summary Generation . . . . .	42
4.6 Experimental Setup . . . . .	42

<b>CONTENTS</b>	<b>vii</b>
4.7 Human Evaluation . . . . .	45
4.8 Analysis . . . . .	47
4.9 Conclusions . . . . .	50
4.10 Reflections . . . . .	50
<b>5 Few-Shot Learning for Opinion Summarization</b>	<b>54</b>
5.1 Introduction . . . . .	55
5.2 Related Work . . . . .	57
5.3 Unsupervised Training . . . . .	57
5.4 Novelty Reduction . . . . .	59
5.5 Summary Adaptation . . . . .	59
5.6 Experimental Setup . . . . .	61
5.7 Evaluation Results . . . . .	64
5.8 Analysis . . . . .	65
5.9 Conclusions . . . . .	69
5.10 Reflections . . . . .	70
<b>6 Efficient Few-Shot Fine-Tuning for Opinion Summarization</b>	<b>73</b>
6.1 Introduction . . . . .	74
6.2 Approach . . . . .	76
6.3 Experimental Setup . . . . .	79
6.4 Results . . . . .	81
6.5 Analysis . . . . .	84
6.6 Related Work . . . . .	86
6.7 Conclusions . . . . .	87
6.8 Future Work . . . . .	87
<b>II High-Resource Opinion Summarization</b>	<b>92</b>
<b>7 Learning Opinion Summarizers by Selecting Informative Reviews</b>	<b>93</b>
7.1 Introduction . . . . .	93
7.2 Dataset . . . . .	95
7.3 Approach . . . . .	97
7.4 Experimental Setup . . . . .	100
7.5 Results . . . . .	103
7.6 Analysis . . . . .	106
7.7 Related Work . . . . .	107
7.8 Conclusions . . . . .	108
7.9 Future Work . . . . .	109



<b>CONTENTS</b>	<b>viii</b>
<b>8 Conclusions and Future Work</b>	<b>113</b>
8.1 Conclusions . . . . .	113
8.2 Future Work . . . . .	115
<b>A Unsupervised Summarization</b>	<b>118</b>
A.1 Human Evaluation Setup . . . . .	118
A.2 Full Human Evaluation Instructions . . . . .	118
A.3 Amazon Summaries Creation . . . . .	119
A.4 Human Interface Examples . . . . .	119
<b>B Few-shot Learning for Opinion Summarization</b>	<b>121</b>
B.1 Best-Worst Scaling Details . . . . .	121
B.2 Human Evaluation Setup . . . . .	121
B.3 Summary Annotation . . . . .	122
<b>C Efficient Few-shot Fine-tuning</b>	<b>123</b>
C.1 Best-Worst Scaling Details . . . . .	123
C.2 Human Evaluation Setup . . . . .	123
<b>D Learning Opinion Summarizers by Selecting Informative Reviews</b>	<b>124</b>
D.1 REINFORCE vs Gumbel-Softmax . . . . .	124
D.2 Human Evaluation Setup . . . . .	124
<b>Bibliography</b>	<b>125</b>

---

## Chapter 1

# Introduction

---

Online reviews play an important role in purchasing decisions we make. They inform us about customer experiences – what aspects users like and dislike – and, ultimately, whether a product<sup>1</sup> or service is worth purchasing. While most e-commerce platforms make customer reviews publicly accessible, their direct utilization for decision making has a number of challenges. We can categorize these challenges as **volume**, **the absence of structure**, and **uninformative content**.

First, products and services often have hundreds or even thousands of reviews. And while platforms like Amazon present reviews ranked by helpfulness,<sup>2</sup> it is rarely enough to read the top ones to get a sufficient understanding of overall opinions.

Second, reviews are often written without an explicit structure. For instance, a user might be interested in the sound quality of a speaker. However, there is no simple way to retrieve all reviews discussing this particular aspect.

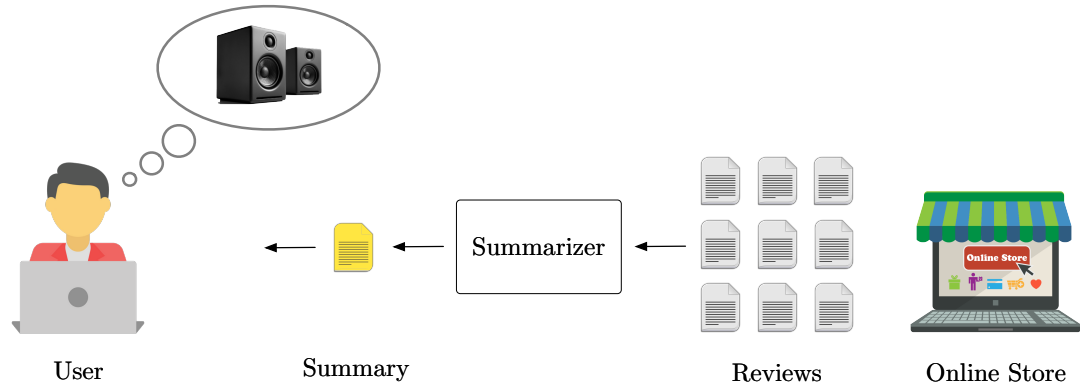
Third and finally, in reviews, informative and uninformative content is often mixed. For example, informative content could be facts and opinions about aspects, while uninformative content could be rare personal experiences. Consequently, a user spends time reading irrelevant parts of the review. This is further exacerbated when looking for information about a particular aspect (e.g., operation speed).

In this light, automatic summarization can be helpful. It acts as an intermediary between a volume of unstructured reviews and the user. A summarizer inputs a large number of reviews and produces a short text summary, as shown in Fig. 1.1.

---

1. For simplicity, we often refer to both products (e.g., iPhone X) and businesses (e.g., a specific Starbucks branch) as *products*.

2. Helpfulness is a score that is calculated based on user votes.



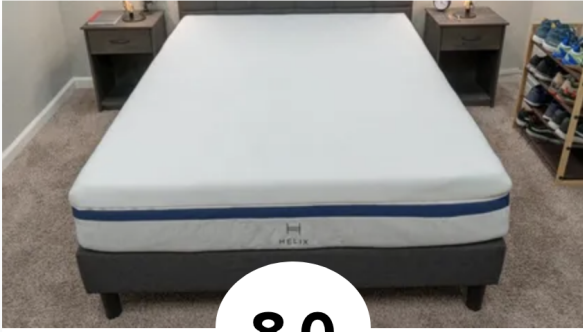
**Figure 1.1:** The summarization process of customer reviews.

Opinion summarization has been historically approached using *extractive methods* due to limited modelling capacity and the absence of large annotated datasets; we will discuss this further in Sec. 3.2. These methods select a small subset of sentences from customer reviews to form a summary. However, such summaries often have incoherent sentences and do not provide a good overview of opinion variations. Unlike extractive summarizers, abstractive summarizers paraphrase, abstract, and deal with conflicting opinions. We will focus on abstractive models in the thesis.

## 1.1 Applications

From the practical perspective, opinion summarization is particularly useful for e-commerce platforms. Automatically produced summaries can help users make faster and better purchasing decisions and thus improve their satisfaction. Generally, e-commerce platforms have become an integral part of our daily lives. Moreover, their popularity has increased significantly due to the global pandemic in recent years. However, such platforms rarely provide the digests of customer reviews or ways to navigate over actionable information. Meanwhile, the demand for opinion summaries is evident from the existence of various independent websites, such as [www.cnet.com](http://www.cnet.com) and [www.bestreviews.com](http://www.bestreviews.com). These websites offer professional product reviews along with summaries, see an example in Fig. 1.2.

Summaries on such websites are written by professional critics who read customer opinions on major e-commerce platforms, test products themselves, and consult consumers. However, this process is time-consuming and inefficient for covering the ‘long tail’ of less popular products. Machine learning can automate the process of summary writing and cover a large volume of products. Automatically generated summaries could be presented on major e-commerce platforms and independent websites.



**8.0**

**Helix**

**\$599 AT HELIX SLEEP**

**LIKE**

- 👍 All firmness levels from soft to firm to suit all sleeping positions
- 👍 It has a sleep quiz to help you choose the right bed
- 👍 Hybrid construction accommodates all body types
- 👍 Helix Plus option for those with plus-size body types
- 👍 Fair price for the base Helix models

**DON'T LIKE**

- 👎 Helix Luxe models are a little pricey
- 👎 No foam version of the hybrid models

**Figure 1.2:** Pros and cons about a mattress on [www.cnet.com](http://www.cnet.com).

More recently, home assistants, such as ALEXA and GOOGLE ASSISTANT, have gained significance popularity. Users vocally interact with these systems by requesting information. When a user asks for an opinion about a product, an assistant could vocalize a short summary. The summary could be personalized, context-dependent, and tailored to the request.

## 1.2 Challenges

Abstractive summarization of customer reviews is a recently emerging task. Due to its recency, it comes with a number of open problems. These open problems are exciting avenues both from scientific and industrial perspectives.

### 1.2.1 Modelling Multiple Reviews

Each product and business can have thousands of reviews to summarize. This is especially true for popular electronic products. For example, in 2022, iPhone X has 17,420 customer reviews on [www.amazon.com](http://www.amazon.com). Even if we had human-written summaries for such products, we would face a modelling issue of how to represent so many reviews. In order to learn powerful features representing input reviews, one needs to utilize a deep encoder, such as a Transformer (Vaswani et al., 2017). However, for a Transformer-based encoder with the self-attention module, the computational and memory requirements grow quadratically with sequence length (Beltagy, Peters, & Cohan, 2020). This makes it infeasible (or very expensive) to perform multi-review summarization in realistic settings. Previous work on abstractive opinion

summarization – MEANSUM (Chu & Liu, 2019) – side-steps this problem by considering only eight reviews as input and condensing each one to a dense feature vector. However, a single dense vector is not optimal for representing information about the entire review (Bahdanau, Cho, & Bengio, 2015).

### 1.2.2 Annotated Data

Supervised models often rely on large collections of annotated data for training. In related domains, such as news summarization, datasets with thousands of human-written summaries are available (Fabbri, Li, She, Li, & Radev, 2019; Hermann et al., 2015; Narayan, Cohen, & Lapata, 2018). In opinion summarization, however, annotated datasets where summaries are paired with reviews are scarce. Such datasets are exclusively created via human efforts.<sup>3</sup> In turn, human efforts are expensive because annotators need to read many reviews to write a summary. Unsurprisingly perhaps, datasets that are available, such as the ones introduced in Angelidis and Lapata (2018); Chu and Liu (2019), contain only a few hundred summaries. Moreover, these summaries are based on a maximum of ten reviews, while real products can have thousands, see Sec. 1.2.1. Fortunately, customer reviews without summaries are available in large quantities. For example, in 2022, Amazon has approximately 250 millions of reviews. This makes it possible to train summarizers without annotated data.

### 1.2.3 Input Faithfulness

For practical applications, summarizers should produce output texts with contents accurately reflecting information in input texts. However, modern summarizers are prone to hallucinations (Maynez, Narayan, Bohnet, & McDonald, 2020), i.e., generation of content unfaithful to input. For example, a summarizer might mistakenly generate the word ‘iPhone’ while it is never mentioned in the reviews of iPad. This, in turn, can lead to user aversion and the loss of trust.

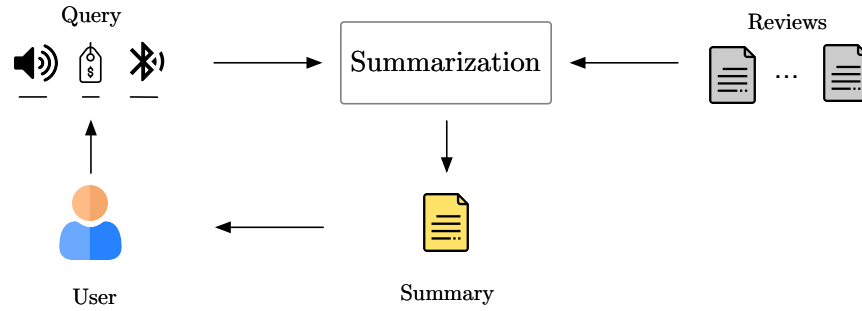
### 1.2.4 Summary Evaluation

In order to develop and compare models, one needs a reliable metric indicating the quality of generated outputs. The most common metric in summarization is ROUGE (Lin, 2004). It is based on n-gram overlap between generated and human-written summaries. We will discuss it in detail in Sec. 3.3. While this metric is very convenient and easy to apply, it has a number of limitations. First of all, the metric operates on the lexical level and often poorly captures semantics. This is particularly problematic for highly abstractive summaries. For example, consider the gold summary: *‘The gelato was delicious, would highly recommend it.’*, and two generated summaries below.

1) *This place has the best ice-cream.*

---

3. To the best of our knowledge, our work in Chapter 7 is the only exception.



**Figure 1.3:** Illustration of personalized summarization. Here, the user submits a query that is passed along with reviews to the summarizer. The summarizer generates a query-based summary.

2) *The coffee was tasteless, would not recommend it.*

While it is obvious that the first summary would be preferred by humans, their ROUGE-L scores contradict this intuition and are 13.33 and 62.50 points. This simple example is supported by a number of studies indicating that ROUGE can be insensitive to flipped sentiment (Tay, Joshi, Zhang, Karimi, & Wan, 2019) and fluency (Paulus, Xiong, & Socher, 2017). Moreover, we naturally expect generated summaries to only contain information supported by input reviews. However, this is an open problem, as we discussed in Sec. 1.2.3. Furthermore, neither ROUGE nor alternative automatic metrics highly correlate with input faithfulness (Fabbri et al., 2021). This makes automatic evaluation results less reliable and hinders model development.

This problem is often alleviated by performing human evaluation studies. Here, hired workers are instructed to evaluate summaries based on various criteria, such as fluency, informativeness, and input fidelity. If such a study is designed well, it provides reliable indicators for output quality. However, it is expensive and requires an extra effort for training and instructing workers. This can make it impractical to use in development cycles and is often performed at the very end.

### 1.2.5 Personalization

Online users often have particular criteria for choosing products. For example, if a user is looking for a wireless headset, the criteria might be ‘*sound quality*’ and ‘*price*.’ Therefore, a *generic summary*, condensing opinions on a wide range of aspects, will unlikely satisfy the request. We can model this scenario as follows. A user submits a text query with multiple aspect keywords, such as ‘*price*’ and ‘*resolution*.’ In turn, a summarizer takes as input a reviews-query pair and summarizes only opinions about these aspects. We illustrate the process in Fig. 1.3.

Summary personalization via aspect queries is useful in practice yet is more challenging for learning as a task than generic summarization. First, the model needs to rely on the query and generate summaries only about particular aspects. Second, user queries can be very diverse, containing both fine-grained and coarse-grained aspects. Consequently, a model would need large annotated resources to yield high quality summaries. However, such datasets are not readily available and are even more expensive to produce than generic ones. This, in turn, calls for creative methods to learn personalized summarizers.

### 1.3 Thesis Statement

In this thesis, we investigate a series of hypotheses related to abstractive summarization of customer reviews. We test these hypotheses through extensive evaluation and analysis of our methods.

*Hypothesis I: Customer reviews without human-written summaries provide a sufficiently strong signal to train an abstractive opinion summarizer that generates fluent, coherent, and input faithful summaries.*

As human-written summaries are scarce and customer reviews are vast, learning summarizers from the latter is a significant step towards the real-world applications of opinion summarization technology. We explored this direction and demonstrated the effectiveness of the unsupervised approach.

*Hypothesis II: A handful of human-written summaries is sufficient to learn key summary characteristics and subsequently improve writing style and informativeness of generated summaries.*

The ability to improve an unsupervised opinion summarizer with a handful of summaries opens exciting avenues for low-budget business applications. We demonstrated that substantial improvements can be achieved even in extremely low-resource settings.

*Hypothesis III: In-domain knowledge can be efficiently stored in small neural modules and subsequently leveraged to generate summaries with accurate product specifics.*

Large language models, pre-trained on generic texts, are rarely accustomed to product specifics. After a naive few-shot fine-tuning, this results in subtle semantic mistakes in generated summaries. For example, *This **hair dryer** is great for **water cooling**.* We show that in-domain knowledge can be effectively stored in small neural modules inserted into the language model. This approach combines powerful language generation and understanding abilities acquired on generic texts with in-domain knowledge indispensable for generating summaries with accurate product specifics.

*Hypothesis IV: High-quality and large scale abstractive summarization dataset can be created from online resources.*

Large annotated resources are vital for progress in the field. While the manual summarization of reviews can be very expensive, we demonstrated that a dataset with summaries could be collected from online resources. The dataset is many times larger than the existing ones and resembles the real-world setting where a product can have thousands of reviews.

*Hypothesis V: Summarization can be done efficiently by learning to select the subsets of informative reviews and summarizing only the content in these reviews.*

The created dataset calls for computationally- and memory-efficient methods to summarize thousands of reviews. We achieved this by training a review selector jointly with a summarizer. The selector learns to select informative review subsets from large review collections that the model subsequently summarizes.

## 1.4 Thesis Overview

In this section, we provide a broad overview of the two thesis parts. Namely, low-resource and high-resource opinion summarization. In the former, we discuss unsupervised and few-shot learning. Here, an unsupervised model is trained solely on customer reviews, which are available in large quantities. In turn, few-shot models are pre-trained on customer reviews to learn in-domain specifics and then fine-tuned on a handful of human-written summaries (less than 100). Also, we focus on a simplified scenario where we summarize a small number of reviews (8). This makes it feasible to produce human-written summaries and perform modelling (see Sec. 1.2.1 and Sec. 1.2.2). However, in reality, a product can have hundreds or thousands of reviews. So, in the second part, we present a large dataset where summaries have up to a couple of thousands of associated reviews. As we explained in Sec. 1.2.1, summarizing so many reviews is challenging. Therefore, we propose to select informative review subsets with a trainable selector. Subsequently, only these small review subsets are summarized.

### 1.4.1 Unsupervised Opinion Summarization

In Chapter 4, we validate the first hypothesis and focus on unsupervised learning, where no annotated datasets are available for training. Here, we introduce the *leave-one-out* unsupervised objective to train a summarizer. Also, we will revisit this objective in Chapters 5 and 6 to pre-train models before few-shot fine-tuning. Under this objective, a model learns to predict one review when the other reviews of a product are passed as input. In this way, the model learns to generate product-related texts by leveraging correlations between reviews. For instance, if the nine reviews mention the price being too high, it provides a strong signal for predicting that the tenth review will also express a negative sentiment toward the price.



Review 1	Review 2
When I first got diabetes I got this. It has a lot of what we need. But later I have switched to another brand.	These capsules are natural alternative to other over-the-counter medications. They are easy to swallow and have a great taste. Overall, great value for money.

**Table 1.1:** Two example reviews where the first one is a more typical review while the second one has more summary-like text characteristics.

Our first model, called COPYCAT, where both individual review and product representations are continuous latent variables. The product representations can store, for example, overall sentiment, common topics, and opinions expressed about the product. In turn, the latent representations of reviews depend on the product representations and capture the content of individual reviews. In this chapter, we show how these representation variables can be used to generate consensus summaries expressing common opinions. We show that our model generates more input faithful summaries than state-of-the-art MEANSUM. In addition, it outperforms MEANSUM by 1.75 ROUGE-L points on the Amazon test set.

#### 1.4.2 Few-shot Opinion Summarization

As COPYCAT is trained in the unsupervised regime on customer reviews solely, it sometimes generates text patterns that are review-like. We provide an example below where such patterns are in **bold**.

*These are the best tights **I've ever worn**. They fit well and are comfortable to wear. **I wish they were** a little bit thicker, but **I'm sure** they will last a long time.*

The highlighted fragments are written in the informal writing style and are not informative. Such fragments are not appropriate in summaries. To address these issues, in Chapter 5, we introduce a few-shot model – FEWSUM. The central idea is to model differences between summaries and customer reviews. In contrast to summaries, we observe that customer reviews are often written in the informal writing style and contain information not present in other reviews (e.g., personal experiences). We provide two example reviews in Table 1.1 that vary in terms of their characteristics.

We refer to these characteristics as *properties* (Ficler & Goldberg, 2017), which are computed automatically, without requiring human annotation.<sup>4</sup> These properties are passed as continuous vectors to a model and indicate how summary-like the target sequence is. Further, we train a model in two steps – unsupervised pre-training on customer reviews and few-shot fine-tuning on a handful of summaries. Specifically, we pre-train the model via the leave-one-out objective and thus expose it to a large variety of property values. Some of the property values correspond

4. A convenient way of thinking about properties is random variables that have particular assignments/values.

to more summary-like reviews and some to less. Further, we fine-tune a tiny *plug-in* network on a handful of summaries to find what property values lead to summaries. Finally, we use the plug-in network in test time to generate summaries. This approach validates *Hypothesis II* in Sec. 1.3. We show in extensive human evaluation studies that FEWSUM-generated summaries are significantly more informative and generally preferred by humans. Additionally, we improve the ROUGE-L score by 2.63 points over COPYCAT on the Amazon test set.

### 1.4.3 Efficient Few-shot Opinion Summarization

In Chapter 6, we focus on pre-trained language models (PLMs) and their efficient utilization for opinion summarization. PLMs were shown to be useful in a large number of NLP domains. However, when billions of parameters are optimized on a handful of summaries, it leads to rapid overfitting and consequently to poor summaries. To alleviate this issue, we only optimize small neural modules (0.6-5% of LM's parameters) injected into Transformer layers, called *adapters* (Houlsby et al., 2019). We refer to this summarizer as ADASUM, and it validates *Hypothesis III*.

PLMs are pre-trained on generic corpora that rarely matches in-domain data, such as customer reviews. This makes them less accustomed to a large variety of product specifics. On the one hand, after the naive fine-tuning on gold samples, they can generate summaries with subtle semantic mistakes. On the other hand, further training of the entire PLM on in-domain data is storage and memory inefficient (Mahabadi, Henderson, & Ruder, 2021), also leading to catastrophic forgetting. Instead, we propose to pre-train adapters to store in-domain and task-oriented knowledge using the leave-one-out objective.

Well-organized content in summaries is important for the user as it is easier to follow. However, the lack of annotated data makes it challenging to learn the desired content structure. To address this issue, we leverage *text planning* (Hua & Wang, 2019; Moryossef, Goldberg, & Dagan, 2019a). Specifically, we introduce ADAQSUM, that inputs intermediate summary representation in the form of a text query consisting of aspect keywords. As we show, this results in more coherent text patterns with fewer redundancies. Moreover, it can be useful for personalized summaries, better reflecting user interests.

All in all, our adapter-based summarizers outperform all few-shot alternatives by a large margin in ROUGE scores. For instance, ADAQSUM achieves an impressive 4.57 ROUGE-L points improvement over FEWSUM on Amazon.

	Ent	Rev/Ent	Summs
AMASUM (Chapter 7)	31,483	326	33,324
SPACE (Angelidis, Amplayo, Suhara, Wang, & Lapata, 2020)	50	100	1,050
COPYCAT (Chapter 4)	60	8	180
FEWSUM (Chapter 5)	60	8	180
MEANSUM (Chu & Liu, 2019)	200	8	200
OPOSUM (Angelidis & Lapata, 2018)	60	10	180

**Table 1.2:** Statistics comparing our dataset to alternatives. For AMASUM, we show the average number of reviews and references per entity.

#### 1.4.4 Informative Review Selection and Summarization

In Chapters 4, 5, and 6, we operate in a simplified scenario where eight reviews are summarized. In practice, however, a product might have hundreds or even thousands of reviews. In Chapter 7, we introduce AMASUM – the largest opinion summarization dataset with more than 33,000 human-written summaries. Each summary is accompanied by up to 2,300 reviews. These summaries were extracted from popular platforms, such as [www.bestreviews.com](http://www.bestreviews.com). And they were written by professional product reviewers that utilize Amazon customer reviews as their main source of information. The proposed dataset validates *Hypothesis IV* and opens new avenues for research and is substantially larger than the available alternatives, see Table 1.2.

Summaries in the dataset are often linked to a large number of reviews. In turn, it is computationally challenging to encode and attend a large number of reviews using the standard Transformer encoder-decoder model. Also, random review subsets might not cover well the content of summaries. As we show, this naive approach to reducing the number of reviews is sub-optimal and results in hallucinations in test time. In this light, we propose to train a review selector along with the summarizer. Essentially, the selector is learning to select reviews that allow the summarizer to accurately predict summaries. By design, the selector relies on simple features, thus scaling the system to large collections of reviews. This system, which is called SELSUM, is trained end-to-end by leveraging amortized variational inference (Kingma & Welling, 2013) and policy gradient methods. This approach, combining the selector and summarizer, validates *Hypothesis V*. We demonstrate that SELSUM improves ROUGE scores over alternatives and results in more input faithful summaries.

#### 1.4.5 Human Evaluation

As we mentioned in Sec. 1.2.4, hallucinations in generated summaries impair practical applications. To assess how input faithful generated summaries are, we propose a human evaluation study. Here, we leverage the Amazon Mechanical Turk crowd-sourcing platform to hire workers to conduct the evaluation. In the evaluation, workers are asked to assess the input faithfulness of each generated sentence with respect to input reviews. Since the workers are well-qualified and pass qualification tests, they help us to get an accurate picture to

compare different abstractive models. We also perform another human study type, called *Best-Worst Scaling* (Kiritchenko & Mohammad, 2016b; Louviere, Flynn, & Marley, 2015; Louviere & Woodworth, 1991). Here we hire workers and ask them to select the best and worst summary for each criterion, such as fluency, coherence, and sentiment alignment. This evaluation allows us to capture the human preference for summaries produced by different systems. These evaluation methods are first introduced in Chapter 4 but are used in Chapters 5, 6, and 7.

## 1.5 Published Works

Chapter 4 is based on Bražinskas, Lapata, and Titov (2020b), published at ACL. Chapter 5 is based on Bražinskas, Lapata, and Titov (2020a), published at EMNLP. Chapter 6 is based on Bražinskas, Nallapati, Bansal, and Dreyer (2022), published at NAACL. Chapter 7 is based on Bražinskas, Lapata, and Titov (2021), published at EMNLP.

# Background: Machine Learning

---

In this chapter we will discuss essential theories and methods for a better understanding of the works. In the beginning, we focus on probabilistic models, inference, and their estimation. Further, we discuss representation learning via auto-encoders and variational auto-encoders. While we provide present these methods in detail, we assume a familiarity with machine learning as expressed in Bishop (2006) and I. Goodfellow, Bengio, and Courville (2016).

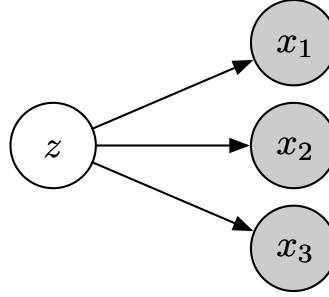
## 2.1 Probabilistic Models

In the thesis, we use probabilistic models. These models have solid roots in probability and information theory, and allow us to capture a model's uncertainty. The uncertainty can be associated with the model's output and latent structures realized as outputs. Each entity in a problem is represented as a *random variable* that have a particular range of assignments – *support*. For instance, in a clinical setting, we might have access to the symptoms of various patents. Further, we might be interested in knowing the disease responsible for these symptoms. To simplify the problem, let's assume that a disease  $z$  causes three symptoms:  $x_1$ ,  $x_2$ , and  $x_3$ . The important distinction between these two types of variables is that we can observe the former ones in data while we cannot observe the latter one. Further, our interest is to *infer* the distribution over possible diseases  $z$  when we observe the symptoms  $x_1$ ,  $x_2$ , and  $x_3$ . We will discuss inference in detail in Sec. 2.2.

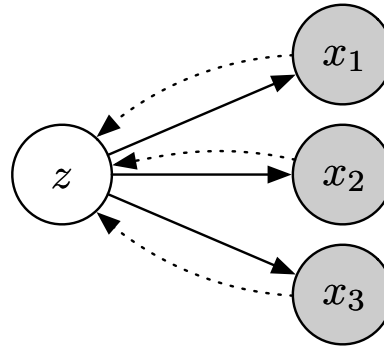
A convenient way to represent our random variables and their dependencies is using the *graphical notation* (Koller & Friedman, 2009). This visualization provides an easy-to-follow overview to the reader. In the thesis, we exclusively represent our models as directed acyclic graphs – *Bayesian networks*. We visualize our previously described model in Fig. 2.1. Conventionally, shaded and unshaded circles indicate observed and unobserved (*latent*) variables, respectively. And the directed edges connecting  $z$  and  $x_1$ ,  $x_2$ , and  $x_3$  indicate dependencies. Often, these edges are perceived to indicate **causal** relationships, which is not necessarily true.<sup>1</sup> Further, we can factorize the joint distribution as shown in Eq. B.1.

---

1. For more details, please refer to Pearl and Mackenzie (2018).



**Figure 2.1:** A graphical representation of a disease ( $z$ ) responsible for three symptoms:  $x_1$ ,  $x_2$ ,  $x_3$ . The shaded and unshaded circles indicate observed and unobserved variables, respectively.



**Figure 2.2:** A graphical representation of a disease ( $z$ ) responsible for three symptoms:  $x_1$ ,  $x_2$ ,  $x_3$ . The dotted lines indicate inference.

$$p(x_1, x_2, x_3, z) = p(x_1, x_2, x_3 | z) p(z) = p(x_1 | z) p(x_2 | z) p(x_3 | z) p(z) \quad (\text{B.1})$$

Here we apply the chain rule and decompose the joint distribution to the *conditional likelihoods* –  $p(x_1 | z)$ ,  $p(x_2 | z)$ ,  $p(x_3 | z)$  – and the *prior* –  $p(z)$ . Intuitively, these conditional likelihoods suggest that our knowledge of the disease  $z$  affects our *belief* about symptoms a patient might experience. The prior  $p(z)$  captures our assumptions about how likely various diseases are. The prior can be set based on previous experiments or assumptions, see Gelman and Hill (2006) for more details. Another property that stems from this framing is called *conditional independence*. Specifically, when we condition on  $z$ , each  $x_i$  becomes independent of each other thus leading to the factorization in Eq. B.1. In turn, this substantially simplifies computations as our distributions become simpler. Back in the day, this factorization was at the core of a popular classification model called *Naive Bayes* (Bishop, 2006).

## 2.2 Inference

In general, inference refers to the reversal of the relationship between random variables. Here, we are interested in the associated distribution – *posterior*. If this distribution is available, we can compute density<sup>2</sup> for the latent variable assignments by conditioning on observed variables. We illustrate the reversal by the dotted lines in Fig. 2.2. Finally, the posterior is computed by leveraging the *Bayes rule*, as shown in Eq. B.2.

$$p(z|x_1, x_2, x_3) = \frac{p(z, x_1, x_2, x_3)}{p(x_1, x_2, x_3)} = \frac{p(x_1, x_2, x_3|z)p(z)}{p(x_1, x_2, x_3)} = \frac{p(x_1, x_2, x_3|z)p(z)}{\int p(x_1, x_2, x_3|z)p(z)dz} \quad (\text{B.2})$$

Our latent codes  $z$  are often continuous and data  $x$  are discrete. A popular choice is to assume that  $z$  follows a Gaussian distribution while  $x$  Categorical. We will explore this in the COPYCAT model presented in Chapter 4. Under this assumption, however, the exact computation of this distribution is not feasible. The reason is that the denominator requires us to integrate over all the possible assignments to  $z$ . In Sec. 2.5, we will discuss how to approximate this distribution.

## 2.3 Model Estimation

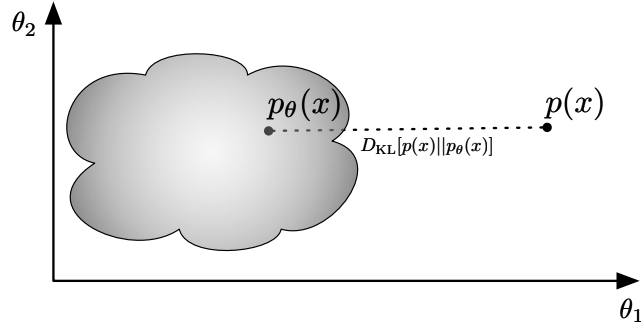
For now, we will focus on a model without latent variables –  $p_\theta(x)$ . As is common in NLP, we will assume that  $x$  follows a Categorical distribution. Unlike the example in Sec. 2.1, this model has free parameters  $\theta \in \Theta$ , which we can select from the space  $\Theta$  to fit a particular dataset. The problem of searching for parameters  $\theta$  is known as *estimation*. And we follow the *maximum likelihood estimation* principle under the lenses of *information theory* (MacKay, 2003). In essence, we assume the existence of true data distribution  $p(x)$ . In turn, data points in our dataset are samples from this distribution. Further, we aim to approximate this distribution using the parametrized model  $p_\theta(x)$ . This can be formalized as a search problem over the parameter space  $\Theta$  in Eq. B.3.

$$\theta^* = \arg \min_{\theta \in \Theta} \mathbb{D}_{\text{KL}} [p(x) || p_\theta(x)] \quad (\text{B.3})$$

The Kullback-Leibler divergence term ( $\mathbb{D}_{\text{KL}} [p(x) || p_\theta(x)]$ ) measures the distance between two distributions in *nats*. And the goal is to find the *estimate*  $\theta^*$  that minimizes the distance. We illustrate this search problem in Fig. 2.3. By leveraging the property of logarithms and expectations, the term can be further decomposed as shown in Eq. B.4.

$$\mathbb{D}_{\text{KL}} [p(x) || p_\theta(x)] = \mathbb{E}_{x \sim p(x)} \left[ \log \frac{p(x)}{p_\theta(x)} \right] = \mathbb{E}_{x \sim p(x)} [\log p(x)] - \mathbb{E}_{x \sim p(x)} [\log p_\theta(x)] \quad (\text{B.4})$$

2. Or probability mass if the variable is discrete.



**Figure 2.3:** The parameter  $\theta$  search illustration. Every point in the shaded space corresponds to a different model  $p_\theta(x)$ . The search objective is the minimization of the Kullback-Leibler divergence between the model  $p_\theta(x)$  and the true data distribution  $p(x)$ .

The first term – *negative entropy* – has no parameters, and thus will not affect the search for  $\theta$ :

$$\mathbb{D}_{\text{KL}}[p(x)||p_\theta(x)] \propto -\mathbb{E}_{x \sim p(x)} [\log p_\theta(x)] = -\sum_{x \in V^T} p(x) \log p_\theta(x). \quad (\text{B.5})$$

Calculation of the remaining term, however, is not feasible – we cannot access the distribution  $p(x)$ . Fortunately, we can approximate the term using an unbiased *Monte Carlo* estimate in a computationally feasible manner:

$$\mathbb{E}_{x \sim p(x)} [\log p_\theta(x)] \approx \frac{1}{K} \sum_{k=1}^K \log p_\theta(x_k). \quad (\text{B.6})$$

Here samples  $x_k$  are assumed to come from the true model  $p(x)$  and to be independent from each other – independent and identically distributed (i.i.d.). In essence, a dataset of such samples  $\mathcal{D} = \{x_k\}_{k=1}^K$  can be utilized to estimate the model  $p_\theta(x)$ . This simplifies the parameter search problem as shown in Eq. B.7.

$$\theta^* = \arg \max_{\theta \in \Theta} \sum_{k=1}^K \log p_\theta(x_k) \quad (\text{B.7})$$

Essentially, we search for  $\theta^*$  that makes our dataset maximally likely under  $p_\theta(x)$ , and it is known as the *maximum likelihood estimation* principle. Other alternatives to estimate  $p_\theta(x)$  also exist, and we refer an interested reader to Huszár (2015) for more information.



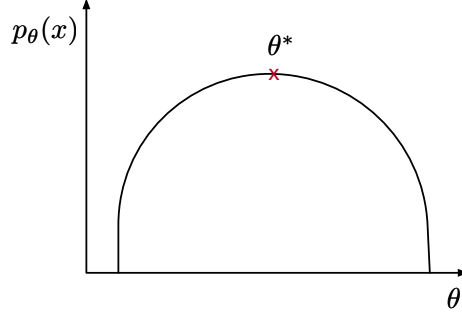


Figure 2.4: Concave search space.

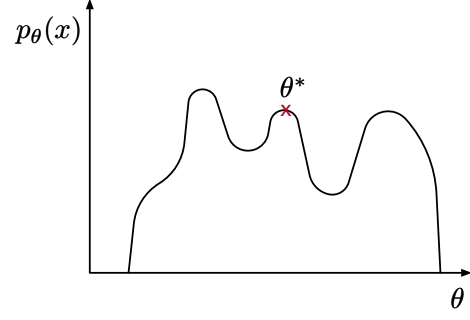


Figure 2.5: Non-concave search space.

The search over the space  $\Theta$ , however, presents its own challenges. First, the search space is non-concave and thus has many locally optimal points. We provide the illustrations of concave and non-concave spaces in Fig. 2.4 and Fig. 2.5, respectively. This implies we have no guarantees of finding the globally optimal  $\theta$ . The common approach is to search using a batch version of *stochastic gradient ascent* where ADAM (Kingma & Ba, 2014) is used to improve convergence.

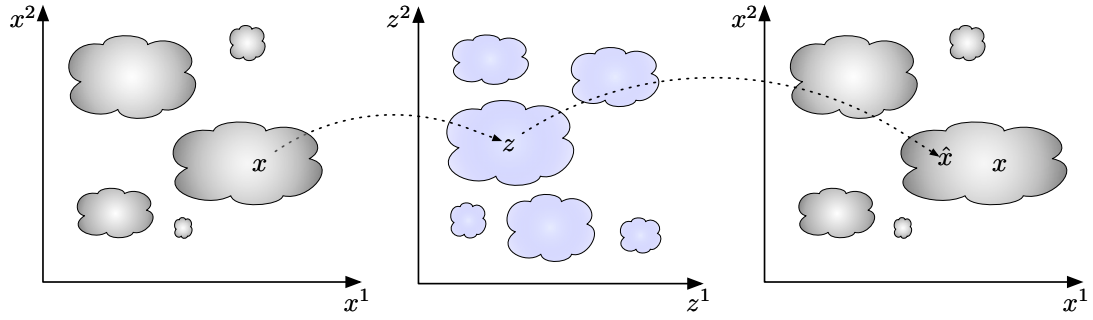
## 2.4 Auto-encoding

In this section, we describe a standard method for learning the semantic representations of data – *auto-encoding*. This method is fundamental for understanding *variational auto-encoding* described in Sec. 2.5. In its simplest form, we would like to represent our data as a continuous vector of features. These features contain compressed information about the actual data point, such as sentiment, style of writing, and content. In practice, the continuous representations are useful for controllable data generation, analysis, and as prior knowledge for downstream tasks.

In the auto-encoder framework, one starts by explicitly defining a feature-extracting function in a specific parameterized closed form. This function, denoted as  $e_\theta$ , we call an *encoder* and will allow the straightforward and efficient computation of a *latent code* vector  $z = e_\theta(x)$  from an input data point  $x$ .

Another function, responsible for the reconstruction of the original datum  $x$  from  $z$  is called a *decoder*. And it is defined as  $\hat{x} = d_\theta(z)$ . It can be seen as a mapping function from the latent space  $Z$  to the data space  $X$ . We train such an encoder-decoder model by optimizing the parameters  $\theta$  via the reconstruction loss for a dataset  $D = \{x\}_{i=1}^N$ , as defined in Eq. B.8.

$$\mathcal{L}(D; \theta) = \sum_{i=1}^N l(x_i, d_\theta(e_\theta(x_i))) \quad (\text{B.8})$$



**Figure 2.6:** Unconstrained latent space illustration. Here a datum  $x$  is mapped to the latent code  $z$  and then (imperfectly) reconstructed to  $\hat{x}$ .

Intuitively,  $l(\cdot)$  measures how well the encoder is able to produce a good representation  $z$  for  $x$  and how well this representation is mapped back to the original data-point  $x$ . Common choices for  $l(\cdot)$  are the Euclidean distance and negative log-likelihood.

It is important to notice that the variable  $z$  is not assumed to follow a predefined distribution. This can result in complex density *manifolds* that are learned (Bengio, Courville, & Vincent, 2013). We provide an illustration in Fig. 2.6.

In the absence of any restrictions, however, many surrounding regions in the manifold can never be explored in training. In turn, this can result in poor *generalization* to unseen data-points (e.g., in a test set). A common way to address this problem is by assigning a distribution over the latent space, such as a Gaussian. We will explore this option next in Sec. 2.5.

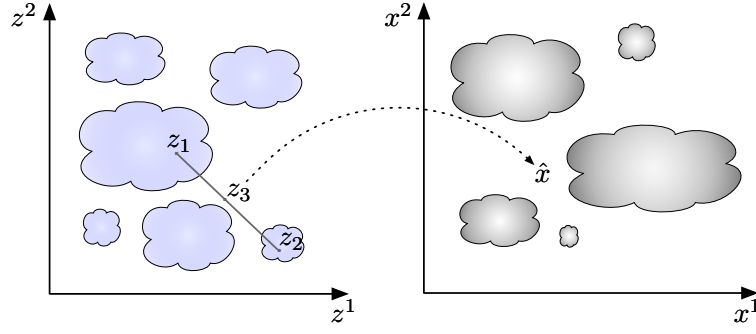
## 2.5 Variational Auto-encoding

One of the limitations of the auto-encoding approach is that the learned latent space often has disconnected regions of high density. There are no trivial ways to navigate these density regions and one might not be able to realize a valid data point from a latent code. Consider a scenario where we *linearly interpolate* between two latent codes  $z_1$  and  $z_2$ , as in Eq. B.9, by choosing  $\theta \in [0, 1]$ .

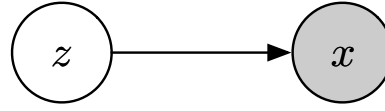
$$z_3 = \theta z_1 + (1 - \theta) z_2 \quad (\text{B.9})$$

This can result in  $z_3$  not covered in training, and thus to an implausible data point  $\hat{x}$  realized by the decoder, see Fig. 2.7.

A regularization of the latent space can alleviate this problem. Specifically, it can be achieved by adding a *prior* distribution  $p(z)$  over latent codes  $z$ . This technique can make the latent space more compact in terms of density. Consequently, it becomes better suited for generation of novel data (Bowman et al., 2016) as only a particular dense region of the latent space is



**Figure 2.7:** Linear interpolation between two latent codes  $z_1$  and  $z_2$ . Here  $z_3$  is realized to an implausible data point  $\hat{x}$  in the data space.



**Figure 2.8:** A graphical illustration of an observed variable  $x$  that depends on a latent variable  $z$ . The shaded and unshaded circles indicate observed and unobserved variables, respectively.

leveraged in training. Also, it often results in better empirical performance on end tasks. From the technical perspective, we assume that latent codes  $z$  are not directly observable unlike data points  $x$ . This *latent model* can be graphically represented in Fig. 2.8 with the log-likelihood shown in Eq. B.10.

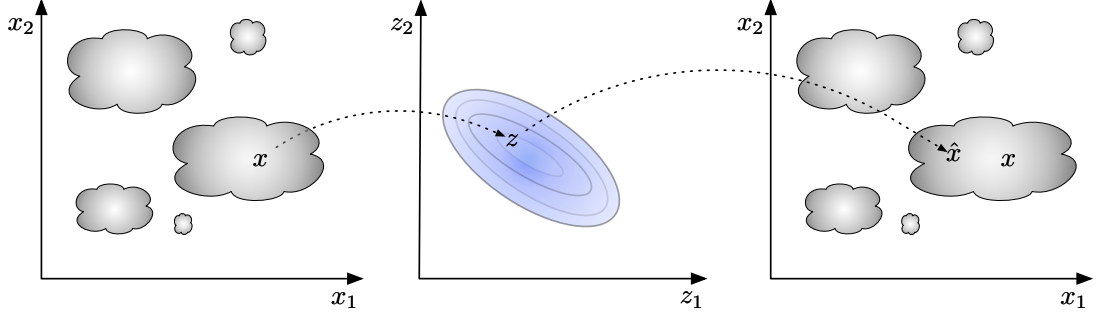
$$\log p_{\theta}(x) = \log \int p_{\theta}(x|z)p(z)dz \quad (\text{B.10})$$

Here,  $p_{\theta}(x|z)$  is known as *conditional likelihood* and we implement it as a decoder that predicts  $x$  when  $z$  is passed as input. And we often implement  $p(z)$  as the multivariate standard Normal. However, to compute the log-likelihood, we need to marginalize over all the possible values of  $z$ , which is not tractable. Furthermore, optimization is also intractable for gradient-based methods, see Eq. B.11.

$$\nabla_{\theta} \log p_{\theta}(x) = \frac{\nabla_{\theta} p_{\theta}(x)}{p_{\theta}(x)} \quad (\text{B.11})$$

Instead of maximizing the exact log-likelihood, we can perform a number of re-formulations in order to arrive to a *lower bound* that we subsequently maximize. First, we introduce the *approximate posterior distribution*  $q_{\phi}(z|x)$ , as shown in Eq. B.12.

$$\log p_{\theta}(x) = \log \int p_{\theta}(x|z)q_{\phi}(z|x)\frac{p(z)}{q_{\phi}(z|x)}dz = \log E_{z \sim q_{\phi}(z|x)} \left[ p_{\theta}(x|z)\frac{p(z)}{q_{\phi}(z|x)} \right] \quad (\text{B.12})$$



**Figure 2.9:** Constrained latent space illustration. Here the data point  $x$  is mapped to the latent code  $z$  and then (imperfectly) reconstructed to  $\hat{x}$ .

After training, the approximate posterior can be used for inference we discussed in Sec. 2.2. For instance, we could map data points to their corresponding latent codes and use these as features for downstream tasks.

For convenience, we can frame the posterior as a multivariate Gaussian distribution with the mean and diagonal covariance computed by neural networks:

$$q_\phi(z|x) = \mathbb{N}(z; \mu_\phi(x), \Sigma_\phi(x)). \quad (\text{B.13})$$

Here, we use neural functions  $\mu_\phi(x)$  and  $\Sigma_\phi(x)$  to yield corresponding statistics – the mean and diagonal covariance, to parametrize the distribution. Notice that we use a separate set of parameters –  $\phi$  – which are known as *variational parameters*. The actual architecture of the neural functions depends on the task. For NLP tasks specifically, we often use feed-forward neural networks that input the features of  $x$ , which are extracted by a separate model.

Further, we use the Jensen’s inequality (Boyd & Vandenberghe, 2004) to get the lower bound shown in Eq. B.15.

$$\log E_{z \sim q_\phi(z|x)} \left[ p_\theta(x|z) \frac{p(z)}{q_\phi(z|x)} \right] \geq E_{z \sim q_\phi(z|x)} \left[ \log p_\theta(x|z) \frac{p(z)}{q_\phi(z|x)} \right] = \quad (\text{B.14})$$

$$E_{z \sim q_\phi(z|x)} [\log p_\theta(x|z)] - \mathbb{D}_{\text{KL}} [q_\phi(z|x) || p(z)] \quad (\text{B.15})$$

The first term in Eq. B.15 is known as *reconstruction*, and the second one as *Kullback-Leibler divergence* (KLD). Intuitively, the former measures how well, on average, we predict the data point  $x$  when the latent code  $z$  becomes observed. The latter measures the deviation of the posterior  $q_\phi(z|x)$  from the prior  $p(z)$ . Essentially, the equation can be seen as a trade-off between storing information about  $x$  to  $z$  via  $q_\phi(z|x)$  and  $z$  being uninformative. We can visualize the learning process as shown in Fig. 2.9.

For Gaussian distributions  $q_\phi(z|x)$  and  $p(z)$ , the KLD term has the exact and tractable solution.<sup>3</sup> However, the reconstruction term does not and integration over all the possible assignments to  $z$  is not computationally feasible. Next, we will discuss efficient ways to compute an estimate of the reconstruction term. Specifically, we will look into two cases – Gaussian and Categorical  $z$  variables in Sec. 2.5.1 and Sec. 2.5.2, respectively.

### 2.5.1 Re-parametrization Trick

In this section, we continue the assumption that  $z$  is a continuous variable following a Gaussian distribution. And we would like to compute a Monte Carlo estimate of the reconstruction term, as in Eq. B.16.

$$\mathbb{E}_{z \sim q_\phi(z|x)} [\log p_\theta(x|z)] \approx \frac{1}{L} \sum_{l=1}^L \log p_\theta(x|z_l) \quad (\text{B.16})$$

Here samples  $\{z_l\}_{l=1}^L$  obtained from the posterior  $q_\phi(z|x)$ . Unfortunately, by default, we cannot backpropagate through the stochastic sampling process. A workaround this problem is called *re-parametrization*. In essence, we decouple trainable parameters from stochastic noise involved in the realization of an assignment to the variable  $z$ :

$$z = g_\phi(\epsilon, x) = \mu_\phi(x) + \Sigma_\phi(x)^{-\frac{1}{2}} * \epsilon. \quad (\text{B.17})$$

Recall that  $\mu_\phi(x)$  and  $\Sigma_\phi(x)$  are neural networks with trainable parameters  $\phi$ . And  $\epsilon \sim \mathcal{N}(0; \mathbb{I})$  has no trainable parameters. In turn, this makes it possible to backpropagate through the samples and thus train the system end-to-end.

### 2.5.2 REINFORCE

While re-parametrization works for certain continuous variables, it does not work for Categorical variables. Fortunately, we can re-formulate the gradient of the reconstruction term using a policy gradient method called *REINFORCE* (Williams & Zipser, 1989). We show the re-formulated reconstruction term's gradient in Eq. B.18.

$$\nabla_\phi \mathbb{E}_{z \sim q_\phi(z|x)} [\log p_\theta(x|z)] = \mathbb{E}_{z \sim q_\phi(z|x)} [\log p_\theta(x|z) \nabla_\phi \log q_\phi(z|x)] \quad (\text{B.18})$$

The gradient can be estimated via Monte Carlo by drawing samples from the posterior as shown in Eq. B.19.

$$\mathbb{E}_{z \sim q_\phi(z|x)} [\log p_\theta(x|z) \nabla_\phi \log q_\phi(z|x)] \approx \frac{1}{L} \sum_{l=1}^L [\log p_\theta(x|z_l) \nabla_\phi \log q_\phi(z_l|x)] \quad (\text{B.19})$$

3. <https://people.eecs.berkeley.edu/~jordan/courses/260-spring10/other-readings/chapter13.pdf>

In practice, however, this method suffers from high variance which makes it difficult to apply. To reduce the variance of the estimate, we can introduce a baseline term  $b(x)$  (Greensmith, Bartlett, & Baxter, 2004; Sutton & Barto, 2018), as shown in Eq. B.20.

$$\mathbb{E}_{z \sim q_\phi(z|x)} [(\log p_\theta(x|z) - b(x)) \nabla_\phi \log q_\phi(z|x)] \quad (\text{B.20})$$

A common choice for the baseline term is an estimate of the expected value:

$$b(x) = \mathbb{E}_{z \sim p_\theta(z)} [\log p(x|z)]. \quad (\text{B.21})$$

Other alternatives also exist, and we refer an interested reader to Rennie, Marcheret, Mroueh, Ross, and Goel (2017).

## 2.6 Language Models

In this section, we discuss categorical sequential data common in natural language processing tasks. Concretely, we focus on  $x = [x_1, \dots, x_T]$  which is a sequence of categorical variables. Each variable  $x_t$  corresponds to a word<sup>4</sup> in a sequence, i.e.,  $x_t \in V$ , where  $V$  is the vocabulary of words. As we are modelling sequences of words, we will further assume that each variable  $x_t$  depends on the *prefix*  $x_{1:t-1}$ . The prefix consists of all the preceding variables until the time-step  $t$ . The prefix is predictive of the current word  $x_t$ , and a language model can leverage the prefix to yield more accurate predictions of  $x_t$ . This follows well the human intuition. For instance, if asked to complete a partial sentence ‘*what a wonderful*’, one would more likely conclude that the next word is ‘*day*’ rather than ‘*play*’ based on prior knowledge. Formally, we can factorize a sequence  $x$  under the language model  $p_\theta(x)$  as in Eq. B.22.

$$p_\theta(x) = \prod_{t=1}^T p_\theta(x_t | x_{1:t-1}) \quad (\text{B.22})$$

Formally, we can state that  $x_t \sim \text{Cat}(f_\theta(x_{1:t-1}))$  for  $t = 1 \dots T$ . Here the function  $f_\theta(\cdot)$  yields a probability distribution over all the possible next words that immediately follow the prefix  $x_{1:t-1}$ . The function is often an auto-regressive neural network, such as Transformer (Vaswani et al., 2017) or RNN variant, such as GRU (Cho et al., 2014). The network has free parameters  $\theta$  that can be optimized to yield more accurate predictions.

---

4. A common alternative to words are subwords, such as BPE (Sennrich, Haddow, & Birch, 2016).

# Background: Summarization

---

In this chapter we focus on summarization. We start from news summarization due to its relevance to opinion summarization in Sec. 3.1. Then we discuss the history of opinion summarization in Sec. 3.2. Finally, we discuss summary evaluation in Sec. 3.3 and Sec. 3.4.

### 3.1 News Summarization

Before we discuss opinion summarization in Sec 3.2, we briefly introduce news summarization. This branch is better established and highly related opinion summarization. Also, we will contrast their major differences.

First of all, news summarization is pre-dominantly single-document (Bražiński, Lapata, & Titov, 2021; Nallapati, Zhou, dos Santos, Güçehre, & Xiang, 2016; Paulus et al., 2017; Rush, Chopra, & Weston, 2015; See, Liu, & Manning, 2017). This setup is simpler as either input documents are relatively short or top paragraphs contain the most important information. In opinion summarization, however, a product can have thousands of reviews. From the modelling perspective, this calls for creative ways to summarize them. Additionally, users who have bought the same product might have conflicting opinions. Unlike in news summarization, here the model needs to perform conflict resolution.

Second, news are factual in terms of their content while reviews are subjective. This affects both the writing style and content. In terms of the latter, it makes it more challenging to assess whether an opinion in the summary is supported by the input reviews.

Lastly, news datasets often have hundreds of thousands of article-summary pairs. For example, CNN/DM (Hermann et al., 2015) has about 300,000 pairs. And modern neural models often require large datasets for fine-tuning. However, in opinion summarization, datasets are scarce and most have less than 300 reviews-summary pairs. In turn, this calls for creative ways these small datasets can be utilized for fine-tuning. We will discuss this in Sec. 1.2.2.

aspect	rating	supporting phrases
sound	5/5	sound is very clear, excellent bass
price	4/5	best deal on the market, good value for the price
shipping	3/5	fast shipping, the order arrived damaged
bluetooth	5/5	great wireless reception, connection is great
...	...	...

**Figure 3.1:** An aspect-rating summary for a product with supporting phrases. Aspects are rated and can be either explicit or implicit.

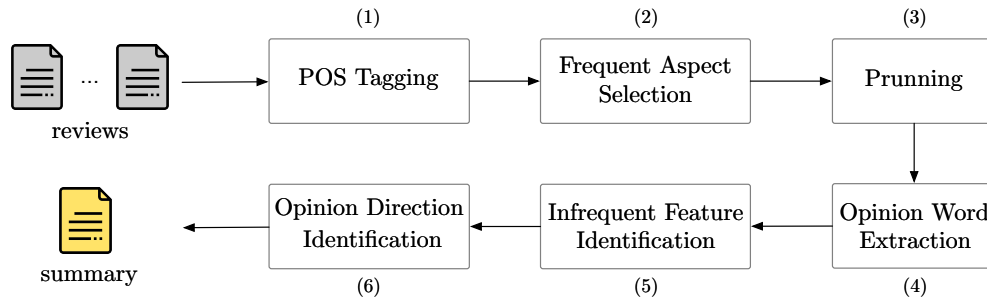
## 3.2 History of Opinion Summarization

Opinion summarization has a long history due to its practical usefulness and challenging research problems. In early 2000, e-commerce was becoming more and more popular and the number of customer reviews for products and services was growing rapidly. This was making it difficult for a user to read them all. This challenge called for methods that can compress opinions to a more compact form to assist the user in making purchasing decisions. The first steps in this direction can be traced back to review sentiment identification (Pang & Lee, 2004; Pang, Lee, & Vaithyanathan, 2002). The task was to accurately determine/classify the user sentiment towards a particular product, often as a rating between 1 and 5. The product summary then can be computed as the average rating. However, it only allowed to identify the sentiment on the document level. In Sec. 3.2.1 we discuss a more fine-grained format – *aspect-rating summaries* – as in Fig. 3.1. These summaries consist of aspects, their associated ratings, and supporting fragments. In Sec. 3.2.2 we will focus on summaries consisting entirely of text snippets. These are either composed of review sentences or generated by a model.

### 3.2.1 Aspect-rating Summaries

Aspect-rating summaries were constructed in a number of steps using pipeline approaches. In general, they had the following schema. First, aspects were identified using an aspect extractor. Second, their ratings were determined using a classifier. Finally, supporting phrases were extracted from reviews to form a summary like in Fig. 3.1. In this section, we will focus on the first step – aspect extraction – and present two salient works. The first one uses heuristics to extract **explicit** aspect mentions. The second one, leverages topic models that can extract both **explicit** and **implicit** aspect mentions.





**Figure 3.2:** Heuristics-based system for opinion summarization.

### Heuristic-based

The first opinion summarization system of this type was proposed in Hu and Liu (2004). From the high-level, the system works as shown in Fig. 3.2. In the first step, review sentences are parsed by a part-of-speech tagger (Manning & Schutze, 1999) and only nouns/noun phrases are retained. In the second step, frequently occurring phrases are determined using an association rule miner (B. Liu, Hsu, Ma, et al., 1998). In the third step, uninteresting and redundant phrases are removed (pruned). In the forth step, opinion words, such as ‘horrible’ and ‘incredible’, surrounding aspects are identified. These are adjacent adjectives that modify the aspect/aspect phrase. In the fifth step, rare but interesting aspects are extracted using heuristics and POS tags. In the sixth step, opinion words are mapped to WordNet (Miller, Beckwith, Fellbaum, Gross, & Miller, 1990) to determine the sentiment of surrounding aspects. Lastly, aspects and their sentiments are aggregated to be presented to the user in the form of a summary.

The main downside of this approach is that it extracts aspects without any consideration of implicit mentions. For example, in the phrase: ‘*This speaker could be a bit louder and less noisy,*’ it would miss the aspect ‘*sound.*’ The approach presented next is based on topic models and can extract implicitly mentioned aspects.

### Topic-based

In MG-LDA (Titov & McDonald, 2008), the authors proposed a more elegant approach based on probabilistic models. Specifically, they proposed to extract aspects from customer reviews by clustering review sentences/phrases by topics. These topics correspond to **ratable aspects** and provide a level of abstraction in contrast to Hu and Liu (2004), that extracts explicitly mentioned aspects from text.

On the modelling side, the authors proposed to extend the LDA topic model (Blei, Ng, & Jordan, 2003) which was not designed for aspect extraction from customer reviews. The proposed approach models both review-level (global) and context window (local) topics. The distribution of global topics is fixed for a review while the distribution of local topics is allowed to vary

	label	top words
MG-LDA local topics	sound quality	sound quality headphones volume bass earphones good
	features	games features clock contacts calendar alarm
	controls	button play track menu song buttons volume album
	battery	battery hours life batteries charge aaa rechargeable
	accessories	usb cable headphones adapter remote plug power

**Table 3.1:** Top words from MG-LDA for MP3 players reviews.

across the review. A word in a review is sampled either from the mixture of global topics or from the mixture of local topics specific for the local context of the word. The hypothesis is that ratable aspects will be captured by local topics and global topics will capture the properties of reviewed items. See an example with word assignments to local topics in Table 3.1. To infer topics for each phrase, they used collapsed Gibbs sampling (Griffiths & Steyvers, 2004). Further, these phrases can be filtered and their sentiment can be determined by a separate model to construct a summary.<sup>1</sup>

Summaries created from aspects and their sentiments are conceptually simple yet often lack details to provide a good overview over associated opinions. One of the limitations is that supporting phrases (see Fig. 3.1) are not necessarily summarizing as are often selected by simple heuristics. Moreover, the summary format is not very natural and can be perceived as not user-friendly (Murray, Hoque, & Carenini, 2017). Next, we will discuss a more natural format, where a summary is comprised only of texts.

### 3.2.2 Text Summaries

Broadly, there are two types of text summaries.<sup>2</sup> The first ones are called *extractive*, and the second ones are *abstractive*. In the first case, a summary is constructed by concatenating summarizing input fragments (e.g., sentences or segments). Often, the summarizer here is responsible for identifying these salient fragments in input reviews and selecting the summarizing ones to form a summary. However, there are three main downsides of extractive opinion summaries that we discuss next.

First, customer reviews are intrinsically controversial as users often disagree about various aspects (e.g., price and photo quality). This makes it challenging to select summarizing fragments to form a summary. As the degree of controversy increases, the less effective extractive summarizers become (Carenini & Cheung, 2008b).

Second, extractive summarizers are unable to abstract and paraphrase information present in input reviews. For example, consider the following sentences about food:

1. A similar approach was also taken in Lu, Zhai, and Sundaresan (2009), where they used topic models to extract ratable aspects.

2. Another type that has attracted some recent attention is called *compressive*. But to the best of our knowledge, there are no compressive opinion summarizers

1. The stake was cold.
2. The pasta was too dry.
3. The chocolate cake was completely tasteless.
4. The fudge had a synthetic taste.

Regardless of what sentence an extractive summarizer selects, it would not be summarizing and only provide a partial perspective about the food. In such a case, an abstractive summary, such as ‘the main dishes and deserts are bad and not recommended’, is a more complete summary. Notice that this summary abstracts details about food by using a richer vocabulary of words (Lebanoff, Song, & Liu, 2018; See et al., 2017).

The third problem comes from the fact that when multiple sentences are extracted, they might not form a coherent summary. This is especially problematic when fine-grained segments are extracted, such as EDUs (Angelidis & Lapata, 2018; Mann & Thompson, 1988).

### Opinosis

OPINOSIS (Ganesan, Zhai, & Han, 2010) was one of the first text generating opinion summarizers. The authors argued that extractive summaries are generally not well suited for this task due to the subtle variations of redundant opinions in customer reviews. Instead, they proposed a text generative model that produces short opinion abstracts. From the technical perspective, the key idea is to construct a textual graph (called OPINOSIS-GRAPH) that represents the text to be summarized. This casts the problem of summarization as finding appropriate paths in the graph. Here, nodes represent words with directed edges representing the structure of sentences. At inference time, it produces a summary by repeatedly searching the OPINOSIS-GRAPH for appropriate subgraphs that both encode a valid sentence and have high redundancy scores (thus representative for the major opinions). The sentences encoded by these subgraphs then form an abstractive summary. This model, however, can produce summaries only containing words that occur in the text to be summarized. In essence, it can be seen as an extractive model with elements of fusion (combining extracted portions) and compression (squeezing out unimportant material from a sentence). Two summaries produced by the system are shown below:

1. *The food was excellent, good and delicious. Very good selection of food.*
2. *Free wine reception in evening. Free coffee and biscotti and wine.*

### Template-based

While OPINOSIS produces short abstractive summaries, it has a number of limitations. First, the method does not provide a well-formed grammatical abstract and the generated summary only contains words that occur in the original input texts. Second, generated summaries do not contain any information about the distribution of opinions. These limitations were addressed in TEMPLATESUM (Gerani, Mehdad, Carenini, Ng, & Nejat, 2014).

The proposed model utilizes microplanning and sentence realization modules to produce summaries. The first module selects a template, such as '*Almost all costumers mentioned the **[X]** and they*' and '*felt that it was very poor*'. Here, **[X]** is an aspect that should be filled by the sentence realization module. The content of the slots is based on aspects and their relations present in input reviews. Which, in turn, are obtained using discourse tree aggregation and aspect importance weighting via a graph-based PAGERANK algorithm (Page, Brin, Motwani, & Winograd, 1999). An example summary produced by the model is shown below with aspects in **bold**.

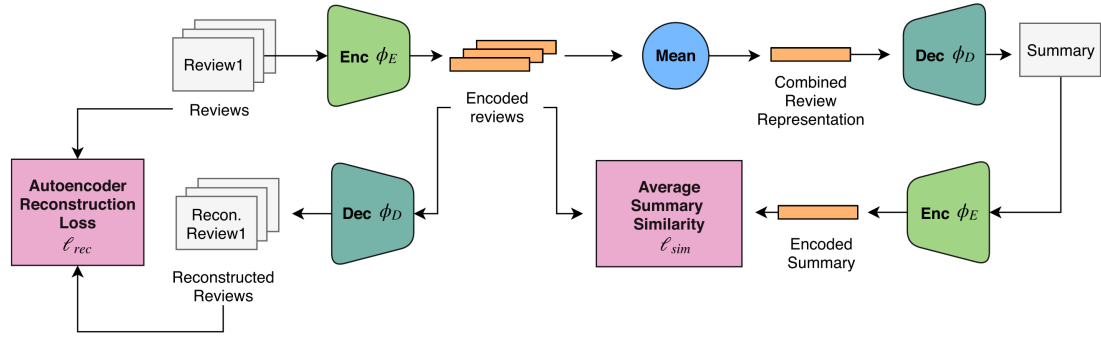
*All reviewers (34 people), who commented on the **camera**, felt that it was really good mainly because of the **picture**. Around 26% of the reviewers expressed their opinion about the **picture quality** and they really liked it.*

While templates can help to abstract information to a certain degree, their number is limited and their creation requires human effort. Moreover, the model can only rely on input review words to fill template slots. This strongly restricts the model's output and the amount of information that can be conveyed. Next, we will discuss a more flexible model that generates summaries using a free vocabulary of words, without reliance on templates.

### MeanSum

One of the central reasons behind a limited progress in abstractive summarization to this point was the absence of annotated datasets. Deep neural encoder-decoder models usually require large amounts of text data for training. In Chu and Liu (2019), the authors approached this problem from the unsupervised perspective by introducing MEANSUM. The central idea was to leverage customer reviews without human-written summaries for training. The customer reviews were grouped by their associated products and used both as source and prediction target by the model. The product summary was represented as a sequence of latent categorical variables. Here, each variable corresponded to a subword that was sampled auto-regressively from the decoder.

On the modelling side, a pre-initialized LSTM-based encoder-decoder (Sutskever, Vinyals, & Le, 2014) is trained with two objectives in a multi-task fashion. Under the first objective, the decoder is reconstructing reviews from their encodings. In essence, this allows for training the text generator (decoder) and representation module (encoder). Under the second objective,



**Figure 3.3:** Illustration of the MEANSUM training procedure.

the model is maximizing the cosine similarity between the summary and review encodings, and it works as follows. First, reviews are encoded, averaged, and passed to the decoder as input. Second, a summary is sampled auto-regressively from the decoder. Third, the sampled summary is encoded and the encoding is compared with input review encodings in terms of the angle. Essentially, the second objective is regularizing the summary to be semantically similar to input reviews. The training procedure is schematically illustrated in Fig. 3.3. After training, one can generate summaries as the one shown below.

*Everything is so good I had the chicken souvlaki with a side of rice. Best decision I've ever had. Not a bad place to eat, but they have a large selection of local food which is nice. My wife and I'll be back for sure.*

While this was the first fully abstractive opinion summarizer, it had a number of limitations. First, the model produces summaries with review-like text characteristics and contents. Specifically, the content is often **uninformative** and **written in the informal writing style**. For instance, it often contains fragments like: *'I'll be back for sure'*. Second, generated summary content is often not present in input reviews, i.e., the model hallucinates (Maynez et al., 2020). For example, the summary might be about 'iPhone' while reviews are about 'iPad'. These two problems make the summarizer not very suitable for practical purposes. In the thesis, we address these problems in our unsupervised (Chapter 4) and few-shot approaches (Chapters 5 and 6).

### 3.3 Automatic Evaluation

The most common automatic evaluation metrics in summarization are ROUGE (Lin, 2004).<sup>3</sup> In essence, they measure lexical overlap between reference ( $r$ ) and hypothesis summaries ( $h$ ). ROUGE includes a number of sub-metrics as described next. The first set of metrics are *ROUGE-N*, which are based on  $n$ -gram overlap between reference and generated summaries. We compute *recall* and *precision* as shown in Eq. C.1 and Eq. C.2, respectively.

$$\text{ROUGE-N}_{rec}(h, r) = \frac{||\text{n-gram}(h) \cap \text{n-gram}(r)||}{||\text{n-gram}(r)||} \quad (\text{C.1})$$

$$\text{ROUGE-N}_{prec}(h, r) = \frac{||\text{n-gram}(h) \cap \text{n-gram}(r)||}{||\text{n-gram}(h)||} \quad (\text{C.2})$$

Here,  $||\text{n-gram}(h) \cap \text{n-gram}(r)||$  is the size of the overlapping set of  $n$ -grams, and  $||\text{n-gram}(r)||$  is the total number of  $n$ -grams in  $r$ . Further, we compute the *F score* as shown in Eq. C.3.

$$\text{ROUGE-N}_f(h, r) = 2 \frac{\text{ROUGE-N}_{prec}(h, r) * \text{ROUGE-N}_{rec}(h, r)}{\text{ROUGE-N}_{prec}(h, r) + \text{ROUGE-N}_{rec}(h, r)} \quad (\text{C.3})$$

The last metric is called *ROUGE-L* and it measures longest common subsequence (LCS) between generated and reference summary in terms of recall and precision in Eq. C.4 and Eq. C.5, respectively.

$$\text{ROUGE-L}_{rec}(h, r) = \frac{||\text{LCS}(h, r)||}{||r||} \quad (\text{C.4})$$

$$\text{ROUGE-L}_{prec}(h, r) = \frac{||\text{LCS}(h, r)||}{||h||} \quad (\text{C.5})$$

Here,  $||\text{LCS}(h, r)||$  corresponds to the size of the longest common subsequence, and  $||r||$  and  $||h||$  to the lengths of reference and generated summaries, respectively. Lastly, we calculate the *F score*, analogously to ROUGE-N.

While ROUGE is ubiquitous in the summarization literature, it has a number of problems. First, it can be insensitive to sentiment (Tay et al., 2019). For instance, reviews might be negative about a product, yet a summarizer might mistakenly generate a positive summary. While this can significantly reduce the trustworthiness of the system, such mistakes often only marginally affect the ROUGE scores. Second, for practical reasons, we want to generate *input faithful* summaries which is an open problem in summarization (Maynez et al., 2020). For instance, the

3. Other alternatives to ROUGE exist, such as METEOR (Banerjee & Lavie, 2005), yet it is used less extensively in summarization literature.

model might confuse ‘iPhone’ and ‘iPad’ due to their semantic similarity. However, a number of studies in various summarization domains have shown that ROUGE is not well correlated with input faithfulness (Fabbri et al., 2021; Falke, Ribeiro, Utama, Dagan, & Gurevych, 2019; Maynez et al., 2020). Finally, it is hard to measure summary informativeness as all n-grams are treated equally. To counter all aforementioned issues, we perform human evaluation studies as we discuss in Sec. 3.4.

## 3.4 Human Evaluation

In order to assess human preference and evaluate the amount of hallucinations, we conduct human evaluation studies. Specifically, we utilize a crowd-sourcing platform like Amazon Mechanical Turk, where we can employ professional workers to perform the evaluation. Such workers are instructed about the task and trained before performing evaluation. Broadly, there are two types of studies we perform. The first one is known as *Best-Worst Scaling* and the second one as *content support*. The former measures human preference based on various criteria and the latter measures input faithfulness.

### 3.4.1 Best-Worst Scaling

This type of human study assesses human preference based on a number of criteria. The judgment criteria are presented below, where non-redundancy and coherence were taken from Dang (2005).

1. *Fluency*: the summary sentences should be grammatically correct, easy to read and understand;
2. *Coherence*: the summary should be well structured and well organized;
3. *Non-redundancy*: there should be no unnecessary repetition in the summary;
4. *Informativeness*: how much useful information about the product does the summary provide?;
5. *Sentiment*: how well the sentiment of the summary agrees with the overall sentiment of the original reviews?

The evaluation is performed using Best-Worst Scaling (BWS) methodology (Louviere et al., 2015; Louviere & Woodworth, 1991). BWS has been shown to produce more reliable results than ranking scales (Kiritchenko & Mohammad, 2016b). The reviews and a number of summaries are presented to workers. The workers are asked to select the best and worst summary for each criterion. For every criterion, a system’s score is computed as the percentage of times it was selected as best minus the percentage of times it was selected as worst (Orme, 2009). The scores range from -1 (unanimously worst) to +1 (unanimously best).

### 3.4.2 Content Support

As we mentioned previously, we assess the input faithfulness of summaries via a content support study. We split summary sentences and present them to workers along with reviews. For each summary sentence, we ask to assess how well content of the sentence is reflected in the reviews. The workers have three available options:

1. *Full support*: all the content is reflected in the reviews;
2. *Partial support*: only some content is reflected in the reviews;
3. *No support*: content is not reflected in the reviews.

Afterwards, we normalize scores to obtain proportions for each category being selected.



## PART I

# Low-Resource Opinion Summarization

# Unsupervised Opinion Summarization as Copycat-Review Generation

---

In this chapter, we validate the first hypothesis (see Sec. 1.3) that customer reviews provide a sufficiently strong signal for learning opinion summarizers. We capitalize on redundancies and common opinions in the reviews of the same product. Intuitively, if we know that nine reviews are negative about price, it makes it very likely that the tenth review is also negative about it. This observation is at the core of the *leave-one-out* unsupervised<sup>1</sup> training objective. Specifically, given a collection of product reviews, a model predicts a held-out review conditioned on the remaining ones. The held-out review acts as a pseudo summary that a model predicts by leveraging common information expressed in other reviews. We will also use this objective for pre-training before few-shot fine-tuning on gold summaries in Chapters 5 and 6.

We also introduce a latent hierarchical model – COPYCAT. Here, we explicitly model individual review and product semantics with continuous variables. These variables have associated prior densities acting as bottlenecks on the amount of information about the pseudo summary the decoder receives. We show that the mean values of these variables result in summarizing texts generated by the decoder. We train the model end-to-end using amortized inference (Kingma & Welling, 2013), and show that our approach generates summaries that are fluent, coherent, and input faithful.

## 4.1 Introduction

Summarization of user opinions expressed in online resources, such as customer reviews, has drawn much attention due to its potential for various applications (Angelidis & Lapata, 2018; Hu & Liu, 2004; Medhat, Hassan, & Korashy, 2014). Opinion summaries capture user experiences and are useful for better purchasing decisions. Furthermore, they can be automatically produced for large volumes of products and services by trained models. However, as we discussed in Sec. 1.2.2, one of the central challenges in opinion summarization is the

---

1. We use terms unsupervised and self-supervised interchangeably.

<b>Summary</b>	This restaurant is a hidden gem in Toronto. The food is delicious, and the service is impeccable. Highly recommend for anyone who likes French bistro.
<b>Reviews</b>	We got the steak frites and the chicken frites both of which were very good ... Great service ...    I really love this place ... Côte de Boeuf ... A Jewel in the big city ...    French jewel of Spadina and Adelaide , Jules ... They are super accommodating ... moules and frites are delicious ...    Food came with tons of greens and fries along with my main course , thumbs uppp ...    Chef has a very cool and fun attitude ...    Great little French Bistro spot ... Go if you want French bistro food classics ...    Great place ... the steak frites and it was amazing ... Best Steak Frites ... in Downtown Toronto ...    Favourite french spot in the city ... crème brule for dessert

**Table 4.1:** A summary produced by our model; colors encode its alignment to the input reviews. The reviews are truncated, and delimited with the symbol ‘||’.

scarcity of annotated data required to train summarizers. Annotated data, where customer reviews are mapped to opinion summaries, is expensive to produce because annotators need to read many reviews. Moreover, annotation efforts would have to be undertaken for multiple domains as online reviews are inherently multi-domain (Blitzer, Dredze, & Pereira, 2007) and summarization systems can be domain-sensitive (Isonuma, Fujino, Mori, Matsuo, & Sakata, 2017). This is in sharp contrast to the summarization of non-subjective documents (P. J. Liu et al., 2018; Nallapati et al., 2016; Paulus et al., 2017; Rush et al., 2015; See et al., 2017) where annotated samples are often available in large quantities (hundreds of thousands). Perhaps unsurprisingly, there is a long history of applying unsupervised and weakly-supervised methods to opinion summarization (e.g., Angelidis and Lapata 2018; Mei, Ling, Wondra, Su, and Zhai 2007; Titov and McDonald 2008). However, these approaches have primarily focused on extractive summarization, i.e., producing summaries by copying parts of the input reviews.

In this chapter, we consider abstractive summarization which involves generating new phrases, possibly rephrasing or using words that were not in the original text. Abstractive summaries are often preferable to extractive ones as they can synthesize content across documents avoiding redundancy (Barzilay, McKeown, & Elhadad, 1999; Carenini & Cheung, 2008a; Di Fabbrizio, Stent, & Gaizauskas, 2014). In addition, we focus on the unsupervised setting and do not use any summaries for training. Unlike aspect-based summarization (B. Liu, 2012), which rewards the diversity of opinions, we aim to generate summaries that represent *consensus* (i.e., dominant opinions in reviews). We argue that such summaries can be useful for quick decision making, and to get an overall feel for a product or business (see the example in Table 4.1).

More specifically, we assume we are provided with a large collection of reviews for various products and businesses and define a generative model of this collection. Intuitively, we want to design such a model that, when generating a review for a product<sup>2</sup> relying on a set of other reviews, we can control the “amount of novelty” going into the new review or, equivalently, vary the extent to which it deviates from the input. At test time, we can force the novelty to be minimal, and generate summaries representing consensus opinions.

We capture this intuition by defining a hierarchical variational autoencoder (VAE) model. Both products and individual reviews are associated with latent representations. Product representations can store, for example, overall sentiment, common topics, and opinions expressed about the product. In contrast, latent representations of reviews depend on the product representations and capture the content of individual reviews. While at training time the latent representations are random variables, we fix them to their respective means at test time. As desired for summarization, these ‘average’ (or ‘copycat’) reviews differ in writing style from a typical review. For example, they do not contain irrelevant details that are common in customer reviews, such as mentioning the occasion or saying how many family members accompanied the reviewer. In order to encourage the summaries to include specific details, the review generator (‘decoder’) has direct access to the text of input reviews through the pointer-generator mechanism (See et al., 2017). In the example in Table 4.1, the model included specific information about the restaurant type and its location in the generated summary. As we will see in ablation experiments, without this conditioning, model performance drops substantially, as the summaries become more generic.

We evaluate our approach on two datasets, Amazon product reviews and Yelp reviews of businesses. The only previous method dealing with unsupervised multi-document opinion summarization, as far as we are aware of, is MEANSUM (Chu & Liu, 2019). Similarly to our work, they generate consensus summaries and consider the Yelp benchmark. Whereas we rely on continuous latent representations, they treat the summary itself as a discrete latent representation of a product. Although this captures the intuition that a summary should relay key information about a product, using discrete latent sequences makes optimization challenging; Chu and Liu (2019); Miao and Blunsom (2016) all have to use an extra training loss term and biased gradient estimators.

Our contributions can be summarized as follows:

- we introduce a simple end-to-end approach to unsupervised abstractive summarization;
- we demonstrate that the approach substantially outperforms the previous method, both when measured with automatic metrics and in human evaluation;

---

2. For simplicity, we refer to both products (e.g., iPhone X) and businesses (e.g., a specific Starbucks branch) as *products*.

- we provide a dataset of abstractive summaries for Amazon products.<sup>3</sup>

## 4.2 Related Work

Extractive weakly-supervised opinion summarization has been an active area of research. A recent example is Angelidis and Lapata (2018). First, they learn to assign sentiment polarity to review segments in a weakly-supervised fashion. Then, they induce aspect labels for segments relying on a small sample of gold summaries. Finally, they use a heuristic to construct a summary of segments. OPINOSIS (Ganesan et al., 2010) does not use any supervision. The model relies on redundancies in opinionated text and PoS tags in order to generate short opinions. This approach is not well suited for the generation of coherent long summaries and although it can recombine fragments of input text, it cannot generate novel words and phrases. LEXRANK (Erkan & Radev, 2004) is an unsupervised extractive approach which builds a graph in order to determine the importance of sentences, and then selects the most representative ones as a summary. Isonuma, Mori, and Sakata (2019) introduce an unsupervised approach for single review summarization, where they rely on latent discourse trees. Other earlier approaches (Di Fabbri et al., 2014; Gerani et al., 2014) relied on text planners and templates, while our approach does not require rules and can produce fluent and varied text. Finally, conceptually related methods were applied to unsupervised single sentence compression (Miao & Blunsom, 2016). The most related approach to ours is MEANSUM (Chu & Liu, 2019) which treats a summary as a discrete latent state of an autoencoder. In contrast, we define a hierarchical model of a review collection and use continuous latent codes. Retrospectively, the authors of CONSISTSUM (Ke, Gao, Shen, & Cheng, 2022) improved the performance over our model by using a large pre-trained language model and a more tailored way to produce synthetic datasets. Specifically, they considered the consistency of aspects and sentiments in synthetic pairs (reviews and pseudo summaries).

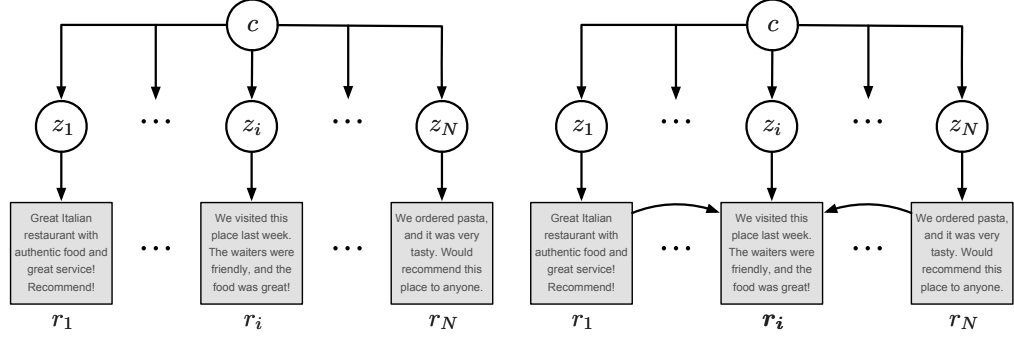
## 4.3 Model Description

### 4.3.1 Unsupervised Training Objective

Human-written summaries are not the only source of training signal for models. Because such summaries are scarce, we use synthetic datasets constructed from customer reviews to resemble the multi-document summarization task. Unlike human-written summaries, customer reviews are available in large quantities. For example, Amazon has more 230 millions of reviews. In the simplest scenario, we construct the dataset as follows. For each set of reviews of a

---

3. Data and code: <https://github.com/abrazinskas/Copypcat-abstractive-opinion-summarizer>.



(a) Conditional independence of the reviews given the group representation  $c$ . (b) The  $r_i$ 's decoder accesses other reviews of the group  $(r_1, \dots, r_{i-1}, r_{i+1}, \dots, r_N)$ .

**Figure 4.1:** Unfolded graphical representation of the model in terms of random variables and associated. Here  $r$  is review texts,  $c$  and  $z$  correspond to representations of a product and review, respectively.

product, we sample a review and use it a summary. In the same vein, we sample  $N$  source reviews to be passed to the model as input. The central observation is that reviews of the same product are correlated in terms of their content (e.g., aspects and entities) and a model can learn these correlations to later generate a representative review. Intuitively, if 9 reviews out of 10 praise meals in a restaurant, it is a strong indicator that the 10th review will praise the meals too. As we showed in this chapter, these patterns combined with modelling choices result in learned summarizers that generate fluent and coherent summaries, reflecting common opinions.

### 4.3.2 Model Overview

Our text collection consists of groups of reviews, with each group corresponding to a single product. Our latent summarization model (COPYCAT) captures this hierarchical organization and can be regarded as an extension of the vanilla text-VAE model Bowman et al. (2016), which we described in Sec. 2.5.

COPYCAT can be graphically represented as a Bayesian network shown in Fig. 4.1a. At the top, we associate each review group (equivalently, each product) with a continuous variable  $c$ , which captures the group's 'latent semantics'. This variable is latent and is not directly observable in data. In addition, we associate each individual review ( $r_i$ ) with a continuous variable  $z_i$ , encoding the semantics of that review. The information stored in  $z_i$  is used by the decoder  $p_\theta(r_i|z_i)$  to produce review text  $r_i$ .

The marginal log-likelihood of one group of reviews  $r_{1:N} = (r_1, \dots, r_N)$  is given by

$$\log p_\theta(r_{1:N}) = \log \int \left[ p_\theta(c) \prod_{i=1}^N \left[ \int p_\theta(r_i|z_i) p_\theta(z_i|c) dz_i \right] dc \right],$$

where we marginalize over variables  $c$  and  $z_{1:N}$ . When generating a new review  $r_i$ , given the set of previous reviews  $r_{1:i}$ , the information about these reviews has to be conveyed through the latent representations  $c$  and  $z_i$ . This bottleneck is undesirable, as it will make it hard for the model to pass fine-grain information. For example, at generation time, the model should be reusing named entities (e.g., product names or technical characteristics) from other reviews rather than ‘hallucinating’ or avoiding generating them at all, resulting in generic and non-informative text. We alleviate this issue by letting the decoder directly access other reviews. We can formulate this as an auto-regressive model:

$$p_{\theta}(r_{1:N}|c) = \prod_{i=1}^N p_{\theta}(r_i|r_1, \dots, r_{i-1}, c). \quad (\text{D.1})$$

As we discuss in Section 4.4, the conditioning is instantiated using the pointer-generator mechanism See et al. (2017), and, thus, specifically helps in predicting/generating rare words (e.g., named entities).

The formulation in Eq. D.1, imposes a particular order due to the chain rule. Instead, we want our summarizer to equally rely on every review, without imposing any order (e.g., temporal) on the generation process. As shown in Fig. 4.1b, when generating  $r_i$ , we let the decoder access all other reviews within a group,  $r_{-i} = (r_1, \dots, r_{i-1}, r_{i+1}, \dots, r_N)$ . This is closely related to pseudolikelihood estimation Besag (1975) or Skip-Thought’s objective Kiros et al. (2015). The final objective that we maximize for each group of reviews  $r_{1:N}$ :

$$\log \int p_{\theta}(c) \prod_{i=1}^N \left[ \int p_{\theta}(r_i|z_i, r_{-i}) p_{\theta}(z_i|c) dz_i \right] dc. \quad (\text{D.2})$$

We will confirm in ablation experiments (in Sec. 4.8.1) that both hierarchical modeling (i.e., using  $c$ ) and the direct conditioning on other reviews are beneficial.

### 4.3.3 Model Estimation

To estimate the model we leverage variational inference and VAE specifically (Kingma & Welling, 2013). For a more detailed presentation of the technique, please refer to Sec. 2.5. We start with the log-likelihood in Eq. D.3a and introduce two approximate posteriors, also called *inference networks*, as in Eq. D.3b.

$$\log \int p_\theta(c) \prod_{i=1}^N \left[ \int p_\theta(r_i | z_i, r_{-i}) p_\theta(z_i | c) dz_i \right] dc = \quad (\text{D.3a})$$

$$\log \int \left[ p_\theta(c) \frac{q_\phi(c | r_{1:N})}{q_\phi(c | r_{1:N})} \prod_{i=1}^N \left[ \int p_\theta(r_i, z | c, r_{-i}) \frac{q_\phi(z | r_i, c)}{q_\phi(z | r_i, c)} dz \right] \right] dc = \quad (\text{D.3b})$$

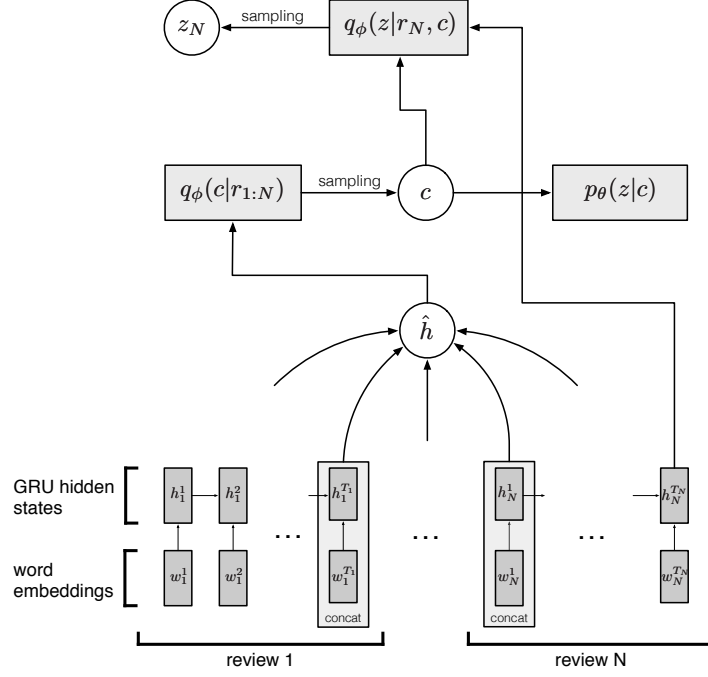
$$\log \mathbb{E}_{c \sim q_\phi(c | r_{1:N})} \left[ \frac{p_\theta(c)}{q_\phi(c | r_{1:N})} \prod_{i=1}^N \mathbb{E}_{z \sim q_\phi(z | r_i, c)} \left[ \frac{p_\theta(r_i, z | c, r_{-i})}{q_\phi(z | r_i, c)} \right] \right] \quad (\text{D.3c})$$

The inference networks –  $q_\phi(c | r_{1:N})$  and  $q_\phi(z_i | r_i, c)$  – are neural networks parameterized with  $\phi$  and will be discussed in detail in Sec. 4.4. They approximate the corresponding posterior distributions of the model. Further, we can re-formulate the sums as the expectation terms shown in Eq. D.3c. At this stage, we can leverage the logarithms concavity (Boyd & Vandenberghe, 2004) to get the lower bound shown in Eq. D.4.

$$\begin{aligned} \log \mathbb{E}_{c \sim q_\phi(c | r_{1:N})} \left[ \frac{p_\theta(c)}{q_\phi(c | r_{1:N})} \prod_{i=1}^N \mathbb{E}_{z \sim q_\phi(z | r_i, c)} \left[ \frac{p_\theta(r_i, z | c, r_{-i})}{q_\phi(z | r_i, c)} \right] \right] &\geq \\ \mathbb{E}_{c \sim q_\phi(c | r_{1:N})} \left[ \sum_{i=1}^N \log \mathbb{E}_{z \sim q_\phi(z | r_i, c)} \left[ \frac{p_\theta(r_i, z | c, r_{-i})}{q_\phi(z | r_i, c)} \right] \right] - \mathbb{D}_{\text{KL}} [q_\phi(c | r_{1:N}) || p_\theta(c)] &\geq \\ \mathbb{E}_{c \sim q_\phi(c | r_{1:N})} \left[ \sum_{i=1}^N \mathbb{E}_{z \sim q_\phi(z | r_i, c)} \left[ \log \frac{p_\theta(r_i, z | c, r_{-i})}{q_\phi(z | r_i, c)} \right] \right] - \mathbb{D}_{\text{KL}} [q_\phi(c | r_{1:N}) || p_\theta(c)] &= \\ \mathbb{E}_{c \sim q_\phi(c | r_{1:N})} \left[ \sum_{i=1}^N \mathbb{E}_{z \sim q_\phi(z | r_i, c)} [\log p_\theta(r_i | z, r_{-i})] - \sum_{i=1}^N \mathbb{D}_{\text{KL}} [q_\phi(z | i) || p_\theta(z | c)] \right] - \mathbb{D}_{\text{KL}} [q_\phi(c | r_{1:N}) || p_\theta(c)] & \quad (\text{D.4}) \end{aligned}$$

As standard with variational inference, instead of maximizing the log-likelihood directly, we maximize its lower bound. The first term in Eq. D.4 is the *reconstruction error*: it encourages the quality reconstruction of the reviews from latent codes. Unfortunately, the exact computation of the term is intractable. In this light, we leverage *the re-parametrization trick* (Kingma & Welling, 2013) and Monte Carlo estimation. See Sec. 2.5.1 for more details about the method.





**Figure 4.2:** Production of the latent code  $z_N$  for review  $r_N$ .

The other two terms in Eq. D.4 are regularizers – Kullback-Leibler divergences (KLDs). They control the amount of information encoded in the latent representation by penalizing the deviation of the estimated posteriors from the corresponding priors. Due to the fact that both the posteriors and priors are Gaussian distributions, the KLD terms are available in the closed form. Lastly, the bound is maximized with respect to both the generative model’s parameters  $\theta$  and inference networks’ parameters  $\phi$ .

The inference network predicting the posterior for a review-specific variable  $q_\phi(z_i|r_i, c)$  is needed only in training and is discarded afterwards. In contrast, we will exploit the inference network  $q_\phi(c|r_{1:N})$  when generating summaries, as discussed in Section 4.5.

## 4.4 Design

### 4.4.1 Text Representations

A GRU encoder (Cho et al., 2014) embeds review words  $w$  to obtain hidden states  $h$ . Those representations are reused across the system, e.g., in the inference networks and the decoder. The full architecture used to produce the latent codes  $c$  and  $z_i$  is shown in Fig. 4.2. We make Gaussian assumptions for all distributions (i.e. posteriors and priors). As in Kingma and Welling (2013), we use separate linear projections (LPs) to compute the means and diagonal log-covariances.

#### 4.4.2 Prior $p(c)$ and posterior $q_\phi(c|r_{1:N})$

We set the prior over group latent codes to the standard normal distribution,  $p(c) = \mathbb{N}(c; 0, I)$ . In order to compute the approximate posterior  $q_\phi(c|r_{1:N})$ , we first predict the contribution (‘importance’) of each word in each review  $\alpha_i^t$  to the code of the group:

$$\alpha_i^t = \frac{\exp(f_\phi^\alpha(m_i^t))}{\sum_{j=1}^N \sum_k^{T_j} \exp(f_\phi^\alpha(m_j^k))},$$

where  $T_i$  is the length of  $r_i$  and  $f_\phi^\alpha$  is a feed-forward neural network (FFNN)<sup>4</sup> which takes as input concatenated word embeddings and hidden states of the GRU encoder,  $m_i^t = [h_i^t \circ w_i^t]$ , and returns a scalar. Next, we compute the intermediate representation with the weighted sum:  $\hat{h} = \sum_{i=1}^N \sum_t^{T_i} \alpha_i^t m_i^t$ .

Finally, we compute the Gaussian’s parameters using the affine projections:

$$\begin{aligned} \mu_\phi(r_{1:N}) &= \hat{L}\hat{h} + b_L \\ \log \sigma_\phi(r_{1:N}) &= \hat{G}\hat{h} + b_G \end{aligned}$$

#### 4.4.3 Prior $p_\theta(z_i|c)$ and posterior $q_\phi(z_i|r_i, c)$

To compute the prior on the review code  $z_i$ ,  $p_\theta(z_i|c) = \mathbb{N}(z_i; \mu_\theta(c), I\sigma_\theta(c))$ , we linearly project the product code  $c$ . Similarly, to compute the parameters of the approximate posterior  $q_\phi(z_i|r_i, c) = \mathbb{N}(z_i; \mu_\phi(r_i, c), I\sigma_\phi(r_i, c))$ , we concatenate the last encoder’s state  $h_i^{T_i}$  of the review  $r_i$  and  $c$ , and perform affine transformations.

#### 4.4.4 Decoder $p_\theta(r_i|z_i, r_{-i})$

To compute the distribution  $p_\theta(r_i|z_i, r_{-i})$ , we use an auto-regressive GRU decoder with the attention mechanism (Bahdanau et al., 2015) and a pointer-generator network.

We compute the context vector  $c_i^t = \text{att}(s_i^t, h_{-i})$  by attending to all the encoder’s hidden states  $h_{-i}$  of the other reviews  $r_{-i}$  of the group, where the decoder’s hidden state  $s_i^t$  is used as a query. The hidden state of the decoder is computed using the GRU cell as

$$s_i^t = \text{GRU}_\theta(s_i^{t-1}, [w_i^t \circ c_i^{t-1} \circ z_i]). \quad (\text{D.5})$$

The cell inputs the previous hidden state  $s_i^{t-1}$ , as well as concatenated word embedding  $w_i^t$ , context vector  $c_i^{t-1}$ , and latent code  $z_i$ .

4. We use FFNNs with the tanh non-linearity in several model components. Whenever a FFNN is mentioned in the subsequent discussion, this architecture is assumed.

Finally, we compute the word distributions using the pointer-generator network:

$$p_{\theta}(r_i|z_i, r_{-i}) = \prod_{t=1}^T g_{\theta}(r_i^t | s_i^t, c_i^t, w_i^t, r_{-i}) \quad (\text{D.6})$$

The pointer-generator network computes two internal word distributions that are hierarchically aggregated into one distribution (Morin & Bengio, 2005). One distribution assigns probabilities to words being generated using a fixed vocabulary, and another one probabilities to be copied directly from the other reviews  $r_{-i}$ . In our case, the network helps to preserve details and, especially, to generate rare tokens.

## 4.5 Summary Generation

Given reviews  $r_{1:N}$ , we generate a summary that reflects common information using trained components of the model. Formally, we could sample a new review from

$$p_{\theta}(r|r_{1:N}) = \mathbb{E}_{c \sim q_{\phi}(c|r_{1:N})} \left[ \mathbb{E}_{z \sim p_{\theta}(z|c)} [p_{\theta}(r|z, r_{1:N})] \right].$$

As we argued in the introduction and will revisit in experiments, a summary or summarizing review, should be generated relying on the mean of the reviews' latent code. Consequently, instead of sampling  $z$  from  $p_{\theta}(z|c) = \mathbb{N}(z; \mu_{\theta}(c), I\sigma_{\theta}(c))$ , we set it to  $\mu_{\theta}(c)$ . We also found beneficial, in terms of evaluation metrics, not to sample  $c$  but instead to rely on the mean predicted by the inference network  $q_{\phi}(c|r_{1:N})$ .

## 4.6 Experimental Setup

### 4.6.1 Datasets

Our experiments were conducted on business customer reviews from the Yelp Dataset Challenge and Amazon product reviews (He & McAuley, 2016). These were pre-processed similarly to Chu and Liu (2019), and the corresponding data statistics are shown in Table 4.2. We selected only businesses and products with a minimum of 10 reviews, and the minimum and maximum length of 20 and 70 words respectively, popular groups above the 90<sup>th</sup> percentile were removed. And each group was set to contain 8 reviews during training. From the Amazon dataset we selected 4 categories: *Electronics*, *Clothing*, *Shoes and Jewelry*, *Home and Kitchen*, *Health and Personal Care*.

Dataset	Training	Validation
Yelp	38,776/1,012,280	4,311/113,373
Amazon	183,103/4,566,519	9,639/240,819

**Table 4.2:** Data statistics after pre-processing. The format in the cells is Businesses/Reviews and Products/Reviews for Yelp and Amazon, respectively.

These datasets present different challenges to abstractive summarization systems. Yelp reviews contain much personal information and irrelevant details which one may find unnecessary in a summary. Our summarizer, therefore, needs to distill important information in reviews while abstracting away from details such as a listing of all items on the menu, or mentions of specific dates or occasions upon which customers visited a restaurant. On the contrary, in Amazon reviews, we observed that users tend to provide more objective information and specific details that are useful for decision making (e.g., the version of an electronic product, its battery life, its dimensions). In this case, it would be desirable for our summarizer to preserve this information in the output summary.

For evaluation, we used the same 100 human-created Yelp summaries released by Chu and Liu (2019). These were generated by Amazon Mechanical Turk (AMT) workers, who summarized 8 input reviews. We created a new test for Amazon reviews following a similar procedure (see Appendix A.3 for details). We sampled 60 products and 8 reviews for each product, and they were shown to AMT workers who were asked to write a summary. We collected three summaries per product, 28 products were used for development and 32 for testing.

#### 4.6.2 Experimental Details

For sequential encoding and decoding, we used GRUs (Cho et al., 2014) with 600-dimensional hidden states. The word embeddings dimension was set to 200, and they were shared across the model (Press & Wolf, 2017). The vocabulary size was set to 50,000 most frequent words, and an extra 30,000 were allowed in the extended vocabulary, the words were lower-cased. We used the Moses’ (Koehn et al., 2007) reversible tokenizer and truecaser. Xavier uniform initialization (Glorot & Bengio, 2010) of 2D weights was used, and 1D weights were initialized with the scaled normal noise ( $\sigma = 0.1$ ). We used the Adam optimizer (Kingma & Ba, 2014), and set the learning rate to 0.0008 and 0.0001 on Yelp and Amazon, respectively. For summary decoding, we used length-normalized beam search of size 5, and relied on latent code means. In order to overcome ‘posterior collapse’ (Bowman et al., 2016) we applied cycling annealing (Fu et al., 2019) with  $r = 0.8$  for both the  $z$  and  $c$  related KL terms, with a new cycle over approximately every 2 epochs over the training set. The maximum annealing scalar was set to 1 for  $z$ -related KL term in on both datasets, and 0.3 and 0.65 for  $c$ -related KL-term on Yelp and Amazon, respectively. The reported ROUGE scores are based on F1.

The dimensions of the variables  $c$  and  $z$  were set to 600, and the  $c$  posterior’s scoring neural network had a 300-dimensional hidden layer and the  $\tanh$  non-linearity.

The decoder’s attention mechanism used a single layer neural network with a 200-dimensional hidden layer, and the  $\tanh$  non-linearity. The copy gate in the pointer-generator network was computed with a 100-dimensional single-hidden layer network, with the same non-linearity.

### 4.6.3 Baseline Models

**OPINOSIS** is a graph-based abstractive summarizer (Ganesan et al., 2010) designed to generate short opinions based on highly redundant texts. Although it is referred to as abstractive, it can only select words from the reviews.

**LEXRANK** is an unsupervised algorithm which selects sentences to appear in the summary based on graph centrality (sentences represent nodes in a graph whose edges have weights denoting similarity computed with tf-idf). A node’s centrality can be measured by running a ranking algorithm such as PageRank (Page et al., 1999).

**MEANSUM** is an unsupervised abstractive summarization model (Chu & Liu, 2019) which treats a summary as a structured latent state of an auto-encoder trained to reconstruct reviews of a product.

**TEXT VAE** we also trained a vanilla text VAE model (Bowman et al., 2016) with our GRU encoder and decoder. When generating a summary for  $r_1, \dots, r_N$ , we averaged the means of  $q_\phi(z_i | r_i)$ .

Finally, we used a number of simple summarization baselines. We computed the *clustroid* review for each group as follows. We took each review from a group and computed ROUGE-L with respect to all other reviews. The review with the highest ROUGE score was selected as the clustroid review. Furthermore, we sampled a *random* review from each group as the summary, and constructed the summary by selecting the *leading sentences* from each review of a group.

Additionally, as an upper bound, we report the performance of an *oracle* review, i.e., the highest-scoring review in a group when computing ROUGE-L against reference summaries.

### 4.6.4 Automatic Evaluation

As can be seen in Table 4.3, our model, COPYCAT, yields the highest scores on both Yelp and Amazon datasets. We observe large gains over the vanilla VAE. We conjecture that the vanilla VAE struggles to properly represent the variety of categories under a single prior  $p(z)$ . For example, reviews about a sweater can result in a summary about socks (see example summaries in Appendix). This contrasts with our model which allows each group to have its own prior  $p_\theta(z|c)$  and access to other reviews during decoding. The gains are especially large on the Amazon dataset, which is very broad in terms of product

	Amazon			Yelp		
	R1	R2	RL	R1	R2	RL
ORACLE	33.98	7.88	21.60	29.07	5.27	18.63
CLUSTROID	29.28	4.41	17.78	26.28	3.48	15.36
RANDOM	27.66	4.72	16.95	23.04	2.44	13.44
LEAD	30.32	<b>5.90</b>	15.78	26.34	3.72	13.86
LEXRANK	28.74	5.47	16.75	25.01	3.62	14.67
OPINOSIS	28.42	4.57	15.50	24.88	2.78	14.09
VAE	22.87	2.75	14.46	25.42	3.11	15.04
MEANSUM	29.20	4.70	18.15	28.46	3.66	15.57
COPYCAT	<b>31.97</b>	<b>5.81</b>	<b>20.16</b>	<b>29.47</b>	<b>5.26</b>	<b>18.09</b>

**Table 4.3:** ROUGE scores on the Amazon and Yelp test sets.

	Fluency	Coherence	Non Redundancy	Opinion Consensus	Overall
GOLD	0.6486	0.8140	0.6667	0.3750	0.8085
LEXRANK	-0.7662	-0.8293	-0.7699	<b>0.3500</b>	-0.5278
MEANSUM	-0.5294	-0.4857	0.0270	-0.6235	-0.7468
COPYCAT	<b>0.5802</b>	<b>0.5161</b>	<b>0.4722</b>	-0.0909	<b>0.3818</b>

**Table 4.4:** Human evaluation results in terms of the Best-Worst scaling on the Yelp test set.

categories. Our model also substantially outperforms MEANSUM. As we will confirm in human evaluation, MEANSUM’s summaries are relatively fluent at the sentence level but often contain hallucinations, i.e., information not present in input reviews. We provide an example summary produced by COPYCAT and MEANSUM in Tables 4.9 and 4.10.

## 4.7 Human Evaluation

### 4.7.1 Best-Worst Scaling

We performed human evaluation using the AMT platform. We sampled 50 businesses from the human-annotated Yelp test set and used all 32 test products from the Amazon set. We recruited 3 workers to evaluate each tuple containing summaries from MEANSUM, our model, LEXRANK, and human annotators. The reviews and summaries were presented to the workers in random order and were judged using Best-Worst Scaling (Louviere et al., 2015; Louviere & Woodworth, 1991). BWS has been shown to produce more reliable results than ranking scale (Kiritchenko & Mohammad, 2016a). Crowdworkers were asked to judge summaries according to the criteria listed below (we show an abridged version below, the full set of instructions is given in Appendix A.2). The non-redundancy and coherence criteria were taken from Dang (2005).

	Fluency	Coherence	Non Redundancy	Opinion Consensus	Overall
GOLD	0.3968	0.7097	0.7460	0.6207	0.7231
LEXRANK	-0.2963	-0.3208	-0.3962	<b>0.4348</b>	<b>0.1064</b>
MEANSUM	-0.6410	-0.8667	-0.6923	-0.7736	-0.8305
COPYCAT	<b>0.4444</b>	<b>0.3750</b>	<b>0.0270</b>	-0.4286	-0.1429

**Table 4.5:** Human evaluation results in terms of the Best-Worst scaling on the Amazon dataset test set.

*Fluency*: the summary sentences should be grammatically correct, easy to read and understand; *Coherence*: the summary should be well structured and well organized; *Non-redundancy*: there should be no unnecessary repetition in the summary; *Opinion consensus*: the summary should reflect common opinions expressed in the reviews; *Overall*: based on your own criteria (judgment) please select the best and the worst summary of the reviews.

For every criterion, a system’s score is computed as the percentage of times it was selected as best minus the percentage of times it was selected as worst (Orme, 2009). The scores range from -1 (unanimously worst) to +1 (unanimously best).

On Yelp, as shown in Table 4.4, our model scores higher than the other models according to most criteria, including overall quality. The differences with other systems are statistically significant for all the criteria at  $p < 0.01$ , using post-hoc HD Tukey tests. The difference in fluency between our system and gold summaries is not statistically significant.

The results on Amazon are shown in Table 4.5. Our system outperforms other methods in terms of fluency, coherence, and non-redundancy. As with Yelp, it trails LEXRANK according to the opinion consensus criterion. Additionally, LEXRANK is slightly preferable overall. All pairwise differences between our model and comparison systems are statistically significant at  $p < 0.05$ .

*Opinion consensus* (OC) is a criterion that captures the coverage of common opinions, and it seems to play a different role in the two datasets. On Yelp, LEXRANK has better coverage compared to our model, as indicated by the higher OC score, but is not preferred *overall*. In contrast, on Amazon, while the OC score is on the same par, LEXRANK is preferred *overall*. We suspect that presenting a breadth of exact details on Amazon is more important than on Yelp. Moreover, LEXRANK tends to produce summaries that are about 20 tokens longer than ours resulting in better coverage of input details.

	Amazon			Yelp		
	Full $\uparrow$	Partial $\uparrow$	None $\downarrow$	Full $\uparrow$	Partial $\uparrow$	None $\downarrow$
MEANSUM	24.41	31.23	44.36	28.41	30.66	40.92
COPYCAT	<b>38.23</b>	<b>33.95</b>	<b>27.83</b>	<b>44.50</b>	<b>32.48</b>	<b>23.01</b>

**Table 4.6:** Content support on the Amazon and Yelp test sets. Percentages are computed by normalizing sentence votes.

	R1	R2	RL
w/o $r_i$	28.66	4.54	18.63
w/o $c$	27.67	5.07	19.19
w/o $z$	29.26	4.16	17.39
Sampling	25.63	4.34	17.16
Full	31.97	5.81	20.16

**Table 4.7:** Ablations, ROUGE scores on the Amazon test set.

## 4.7.2 Content Support

The ROUGE metric relies on unweighted n-gram overlap and can be insensitive to hallucinating facts and entities (Falke et al., 2019). For example, referring to a burger joint as a veggie restaurant is highly problematic from a user perspective but yields only marginal differences in ROUGE. To investigate how well the content of the summaries is supported by the input reviews, we performed a second study. We used the same sets as in the human evaluation in Section 4.7.1, and split MEANSUM and our system’s summaries into sentences. Then, for each summary sentence, we assigned 3 AMT workers to assess how well the sentence is supported by the reviews. Workers were advised to read the reviews and rate sentences using one of the following three options. *Full support*: all the content is reflected in the reviews; *Partial support*: only some content is reflected in the reviews; *No support*: content is not reflected in the reviews.

The results in Table 4.6 indicate that our model is better at preserving information than MEANSUM by generating fewer hallucinations. We observed that this is especially prominent for rare words, such as brand names.

## 4.8 Analysis

### 4.8.1 Ablations

To investigate the importance of the model’s individual components, we performed ablations by removing the latent variables ( $z_i$  and  $c$ , one at a time), and attention over the other reviews. The models were re-trained on the Amazon dataset. The results are shown in Table 4.7. They indicate that all components play a role, yet the most significant drop in ROUGE was achieved



when the variable  $z$  was removed, and only  $c$  remained. Summaries obtained from the latter system were wordier and looked more similar to reviews. Dropping the attention (w/o  $r_i$ ) results in more generic summaries as the model cannot copy details from the input. Finally, the smallest quality drop in terms of ROUGE-L was observed when the variable  $c$  was removed.

In the introduction, we hypothesized that using the mean of latent variables would result in more “grounded” summaries reflecting the content of the input reviews, whereas sampling would yield texts with many novel and potentially irrelevant details. To empirically test this hypothesis, we sampled the latent variables during summary generation, as opposed to using mean values (see Sec. 4.5). We indeed observed that the summaries were wordier, less fluent, and less aligned to the input reviews, as is also reflected in the ROUGE scores (Table 4.7).

#### 4.8.2 Copy Mechanism

Finally, we analyzed which words are copied by the full model during summary generation. Generally, the model copies around 3-4 tokens per summary. We observed a tendency to copy product-type specific words (e.g., *shoes*) as well as brands and names.

#### 4.8.3 Latent Codes

mean $z$	Bought this for my Kindle Fire HD and it works great. I have had no problems with it. I would recommend it to anyone looking for a good quality cable.
$z_1$	Works fine with my Kindle Fire HD 8.9". The picture quality is very good, but it doesn't work as well as the picture. I'm not sure how long it will last, but i am very disappointed.
$z_2$	This is a great product. I bought it to use with my Kindle Fire HD and it works great. I would recommend it to anyone who is looking for a good quality cable for the price.
$z_3$	Good product, does what it is supposed to do. I would recommend it to anyone looking for a HDMI cable.
Reviews	Love this HDMI cable , but it only works with HD Kindle and not the HDX Kindle which makes me kinda crazy . I have both kinds of Kindles but the HDX is newer and I can t get a cable for the new one . I guess my HD Kindle will be my Amazon Prime Kindle . It works great ! </s> I got a kindle for Christmas . I had no idea how to work one etc . I discovered you can stream movies to your tv and this is the exact cable for it . Works great and seems like its good quality . A bit long though. </s> this is great for watching movies from kindle to tv . Now the whole family can enjoy rather than one person at a time . Picture quality isn't amazing , but it s good . </s> I just received this wire in the mail , and it does not work in the slightest . I am very displeased with this product . </s> Works great ! ! Now I can watch Netflix on my TV with my Kindle Fire HD ... I love it and so will you ! </s> Works awesome . Great item for the price. Got it very quickly . Was as described in the ad. Exactly what I was looking for. </s> I plugged it into my Kindle fire HD and into the TV and works perfectly . Have had no problems with it ! </s> This is just what I was looking for to connect my Kindle Fire to view on our TV ! Great price too!

**Table 4.8:** Amazon summaries of the full model with sampled and mean assignment to  $z$ . The assignment to  $c$  was fixed, and was the mean value based on the approximate posterior  $q_\phi(c|r_1, \dots, r_N)$ . Reviews are separated by ‘</s>’.

We performed a qualitative analysis of the latent variable  $z$  to shed additional light on what it stores and sensitivity of the decoder with respect to its input. Specifically, we computed the mean value for the variable  $c$  using the approximate posterior  $q_\phi(c|r_1, \dots, r_N)$ , and then sampled  $z$  from the prior  $p_\theta(z|c)$ .

First, we observed that the summaries produced using the mean of  $z$  are more fluent. For example, in Table 4.8, the  $z_1$  based summary states: ‘The picture quality is very good, but it doesn’t work aswell as the picture.’, where the second phrase could be rewritten in a more fluent matter. Also, we found that mean based summaries contain less details that are partially or not supported by the reviews. For example, in the table,  $z_1$  based summary mentions Kindle Fire HD 8.9’, while the dimension is never mentioned in the reviews. Finally, different samples were observed to result in texts that contain different details about the reviews. For example,  $z_1$  sample results in the summary that captures the picture quality, while  $z_3$  that the item is good for its price. Overall, we observed that the latent variable  $z$  stores content based information, that results in syntactically diverse texts, yet reflecting information about the same businesses or product.

#### 4.8.4 Repetitions

We observed an increase in the amount of generated repetitions both in the reconstructed reviews and summaries when the  $z$ -related KL term is low and beam search is used. Intuitively, the initial input to the decoder becomes less informative, and it starts relying on learned local statistics to perform reconstruction. When the  $z$ -related KLD vanishes to zero (see Eq. D.4), the decoder essentially becomes a unconditional language model, for which beam search was shown to lead to generation of repetitions (Holtzman, Buys, Forbes, & Choi, 2019).

#### 4.8.5 Review-like Fragments

As the model is trained solely on customer reviews, it can, unsurprisingly, generate review-like fragments. These fragments can be uninformative and written in the informal (review-like) writing style. For example in Table 4.9, the model generates: ‘*I would recommend it to anyone who wants to protect their laptop*’. This is a common concluding phrase in customer reviews. Content-wise, it is suitable for a summary, however, its writing style needs to be addressed. If we could retain the same meaning but modify the style, we would prefer a sentence like: ‘*This case is highly recommended for laptop protection*’. As we show in Chapter 5, this problem has a very simple solution.

## 4.9 Conclusions

In this chapter, we presented an abstractive summarizer of opinions, which does not use any summaries in training and is trained on a large collection of reviews. The model has an explicit hierarchical latent structure. Specifically, we represent individual review and product semantics as continuous latent variables following Gaussian distributions. And the model is trained end-to-end using amortized variational inference. In automatic evaluation, the model outperforms competitors, especially the only other unsupervised abstractive multi-review summarizer – MEANSUM. Furthermore, the human evaluation of the generated summaries (by considering their alignment with the reviews) shows that those created by our model better reflect the content of the input reviews.

We also presented *leave-one-out* unsupervised objective, which is model agnostic. This objective allows us to construct a synthetic reviews-summary dataset, where a random review is used as a summary. In this way the model learn useful correlations between the reviews of the same product. As we will show in Chapters 5 and 6, this objective is also useful for model pre-training before fine-tuning on human-written summaries.

In the next chapter, we will focus on improving the summary quality by leveraging a handful of summaries. Specifically, we will focus on review-like fragments generated by COPYCAT (see Sec. 4.8.5 for discussion). Our proposed model will be simpler – no latent variables – and capitalize on few-shot learning. We will re-use the same objective for the model's pre-training discussed in Sec. 4.3.1. However, we will also leverage a handful of summaries to efficiently improve the summarizer.

## 4.10 Reflections

During work on COPYCAT, no unsupervised abstractive opinion summarizers were available. In this light, we proposed an unsupervised approach to train a latent model on customer reviews. Below we reflect on these two aspects – COPYCAT and the unsupervised leave-one-out objective.

For this model, we used two levels of latent variables – individual review semantics and product semantics. To increase the model flexibility, more levels can be added. For instance, we could assume that review semantics depend on review types. These types could correspond to mayor aspects or general review topics and be modelled by individual review Categorical variables. Following this intuition, an extended class-based COPYCAT was proposed in Nguyen, Shen, and Hovsepian (2021). The extended model can be conditioned on different classes and

generate summaries corresponding to different topics/aspects. In the same vein, one could consider a more fine-grained approach by assigning a topic to each review sentence as in RECURSUM (Isonuma, Mori, Bollegala, & Sakata, 2021a). The model utilizes a hierarchical tree structure of topics, making it possible to control the granularity of summary sentences.

On the model’s architecture side, we leveraged GRUs (Cho et al., 2014), which were often utilized in research and made our model comparable to MEANSUM. However, a more powerful modelling backbone is also available – Transformers (Vaswani et al., 2017). This architecture is more computationally efficient and better at modelling long-distance dependencies. Consequently, we used this architecture in our next work described in Chapter 5.

Similarly, to make COPYCAT comparable to MEANSUM, we trained it from scratch. However, an alternative would be to leverage a language model pre-trained on generic texts. Models, such as BERT (Devlin, Chang, Lee, & Toutanova, 2019), BART (Lewis, Liu, et al., 2020), and T5 (Raffel et al., 2019), have powerful language understanding and generation abilities useful for the task.

Despite a latent structure offering text generation controllability, we later found easier ways to assure desired text characteristics in generated summaries. Specifically, we found that inductive biases in terms of latent variables can be effectively replaced by fine-tuning a model on a handful of summaries. This few-shot perspective is attractive as it can reduce the model’s complexity and does not require a high budget for annotations. We will discuss this in Chapters 5 and 6.

The proposed unsupervised objective in Sec. 4.3.1 – *leave-one-out* – can be extended and re-utilized in a number of ways. First, the objective can be extended to a multimodal version as in Isonuma et al. (2021a) by also conditioning on product images. In (Elsahar, Coavoux, Rozen, & Gallé, 2021), the objective was extended by a more careful input reviews selection. Specifically, they leveraged cosine similarity over tf-idf bag-of-word representations thus producing pairs with better content alignment. Furthermore, the objective can be re-oriented to pre-training that precedes fine-tuning. Under this objective, the model can learn in-domain and task-related specifics as in Chapters 5 and 6. Lastly, the effectiveness of this objective in other multi-document summarization domains was shown in Vogler, Li, Xu, Mi, and Berg-Kirkpatrick (2022).

From a more global perspective, the work in this chapter opened an unsupervised opinion summarization avenue by providing insights, datasets, and evaluation methods. Consequently, this led to various models of the same type, which include RECURSUM (Im, Kim, Lee, Cho, & Chung, 2021), MULTIMODALSUM (Isonuma et al., 2021a), COOP (Iso, Wang, Suhara, Angelidis, & Tan, 2021), SELFSUM (Elsahar et al., 2021), PASS (Oved & Levy, 2021), to name a few.

Summaries	
COPYCAT	This is a great case for the Acer Aspire 14 laptop. It is a little snug for my laptop, but it's a nice case. I would recommend it to anyone who wants to protect their laptop.
MEANSUM	I was looking for a cover with a small case, and this one was a great price. I was looking for a case that is not too big and not too heavy. This one is great and I love it. I would recommend this product.
GOLD	This 14' laptop sleeve has an attractive design and is made of quality materials. It has a very snug fit, so be cautious of buying it if your laptop is even slightly over 14'. The only real issue is the rubbery smell that comes with it. Overall, I would recommend this, unless you have a sensitive nose.
Reviews	
Love the design of this sleeve. When I don't have a table I use the sleeve as a mat to protect my thighs from the heat of my laptop. Wish it had a bit more padding and cushion but other than that it's a pretty carrying sleeve.	
This was bought for our teen daughter for Christmas, the only negative things about it are 1.no strap 2.no pockets, other than that it is great, but in all fairness i knew both things about it when ordered, it came in very fast, was just as described, great item for the price.	
This case fits perfectly on my 14inch hp. I disagree with the comments that talk about the smell being terrible. The smell really wasn't that strong and went away after a couple of weeks. I love the design and it was a fair price. Shipping was a little expensive though. It arrived quickly and I would buy from this seller again.	
I really enjoy the design of this case but I thought it would be lighter. When it came, it was a very dark grey which was a bummer. But the quality is good and it has a nice thick padding to protect my laptop.	
I like this case very much! It fits my Acer Aspire 14 'as a glove!!! Yes, it smells a little bit but I wouldn't say that it's so bad as mentioned in some reviews. And it's very cute too. I recommend this product to anyone.	
I BOUGHT IT FOR MY ACER 14 ', IT IS A LITTLE BIT TIGHT, BUT THE WORST WAS IT HAS HEAVY RUBBER SMELL, WHEN I LEAVE THE CASE IN MY ROOM, THE RUBBER SMELL WAS FILLED WITH THE AIR EVERYWHERE OF MY ROOM, MADE ME SICK. DON'T BUY IT.	
I have a dell inspiron 14, and the cover is a little snug. It's working well though and now I can carry my laptop in my school backpack rather than carry a separate laptop bag. I'd recommend this product, just do a bunch of research to make sure it will fit.	
I really love it! It doesn't smell a lot as some reviews say and the draw is well defined.I bought this size for a 13.3 'because my laptop will have a cover on it so I don't know how it fits for a traditional 14 inch	

**Table 4.9:** Amazon summaries generated by COPYCAT and MEANSUM.

Summaries	
COPYCAT	This place is the best Mexican restaurant i have ever been to. The food was delicious and the staff was very friendly and helpful. Our server was very attentive and made sure we were taken care of. We'll be back for sure.
MEANSUM	A little on the pricey side but I was pleasantly surprised. We went there for a late lunch and it was packed with a great atmosphere, food was delicious and the staff was super friendly. Very friendly staff. We had the enchiladas with a few extra veggies and they were delicious! Will be back for sure!
GOLD	This place is simply amazing! Its the best Mexican spot in town. Their tacos are delicious and full of flavor. They also have chips and salsa that is to die for! The salsa is just delectable! It has a sweet, tangy flavor that you can't find anywhere else. I highly recommend!
Reviews	
Classic style Mexican food done nicely! Yummy crispy cheese crisp with a limey margarita will win my heart any day of the week! The classic frozen with a chambord float is my favorite and they do it well here. The salad carbon was off the chain- served on a big platter and worked for me as 2 full dinners.	
For delicious Mexican food in north Phoenix, try La Pinata. This was our visit here and we were so stunned by the speed in which our food was prepared that we were sure it was meant for another table. The food was hot and fresh and well within our budget. My husband got a beef chimichanga and I got bean and cheese burrito, which we both enjoyed. Chips and salsa arrived immediately; the salsa tastes sweeter than most and is equally flavorful. We will be back!	
Good food! Great atmosphere! Great patio. Staff was super friendly and accommodating! We will definately return!	
This place was very delicious! I got the ranchero burro and it was so good. The plate could feed at least two people. The staff was great and so nice! I also got the fried ice cream it was good. I would recommend this place to all my friends.	
We arrive for the first time, greeted immediately with a smile and seated promptly. Our server was fantastic, he was funny and fast. Gave great suggestions on the menu and we both were very pleased with the food, flavors, speed and accuracy of our orders. We will definitely be going back for more great food!	
Well was very disappointed to see out favorite ice cream parlor closed but delightfully surprised at how much we like this spot!!Service was FANTASTIC TOP notch!! Taco was great lots of cheese. Freshly deep fried shell not like SO MANY Phoenix mex restaurants use! Enchilada was very good. My wife really enjoyed her chimichanga. My moms chilli reanno was great too. Everything we had so far was great. We will return. Highly recommended.	
I'm only on the salsa and it's just as fabulous as always. I love the new location and the decor is beautiful. Open 5 days and the place is standing room only. To the previous negative commentor, they are way too busy to fill an order for beans. Go across the street....you'll be angry lol.	
I just tried to make a reservation for 15 people in March at 11 am on a Tuesday and was informed by a very rude female. She said "we do not take reservations" and I asked if they would for 15 people and she said " I told you we don't take reservations" and hung up on me. Is that the way you run a business? Very poor customer service and I have no intentions of ever coming there or recommending it to my friends.	

**Table 4.10:** Yelp summaries generated by COPYCAT and MEANSUM.

# Few-Shot Learning for Opinion Summarization

---

In Chapter 4, we discussed an unsupervised model – COPYCAT – trained solely on customer reviews. As the model is never exposed to human-written summaries, it sometimes generates summaries with review-like fragments. These fragments can be uninformative and written in a review-like style. In this chapter, we show how a handful of annotated samples can be utilized to improve an unsupervised opinion summarizer and thus validate the second hypothesis (see Sec. 1.3).

Some desired summary characteristics are hard to learn from customer reviews alone. For instance, reviews vary in terms of writing style but summaries are expected to be written in a formal writing style. Because the amount of human-written summaries is rather limited, the naive fine-tuning of the full model on a handful of summaries leads to rapid overfitting. Consequently, generated summaries by such a model poorly reflect the desired characteristics. To address this challenge, we propose a few-shot model – FEWSUM.

First, we define *properties* (Fan, Grangier, & Auli, 2018) that capture desired summary characteristics. These are computed automatically using a heuristic for any given pair of reviews-summary. Second, we pre-train the model on customer reviews using the leave-one-out objective where the decoder also receives property values for the pseudo summary. This exposes the model to a large variety of property values, some of which correspond to reviews that are like desired summaries. Third, we train a small module on gold reviews-summary pairs to produce property values leading to summaries. Lastly, this module is used in test time to generate summaries. In automatic and human evaluation we show that FEWSUM-generated summaries are substantially better than COPYCAT’s. Moreover, the model generalizes well in the cross-domain setting.

<b>Gold</b>	These shoes run <b>true to size</b> , <b>do a good job supporting the arch of the foot</b> and <b>are well-suited for exercise</b> . They're good looking, <b>comfortable</b> , and the sole feels soft and cushioned. Overall they are a nice, <b>light-weight pair of shoes</b> and come in a variety of stylish colors.
<b>FewSum</b>	These running shoes are great! They <b>fit true to size</b> and are <b>very comfortable to run around in</b> . They are <b>light weight</b> and <b>have great support</b> . They run <b>a little on the narrow side</b> , so make sure to <b>order a half size larger than normal</b> .
<b>Reviews</b>	<b>perfect fit for me ... supply the support that I need ... are flexible and comfortable ...    ... It is very comfortable ... I enjoy wearing them running ...    ... running shoes ... felt great right out of the box ... They run true to size ...    ... my feet and feel like a dream ... Totally light weight ...    ... shoes run small ... fit more true to size ... fit is great! ... supports my arch very well ...    ... They are lightweight... usually wear a size women's 10 ... ordered a 10.5 and the fit is great!</b>

**Table 5.1:** Example summaries produced by our system and an annotator; colors encode its alignment to the input reviews. The reviews are truncated, and delimited with the symbol ‘||’.

## 5.1 Introduction

The absence of large annotated resources is one of the main challenges in opinion summarization, as we discussed in Sec. 1.2.2. In this light, a number of unsupervised models were introduced, such as MEANSUM (Chu & Liu, 2019), and DENOISESUM (Amplayo & Lapata, 2020). In Chapter 4 we also discussed an unsupervised model – COPYCAT. However, unsurprisingly perhaps, since the models are not exposed to the actual summaries, they are unable to learn their key characteristics. For instance, MEANSUM is prone to producing summaries containing a significant amount of information unsupported by reviews; COPYCAT generates summaries that are better aligned with reviews, yet they are limited in detail. Moreover, both systems are trained mostly on subjectively written reviews and, as a result, tend to generate summaries in the same writing style.

To learn the expected summary characteristics, one could leverage small annotated datasets. However, unlike recent approaches to language model adaptation for abstractive single-document summarization (Hoang, Bosselut, Celikyilmaz, & Choi, 2019; Raffel et al., 2019) that utilize hundreds of thousands of summaries, our two annotated datasets consist of only 60 and 100 annotated data points. It was also observed that a naive fine-tuning of multi-million parameter models on small corpora leads to rapid over-fitting and poor generalization (Finn, Abbeel, & Levine, 2017; Vinyals, Blundell, Lillicrap, Wierstra, et al., 2016). In this light, we propose a few-shot learning framework and demonstrate that even a tiny number of annotated samples is sufficient to bootstrap generation of formal summary texts that are both informative and fluent (see Table 5.1). To the best of our knowledge, it is the first few-shot learning approach applied to summarization.



We observe that reviews in a large unannotated collection vary a lot. For example, they differ in style, the level of detail, or how much they diverge from other reviews of the product in terms of content and overall sentiment. We refer to individual review characteristics and their relations to other reviews as *properties* (Ficler & Goldberg, 2017). While reviews span a large range of property values, only a subset of them is appropriate for summaries. For example, summaries should be close to the product’s reviews in content, avoid using the first-person pronouns and agree with the reviews in sentiment. Our approach starts with estimating a property-aware model on a large collection of reviews and then adapts the model using a few annotator-created summaries, effectively switching the generator to the summarization regime. As we demonstrate in our experiments, the summaries do not even have to come from the same domain.

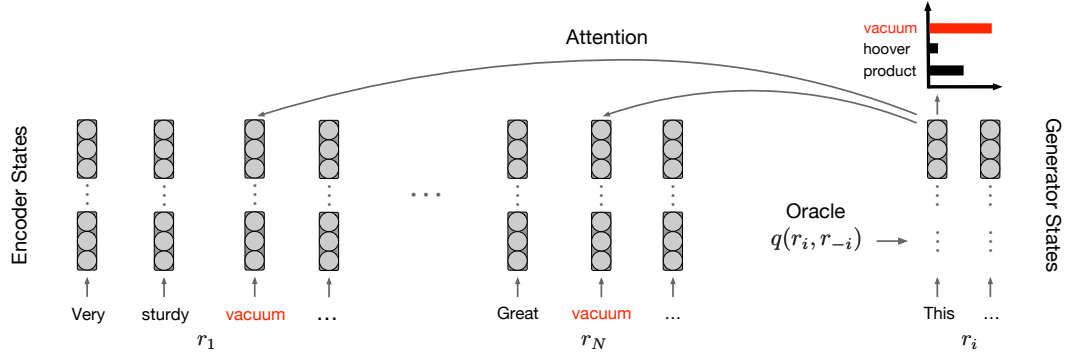
More formally, we estimate a text model on a dataset of reviews; the generator is a Transformer conditional language model (CLM) that is trained with a ‘leave-one-out’ objective (Besag, 1975; Bražinskas et al., 2020b) by attending to other reviews of the product. This objective is the same as for COPYCAT we discussed in Chapter 4. We define properties of unannotated data that are directly related to the end task of summarization. Those properties are easy to derive from reviews, and no extra annotation effort is required. The CLM is conditioned on these properties in training. The properties encode partial information about the target review that is being predicted. We capitalize on that by fine-tuning parts of the model jointly with a tiny *plug-in network* on a handful of human-written summaries. The plug-in network is trained to output property values that make the summaries likely under the trained CLM. The plug-in has less than the half a percent of the original model’s parameters, and thus is less prone to over-fitting on small datasets. Nevertheless, it can successfully learn to control dynamics of a large CLM by providing property values that force generation of summaries. We shall refer to the model produced using the procedure as **Few Shot Summarizer** (FEWSUM).

We evaluate our model against both extractive and abstractive methods on Amazon and Yelp human-created summaries. Summaries generated by our model are substantially better than those produced by competing methods, as measured by automatic and human evaluation metrics on both datasets. Finally, we show that it allows for successful cross-domain adaption. Our contributions can be summarized as follows:

- we introduce the first few-shot learning framework for abstractive opinion summarization;
- we demonstrate that the approach substantially outperforms extractive and abstractive models, both when measured with automatic metrics and in human evaluation;
- we release datasets with abstractive summaries for Amazon products and Yelp businesses.<sup>1</sup>

---

1. Both the code and datasets are available at: <https://github.com/abrazinskas/FewSum>



**Figure 5.1:** Illustration of the FEWSUM model that uses the leave-one-out objective. Here predictions of the target review  $r_i$  is performed by conditioning on the encoded source reviews  $r_{-i}$ . The generator attends the last encoder layer’s output to extract common information (in red). Additionally, the generator has partial information about  $r_i$  passed by the oracle  $q(r_i, r_{-i})$ .

## 5.2 Related Work

The most related unsupervised approach to FEWSUM is COPYCAT in Chapter 4. Unlike that model, we rely on a powerful generator to learn conditional spaces of text without hierarchical latent variables. Finally, in contract to MEANSUM (Chu & Liu, 2019), our model relies on inductive biases without explicitly modeling of summaries. A concurrent model DENOISESUM (Amplayo & Lapata, 2020) uses a syntactically generated dataset of source reviews to train a generator to denoise and distill common information. Another parallel work, OPINIONDIGEST (Suhara, Wang, Angelidis, & Tan, 2020a), considers controllable opinion aggregation and is a pipeline framework for abstractive summary generation. Our conditioning on text properties approach is similar to Fidler and Goldberg (2017), yet we rely on automatically derived properties that associate a target to source, and learn a separate module to generate their combinations. Moreover, their method has not been studied in the context of summarization.

## 5.3 Unsupervised Training

User reviews about an entity (e.g., a product) are naturally inter-dependent. For example, knowing that most reviews are negative about a product’s battery life, it becomes more likely that the next review will also be negative about it. To model inter-dependencies, yet to avoid intractabilities associated with undirected graphical models (Koller & Friedman, 2009), we use the leave-one-out setting (Besag, 1975; Bražinskas et al., 2020b). This objective was originally introduced for COPYCAT we discussed in Chapter 4.

Specifically, we assume access to a large corpus of user text reviews, which are arranged as  $M$  groups  $\{r_{1:N}\}_{j=1}^M$ , where  $r_{1:N}$  are reviews about a particular product that are arranged as a *target review*  $r_i$  and  $N-1$  *source reviews*  $r_{-i} = \{r_1, \dots, r_{i-1}, r_{i+1}, \dots, r_N\}$ . Our goal is to estimate the conditional distribution  $r_i|r_{-i}$  by optimizing the parameters  $\theta$  as shown in Eq. E.1.

$$\theta^* = \arg \max_{\theta} \frac{1}{MN} \sum_{j=1}^M \sum_{i=1}^N \log p_{\theta}(r_i^j | r_{-i}^j) = \arg \max_{\theta} \frac{1}{MN} \sum_{j=1}^M \sum_{i=1}^N \log G_{\theta}(r_i^j | E_{\theta}(r_{-i}^j)) \quad (\text{E.1})$$

Our model has an encoder-generator Transformer architecture (Vaswani et al., 2017), where the encoder  $E_{\theta}$  produces contextual representations of  $r_{-i}$  that are attended by the generator  $G_{\theta}$ , which in-turn is a conditional language model predicting the target review  $r_i$ , estimated using teacher-forcing (Williams & Zipser, 1989). An illustration is presented in Fig. 5.1.

The objective lets the model exploit common information across reviews, such as rare brand names or aspect mentions. For example, in Fig. 5.1, the generator can directly attend to the word *vacuum* in the source reviews to increase its prediction probability. Additionally, we condition on partial information about the target review  $r_i$  using an oracle  $q(r_i, r_{-i})$  as shown in Eq. E.2.

$$\frac{1}{MN} \sum_{j=1}^M \sum_{i=1}^N \log G_{\theta}(r_i^j | E_{\theta}(r_{-i}^j), q(r_i^j, r_{-i}^j)) \quad (\text{E.2})$$

We refer to this partial information as *properties* (Ficler & Goldberg, 2017), which correspond to text characteristics of  $r_i$  or relations between  $r_i$  and  $r_{-i}$ . For example, one such property can be the ROUGE score (Lin, 2004) between  $r_i$  and  $r_{-i}$ , which indicates the degree of overlap between  $r_i$  and  $r_{-i}$ . In Fig. 5.1, a high ROUGE value can signal to the generator to attend the word *vacuum* in the source reviews instead of predicting it based on language statistics. Intuitively, while the model observes a wide distribution of ROUGE scores during training on reviews, during summarization in test time we can achieve a high degree of input-output text overlap by setting the property to a high value. We considered properties that are listed below.

*Content Coverage*: ROUGE-1, ROUGE-2, and ROUGE-L F1 scores between  $r_i$  and  $r_{-i}$  signal to  $G_{\theta}$  how much to rely on syntactic information in  $r_{-i}$  during prediction of  $r_i$ . *Writing Style*: as a proxy for formal and informal writing styles, we compute pronoun counts, and create a distribution over three points of view. We also added an additional class for cases with no pronouns, see Appendix 5.8.4 for details and examples; *Rating Deviation*: we compute the difference between the  $r_i$ 's rating and the average  $r_{-i}$  rating; *Length Deviation*: we similarly compute the difference between the  $r_i$ 's length and the average length of  $r_{-i}$ , in terms of tokens.

## 5.4 Novelty Reduction

While summary and review generation are technically similar, there is an important difference that needs to be addressed. Reviews are often very diverse, so when a review is predicted, the generator often needs to predict content that is not present in source reviews. On the other hand, when a summary is predicted, its semantic content always matches the content of the source reviews. To address this discrepancy, in addition to using the ROUGE scores, as was explained previously, we introduce a *novelty reduction* technique, which is similar to label smoothing (Pereyra, Tucker, Chorowski, Kaiser, & Hinton, 2017).

Specifically, we add a regularization term  $\mathcal{L}$ , scaled by  $\lambda$ , that is applied to word distributions produced by the generator  $G_\theta$  as shown in Eq. E.3.

$$\frac{1}{MN} \sum_{j=1}^M \sum_{i=1}^N \left[ \log G_\theta(r_i^j | E_\theta(r_{-i}^j), q(r_i^j, r_{-i}^j)) - \lambda \mathcal{L}(G_\theta(r_i^j | E_\theta(r_{-i}^j), q(r_i^j, r_{-i}^j))) \right] \quad (\text{E.3})$$

It penalizes assigning the probability mass to words not appearing in  $r_{-i}$ , as shown in Eq. E.4, and thus steers towards generation of text that is more grounded in content of  $r_{-i}$ .

$$\mathcal{L}(G_\theta(r_i | r_{-i}, q(r_i, r_{-i}))) = \sum_{t=1}^T \sum_{w \notin V(r_{-i})} G_\theta(W_t = w | r_i^{1:t-1}, r_{-i}, q(r_i, r_{-i})) \quad (\text{E.4})$$

Here,  $T$  is the length of  $r_i$ , and the inner sum is over all words that do not appear in the word vocabulary of  $r_{-i}$ . Intuitively, in Fig. 5.1, the penalty could reduce the probability of the word *hoover* to be predicted as it does not appear in the source reviews.

## 5.5 Summary Adaptation

Once the unsupervised model is trained on reviews, our task is to adapt it to generation of summaries. Here, we assume access to a small number of annotator-written summaries  $(s^k, r_{1:N}^k)_{k=1}^K$  where  $s$  is a summary for  $r_{1:N}$  input reviews. As we will show in Sec. 5.8.1, naive fine-tuning of the unsupervised model on a handful of annotated data-points leads to poor generalization. Instead, we capitalize on the fact that the generator  $G_\theta$  has observed a wide range of property values associated with  $r_i$  during the unsupervised training phase. Intuitively, some combinations of property values drive it into generation of text that has qualities of a summary while others of a review. However, we might not know values in advance that are necessary for generation of summaries. Furthermore,  $q(r_i, r_{-i})$  cannot be applied at test time as it requires access to target texts. In the following section, we describe a solution that switches the generator to the summarization mode relying only on input reviews.

### 5.5.1 Plug-in Network

We start by introducing a parametrized *plug-in network*  $p_\phi(r_{-i})$  that yields the same types of properties as  $q(r_i, r_{-i})$ . From a practical perspective, the plug-in should be input-permutation invariant and allow for an arbitrary number of input reviews (Zaheer et al., 2017). Importantly, the trainable plug-in can have a marginal fraction of the main model's parameters, which makes it less prone to over-fitting when trained on small datasets. We initialize the parameters of  $p_\phi(r_{-i})$  by matching its output to  $q(r_i, r_{-i})$  on the unannotated reviews. Specifically, we used a weighted combination of distances as shown for one group of reviews in Eq. E.5.

$$\sum_{i=1}^N \sum_{l=1}^L \alpha^l D^l(p_\phi(r_{-i})^l, q(r_i, r_{-i})^l) \quad (\text{E.5})$$

Here,  $D^l(p_\phi(r_{-i})^l, q(r_i, r_{-i})^l)$  is a distance for the property  $l$ , and  $\alpha^l$  is an associated weight. Specifically, we used L1 norm for *Content Coverage*, *Rating and Length Deviations*, and Kullback-Leibler divergence for *Writing Style*.

For the plug-in network, we employed a multi-layer feed-forward network with multi-head attention modules over the encoded states of the source reviews at each layer, followed by a linear transformation, predicting property values. Note that the encoder is shared with the main model.

### 5.5.2 Fine-Tuning

Unsurprisingly, perhaps, the network  $p_\phi$  being initialized on unannotated reviews inherits a strong bias towards outputting property values resulting in generation of reviews, which should not be appropriate for generating summaries. Fortunately, due to the simplicity of the chosen properties, it is possible to fine-tune  $p_\phi$  to match the output of  $q$  on the annotated data  $(s^k, r_{1:N}^k)_{k=1}^K$  using Eq. E.5.

An alternative is to optimize the plug-in to directly increase the likelihood of summaries under  $G_\theta$  while keeping all other parameters fixed. We explored that option, and observed that it works similarly, yet leads to a slightly worse result.

As the generator is trained on unannotated reviews, it might not encounter a sufficient amount of text that is written as a summary, and that highly overlaps in content with the input reviews. We address that by unfreezing the attention module of  $G_\theta$  over input reviews and the plug-in  $p_\phi$ , and by maximizing the likelihood of summaries:

$$\frac{1}{K} \sum_{k=1}^K \left[ \log G_\theta(s^k | E_\theta(r_{1:N}^k), p_\phi(r_{1:N}^k)) \right] \quad (\text{E.6})$$

Dataset	Training	Validation
Yelp	38,913/1,016,347	4,324/113,886
Amazon	182,932/3,889,782	9,629/205,992

**Table 5.2:** Data statistics after pre-processing. The format in the cells is Businesses/Reviews and Products/Reviews for Yelp and Amazon, respectively.

This allows the system to learn an interaction between  $G_\theta$  and  $p_\phi$ . For example, what property values are better associated with summaries and how  $G_\theta$  should better respond to them.

## 5.6 Experimental Setup

### 5.6.1 Dataset

For training we used customer reviews from Amazon (He & McAuley, 2016) and Yelp.<sup>2</sup> From the Amazon reviews we selected 4 categories: *Electronics; Clothing, Shoes and Jewelry; Home and Kitchen; Health and Personal Care*. We selected only Amazon products and Yelp businesses with minimum of 10 reviews, and the minimum and maximum lengths of 20 and 70 words, respectively. Also, popular products/businesses above the 90<sup>th</sup> percentile were removed. From each business/product we sampled 9 reviews without replacement to form groups of reviews.

For training, we partitioned business/product reviews to the groups of 9 reviews by sampling without replacement. Thus, for unsupervised training in Sec. 5.3, we conditioned on 8 reviews for each target review. The data-statistics are shown in Table 5.2.

We obtained 480 human-written summaries (180 for Amazon and 300 for Yelp) for 8 reviews each, using Amazon Mechanical Turk (AMT). Each product/business received 3 summaries, and averaged ROUGE scores are reported in the following sections.

From the Amazon annotated dataset, We used 28, 12, 20 products for training, validation, and testing, respectively. On Yelp, we used 30, 30, 40 for training, validation, and testing, respectively. Both the automatic and human evaluation experiments were performed on the test sets.

2. <https://www.yelp.com/dataset/challenge>

### 5.6.2 Experimental Details

For the main model, we used the Transformer architecture (Vaswani et al., 2017) with trainable length embeddings and shared encoder-generator parameters (Raffel et al., 2019). Subwords were obtained with BPE (Sennrich et al., 2016) using 32000 merges. Subword embeddings were shared across the model as a form of regularization (Press & Wolf, 2017). For a fair comparison, we approximately matched the number of parameters to the abstractive baseline models MEANSUM (Chu & Liu, 2019) and COPYCAT (Chapter 4). We randomly initialized all parameters with Glorot (Glorot & Bengio, 2010). For the plug-in network used the Transformer stack architecture as a basis. We employed a multi-layer feed-forward with multi-head attention modules over source reviews at each layer. After the last layer, we performed a linear projection to compute property values. In terms of the number of parameters, the plug-in network had less than 0.5 % of the main model's. Further, parameter optimization was performed using ADAM (Kingma & Ba, 2014), and beam search with n-gram blocking was applied to our model and COPYCAT for summary generation. All experiments were conducted on 4 x GeForce RTX 2080 Ti.

### 5.6.3 Hyperparameters

Our parameter-shared encoder-generator model used a 8-head and 6-layer Transformer stack. Dropout in sub-layers and subword embeddings dropout was both set to 0.1, and we used 1000 dimensional position-wise feed-forward neural networks. We set subword and length embeddings to 390 and 10 respectively, and both were concatenated to be used as input. For the plug-in network, we set the output dimension to 30 and internal feed-forward network hidden dimensions to 20. We used a stack of 3 layers, and the attention modules with 3 heads at each layer. We applied 0.4 internal dropout and 0.15 attention dropout. Property values produced by the plug-in or oracle were concatenated with subword and length embeddings and linearly projected before being passed to the generator. In total, our model had approximately 25M parameters, while the plug-in network only 100K (i.e., less than 0.5 % of the main model's parameters). In all experiments, the hyperparameter tuning was performed based on the ROUGE-L score on Yelp and Amazon validation sets.

### 5.6.4 Training Procedure

First, to speed-up the training phase, we trained an unconditional language model for 13 epoch on the Amazon reviews with the learning rate (LR) set to  $5 * 10^{-4}$ . On Yelp we trained it for 27 epochs with LR set to  $7 * 10^{-4}$ . The language model was used to initialize both the encoder and generator of the main model.

Subsequently, we trained the model using Eq. E.2 for 9 epochs on the Amazon reviews with  $6 * 10^{-5}$  LR, and for 57 epochs with LR set to  $5 * 10^{-5}$ . Additionally, we reduced novelty using Eq. E.4 by training the model further for 1 epoch with  $10^{-5}$  LR and  $\lambda = 2$  on Amazon; On Yelp we trained for 4 epochs, with  $3 * 10^{-5}$  LR, and  $\lambda = 2.5$ .

For the plugin network's initialization, as explained in Sec. 5.5.1, we performed optimization by output matching with the oracle for 11 epochs on the unannotated Amazon reviews with  $1 * 10^{-5}$  LR. On Yelp we trained for 87 epochs with  $1 * 10^{-5}$ . Lastly, we fine-tuned the plugin network on the human-written summaries by output matching with the oracle<sup>3</sup>. On the Amazon data for 98 epochs with  $7 * 10^{-4}$ , and for 62 epochs with  $7 * 10^{-5}$  on Yelp. We set weights to 0.1, 1., 0.08, 0.5 for length deviation, rating deviation, POV, and ROUGE scores, respectively. Then fine-tuned the attention part of the model and the plug-in network jointly for 33 epochs with  $1 * 10^{-4}$  on the Amazon data. And 23 epochs with  $1 * 10^{-4}$  LR on Yelp.

### 5.6.5 Baselines

**LEXRANK** (Erkan & Radev, 2004) is an unsupervised extractive graph-based algorithm selecting sentences based on graph centrality. Sentences represent nodes in a graph whose edges have weights denoting similarity computed with tf-idf.

**MEANSUM** is an unsupervised abstractive summarization model (Chu & Liu, 2019) that treats a summary as a discrete latent state of an autoencoder. The model is trained in a multi-task fashion with two objectives, one for prediction of reviews and the other one for summary-reviews alignment in the semantic space using the cosine similarity.

**COPYCAT** is the state-of-the-art unsupervised abstractive summarizer that uses continuous latent representations to model review groups and individual review semantics. It has an implicit mechanism for novelty reduction and uses a copy mechanism. We discussed this model in Chapter 4.

As is common in the summarization literature, we also employed a number of simple summarization baselines. First, the **CLUSTROID** review was computed for each group of reviews as follows. We took each review from a group and computed ROUGE-L with respect to all other reviews. The review with the highest ROUGE score was selected as the clustroid review. Second, we sampled a **RANDOM** review from each group to be used as the summary. Third, we constructed the summary by selecting the *leading sentences* (**LEAD**) from each review of a group.

---

3. We set rating deviation to 0 as summaries do not have associated human-annotated ratings.



## 5.7 Evaluation Results

### 5.7.1 Automatic Evaluation

	Amazon			Yelp		
	R1	R2	RL	R1	R2	RL
LEXRANK	27.72	5.06	17.04	26.96	4.93	16.13
CLUSTROID	27.16	3.61	16.77	28.90	4.90	18.00
LEAD	27.00	4.92	14.95	26.20	4.57	14.32
RANDOM	25.00	3.82	15.72	21.48	2.59	13.87
MEANSUM	26.63	4.89	17.11	27.50	3.54	16.09
COPYCAT	27.85	4.77	18.86	28.12	5.89	18.32
FEWSUM	<b>33.56</b>	<b>7.16</b>	<b>21.49</b>	<b>37.29</b>	<b>9.92</b>	<b>22.76</b>

**Table 5.3:** ROUGE scores on the Amazon and Yelp test sets.

We report ROUGE F1 score (Lin, 2004) based evaluation results on the Amazon and Yelp test sets in Table 5.3, respectively. The results indicate that our model outperforms abstractive and extractive methods on both datasets. Also, the results are supported by qualitative improvements over other models, see example summaries in Tables 5.12 and 5.13.

### 5.7.2 Best-Worst Scaling

We performed human evaluation with the Best-Worst scaling (Kiritchenko & Mohammad, 2016a; Louviere et al., 2015; Louviere & Woodworth, 1991) on the Amazon and Yelp test sets using the AMT platform. We assigned multiple workers to each tuple containing summaries from COPYCAT, our model, LEXRANK, and human annotators. The judgment criteria were the following: *Fluency*, *Coherence*, *Non-redundancy*, *Informativeness*, *Sentiment*. Details are provided in Appendix B.1.

For every criterion, a system’s score is computed as the percentage of times it was selected as best, minus the percentage of times it was selected as worst (Orme, 2009). The scores range from -1 (unanimously worst) to +1 (unanimously best).

The results are presented in Tables 5.4 and 5.5 for Amazon and Yelp, respectively. On the Amazon data, they indicate that our model is preferred across the board over the baselines. COPYCAT is preferred over LEXRANK in terms of fluency and non-redundancy, yet it shows worse results in terms of informativeness and overall sentiment preservation. In the same vein, on Yelp in Table 5.5 our model outperforms the other models.

All pairwise differences between our model and other models are statistically significant at  $p < 0.05$ , using post-hoc HD Tukey tests. The only exception is non-redundancy on Yelp when comparing our model and COPYCAT (where our model shows a slightly lower score).

	Fluency	Coherence	Non-Redundancy	Informativeness	Sentiment
LEXRANK	-0.4848	-0.5161	-0.5862	-0.3488	-0.0909
COPYCAT	-0.1765	-0.5333	-0.2727	-0.7455	-0.7143
FEWSUM	<b>0.1000</b>	<b>0.1429</b>	<b>0.1250</b>	<b>0.2000</b>	<b>0.3061</b>
GOLD	0.5667	0.6364	0.6066	0.6944	0.4138

**Table 5.4:** Human evaluation results in terms of the Best-Worst scaling on the Amazon test set.

	Fluency	Coherence	Non-Redundancy	Informativeness	Sentiment
LEXRANK	-0.5588	-0.5312	-0.6393	-0.6552	-0.4769
COPYCAT	-0.2000	-0.0769	<b>0.1053</b>	-0.4386	-0.2857
FEWSUM	<b>0.1636</b>	<b>0.1429</b>	0.0000	<b>0.3793</b>	<b>0.3725</b>
GOLD	0.5278	0.3784	0.4795	0.6119	0.4118

**Table 5.5:** Human evaluation results in terms of the Best-Worst scaling on the Yelp test set.

### 5.7.3 Content Support

As was discussed in Chapter 4 and was observed by Falke et al. (2019); Tay et al. (2019), the ROUGE metric can be insensitive to hallucinating facts and entities. We also investigated how well generated text is supported by input reviews. We split summaries generated by our model and COPYCAT into sentences. Then for each summary sentence, we hired 3 AMT workers to judge how well content of the sentence is supported by the reviews. Three following options were available:

- *Full support*: all the content is reflected in the reviews;
- *Partial support*: only some content is reflected in the reviews;
- *No support*: content is not reflected in the reviews.

The results are presented in Table 5.6. Despite not using the copy mechanism, that is beneficial for fact preservation (Falke et al., 2019) and generation of more diverse and detailed summaries, we score on par with COPYCAT.

## 5.8 Analysis

### 5.8.1 Alternative Adaptation Strategies

We further explored alternative utilization approaches of annotated data-points, based on the same split of the Amazon summaries as explained in Sec. 5.6.1. First, we trained a model in an unsupervised learning setting (USL) on the Amazon reviews with the leave-one-out objective in Eq. E.1. In this setting, the model has neither exposure to summaries nor the properties, as the oracle  $q(r_i, r_{-i})$  is not used. Further, we considered two alternative settings how the pre-trained unsupervised model can be adapted on the gold summaries. In the first setting, the model is fine-tuned by predicting summaries conditioned on input reviews (USL+F). In

	Full $\uparrow$	Partial $\uparrow$	No $\downarrow$
COPYCAT	<b>46.15</b>	27.18	26.67
FEWSUM	43.09	<b>34.14</b>	<b>22.76</b>

**Table 5.6:** Content support on the Amazon test set in terms of percentages, which were computed by normalizing worker votes over sentences.

	R1	R2	RL
RANDOM	25.00	3.82	15.72
MTL	24.03	4.35	16.27
USL	21.45	3.15	15.23
USL+F	28.23	6.24	19.64
FEWSUM	33.56	7.16	21.49

**Table 5.7:** ROUGE scores on the Amazon test set for alternative summary adaptation strategies.

the second one, similar to Hoang et al. (2019), we performed adaptation in a multi-tasking learning (MTL) fashion. Here, USL is further trained on a mixture of unannotated corpus review and gold summary batches with a trainable embedding indicating the task.<sup>4</sup> The results are presented in Table 5.7.

First, we observed that USL generates summaries that get the worst ROUGE scores. Additionally, the generated text tends to be informal and substantially shorter than an average summary, we shall discuss that in Sec. 5.8.2.

Second, when the model is fine-tuned on the gold summaries (USL+F), it noticeably improves the results, yet they are substantially worse than of our proposed few-shot approach. It can be explained by strong influence of the unannotated data stored in millions of parameters that requires more annotated data-points to overrule. Finally, we observed that MTL fails to decouple the tasks, indicated by only a slight improvement over USL.

### 5.8.2 Influence of Unannotated Data

We further analyzed how plain fine-tuning on summaries differs from our approach in terms of capturing summary characteristics. For comparison, we used USL and USL+F, which are presented in Sec. 5.8.1. Additionally, we analyzed unannotated reviews from the Amazon training set. Specifically, we focused on text formality and the average word count difference (*Len*) from the gold summaries in the Amazon test set. As a proxy for the former, we computed the marginal distribution over points of view (POV), based on pronoun counts; an additional class (*NoPr*) was allocated to cases of no pronouns. The results are presented in Table 5.8.<sup>4</sup>

4. We observed that the 1:1 review-summary proportion works the best.

	1st	2nd	3rd	NoPr	Len
Reviews	49.0	7.3	35.6	8.1	-17.6
USL	56.7	0.0	43.3	0.0	-32.7
USL+F	29.7	0.0	45.3	25.0	-28.6
FEWSUM	0.5	1.3	83.2	15.0	3.4
GOLD	0.0	1.7	60.0	38.3	0.0

**Table 5.8:** Text characteristics of generated summaries by different models on the Amazon test set.

<b>USL</b>	This is my second pair of Reebok running shoes and I love them. They are the most comfortable shoes I have ever worn.
<b>USL+F</b>	This is my second pair of Reebok running shoes and they are the best running shoes I have ever owned. They are lightweight, comfortable, and provide great support for my feet.
<b>FewSum</b>	These running shoes are great! They fit true to size and are very comfortable to run around in. They are light weight and have great support. They run a little on the narrow side, so make sure to order a half size larger than normal.
<b>Gold</b>	These shoes run true to size, do a good job supporting the arch of the foot and are well-suited for exercise. They're good looking, comfortable, and the sole feels soft and cushioned. Overall they are a nice, light-weight pair of shoes and come in a variety of stylish colors.

**Table 5.9:** Example summaries produced by models with different adaptation approaches.

First, we observed that the training reviews are largely informal (49.0% and 7.3% for 1st and 2nd POV, respectively). Unsurprisingly, the model trained only on the reviews (USL) transfers a similar trait to the summaries that it generates.<sup>5</sup> On the contrary, the gold summaries are largely formal - indicated by a complete absence of the 1st and a marginal amount of 2nd POV pronouns. Also, an average review is substantially shorter than an average gold summary, and consequently, the generated summaries by USL are also shorter. Example summaries are presented in Table 5.9.

Further, we investigated how well USL+F, adapts to the summary characteristics by being actually fine-tuned on them. Indeed, we observed that USL+F starts to shift in the direction of the summaries by reducing the pronouns of the 1st POV and increasing the average summary length. Nevertheless, the gap is still wide, which would probably require more data to be bridged. Finally, we observed that our approach adapts much better to the desired characteristics by producing well-formed summary text that is also very close in length to the gold summaries.

5. As beam search, attempting to find the most likely candidate sequence, was utilized, opposed to a random sequence sampling, we observed that generated sequences had no cases of the 2nd POV pronouns and complete absence of pronouns (NoPr).

Domain	In-domain	Cross-domain
CLOTH	21.88	22.20
ELECTRONICS	21.46	21.36
HEALTH	21.21	19.09
HOME	21.39	22.50
AVG	21.49	21.29

**Table 5.10:** In and cross domain experiments on the Amazon dataset, ROUGE-L scores are reported.

1ST	I bought this as a gift for my husband. </s> I've been using Drakkar Noir Balm for over twenty years. </s> I purchased these for my son as a kind of a joke.
2ND	This is the best product you can buy! </s> You get what you pay for. </s> Please do yourself a favor and avoid this product.
3RD	This is his every work day scent. </s> It's very hard to buy the balm separately. </s> It smells like Drakkar, but it is hard to find
NO PRONOUNS	Very nice, not too overpowering. </s> This product has no smell what ever. </s> Nice to use for hardwood floors.

**Table 5.11:** Examples of review sentences that contain only pronouns belonging to a specific class. Sentences are separated by '</s>'.

### 5.8.3 Cross-Domain

We hypothesized that on a small dataset, the model primarily learns course-grained features, such as common writing phrases, and their correlations between input reviews and summaries. Also, that they, in principle, could be learned from remotely related domains. We investigated that by fine-tuning the model on summaries that are not in the target domain of the Amazon dataset. Specifically, we matched data-point count for 3/4 domains of training and validation sets to the in-domain Amazon data experiment presented in Sec 5.7; the test set remained the same for each domain as in the in-domain experiment. Then, we fine-tuned the same model 5 times with different seeds per target domain. For comparison, we used the in-domain model which was used in Amazon experiments in Sec. 5.7. We computed the average ROUGE-L score per target domain, where overall  $\sigma$  was 0.0137. The results are reported in Table 5.10.

The results indicate that the models perform on-par on most of the domains, supporting the hypothesis. On the other hand, the in-domain model shows substantially better results on the *health* domain, which is expected, as, intuitively, this domain is the most different from the rest.

#### 5.8.4 Points of View

Summaries differ from reviews in terms of the writing style. Specifically, reviews are predominantly written informally, populated by pronouns such as *I* and *you*. In contrast, summaries are desirable to be written formally. We found a surprisingly simple way to achieve that by conditioning the generator on the distribution over pronoun classes of the target review. We computed pronoun counts and produced the 4 class distributions: 1st, 2nd, 3rd person POV, and ‘other’ in case if no pronouns are present.

We illustrate how the writing style differs based on review sentences. Consider the example sentences in Table 5.11. Here one can observe that the sentences of different pronoun classes differ in the style of writing and often the intention of the message: 1st POV sentences tend to provide clues about the personal experience of the user. 2nd POV sentences, commonly convey recommendations to a reader. Finally, 3rd POV and ‘other’ sentences often describe aspects and their associated opinions.

### 5.9 Conclusions

In this chapter, we introduced the first to our knowledge few-shot framework for abstractive opinion summarization. We showed that it can efficiently utilize even a handful of annotated reviews-summary pairs to train models that generate fluent, informative, and overall sentiment reflecting summaries. From the technical perspective, we proposed to exploit summary related properties in unannotated reviews that are used for unsupervised training of the model. Then we trained a tiny plug-in network that learns to switch the model to the summarization regime. We demonstrated that our approach substantially outperforms competitive ones, both abstractive and extractive, in human and automatic evaluation. Finally, we showed that it also allows for successful cross-domain adaptation.

While properties often reflect human intuition and provide a strong signal about the target sequence, they require human-involved design and engineering. Also, we used a relatively small neural model, pre-trained on customer reviews from scratch. In the next chapter, we will use large pre-trained models with powerful language understanding and generation abilities acquired on generic texts. Further, we will explore a more efficient few-shot method that does not require human efforts.

## 5.10 Reflections

In this chapter we discussed two central components. First, few-shot framework for learning summarizers from a handful of annotated samples – FEWSUM. Second, a set of insights about customer reviews – their content structure, writing style, and how they differ from human-written summaries. These insights motivated properties we used in FEWSUM, and are essential for lexical features we will describe in Chapter 7. Moreover, another study in opinion summarization (Im et al., 2021) confirmed the effectiveness of the proposed properties.

We also believe that the model can be improved in future works in a number of ways. First, we could additionally leverage semantic features as properties to better capture correspondences between reviews and summaries. These could be pre-computed on the document or sentence level using a pre-trained language model (PLM), such as BERT (Devlin et al., 2019). Second, we could explore a different direction by completely removing properties and focus instead on better few-shot fine-tuning methods of pre-trained models. In Oved and Levy (2021), the authors explored this direction and demonstrated that a large PLM fine-tuned on gold summaries can outperform FEWSUM that was trained from scratch. In Chapter 6, we show that a PLM can be fine-tuned using an efficient method and improve the summary quality even further.

Summaries	
FEWSUM	This is a very nice table set for the price. It was easy to assemble and looks great in the kitchen. The only problem is that it is not sturdy enough to hold a lot of weight. It would be nice if it had a little more weight to it so that it would not tip over.
COPYCAT	This is a great table set for the price. It was easy to put together and looks great. The only thing is that the chairs are a little flimsy, but they are easy to assemble.
MEANSUM	The table was very easy to assemble and was easy to assemble. The only thing I would say is that the box is very small and not very sturdy. The table is very sturdy. I would recommend it to anyone looking for a sturdy table and to put on the wall.
GOLD	This is a perfect compact table that fits well in many places. The chairs are surprisingly very comfortable as well. It is cute and perfect for smaller living quarters and the best part is assembly is simple and straightforward.
Reviews	
It was easy to put together and looks great. However, when the item was shipped to me, one of the backs of the chairs was broken. I just fixed it myself with wood glue. Its not even visible now. The rest of it was in perfect condition.	
The table and chairs are very nice but not quite the color I expected (but I am getting used to it). Table and chairs delivered by the carrier right on time and with no damage. Very easy to assemble, but very difficult to get out of the box it was so well protected.	
This table was super easy to put together. The table and chairs are solid and sturdy! The seats are very comfortable. The table is the perfect size for our not so big kitchen. We are very pleased with this purchase.	
Moved to smaller living quarters and this just fits the bill. Color is perfect and it was easy to assemble. One fault to find is that the top scratches easily. It even came with a scratch. Other than that it is fine.	
I love my new dining room set. The set is very sturdy, the walnut finish is a nice color. This set is great for a small area, kitchen nook. Would not recommend for a large eating area. Table is small and so are the chairs. Yet strong enough to hold big boys and girls, thumps up, great price, packed well, arrived in a timely matter.	
It fits perfectly in the kitchen at the office. My staff assembled it without any delay. Everyone loves the dining set and they can't believe I ordered it on-line. I made the measurements and made sure of the dimensions of the room and the dining set and it's a perfect fit.	
I received this table and chairs completely damaged. The customer service experience with this company was terrible. In my opinion, this set is cheap and overpriced. It's not durable and not worth the money. Don't waste your time.	
The box looked like it had been opened, and then re-taped for resell. One of the chairs was broken, and the broken piece was nowhere close to the originating piece. Possibly other pieces damaged too, though didn't bother looking, instead just re-taped it back up to be sent back. I hope they don't just resell it to someone else.	

**Table 5.12:** Amazon summaries generated by FEWSUM, COPYCAT, and MEANSUM.



Summaries	
FEWSUM	This nail salon is very clean and the staff is very friendly. They have a wide variety of gel colors to choose from. The prices are very reasonable and they do a great job. The nail techs are very nice and do great work.
COPYCAT	This is the best nail salon I have ever been to. Everyone is very friendly and professional. My nails look great and I'm glad I did! I will definitely be coming back to this place.
MEANSUM	The owner is so nice and accommodating. I went to get my nails done by a friend, and I was extremely happy with this salon. Everyone was very friendly and I was able to use them for nails. They did a great job on my nails and the best part about it was that it was a busy day but it was a treat! Highly recommend them.
GOLD	It is very clean and nice inside. Everyone is so kind and friendly. They do an amazing job on both nails and pedis. They get it done with speed and precision with a price that is very much affordable. They have the best customer service.
Reviews	
Tim and Tami always always always have the best customer service and do the best nails. I will NEVER go anywhere else. Even after weeks my nails look and feel as good as they did when I first got them done! I'm so dedicated I recommend and bring in all my friends!	
Definitely my new nail salon! Everyone is so friendly and kind, I felt so welcomed! B did an amazing job on my nails. He made sure everything was perfect and happily changed something to make me happy. I would highly recommend this place to anyone who wants A + work at a totally affordable price. Love it!!:)	
Amazing service and nails. This is the second time I have been here, they did a perfect job again. They get it done fast yet with precision. Everyone is so friendly there too. Best nail salon I have ever been too. I'm glad I found it.	
I really enjoy coming here to get my nails done. They do a wonderful job on both pedis and nails. It is nice and clean inside. They are very friendly and welcoming. It is worth it to stop in and try it out.	
My first set of acrylics ever... I decided 27 years was a lot enough time to wait, and I'm SO happy with them. I'm not a huge nail person, and was glad to stumble upon this salon. My nail tech was quiet, clean, and very detail-oriented. Very pleased with my experience here and I recommend this place.	
I called to make an appointment for later today for 3 adults and 2 kids and the man who answered the phone said 'we only have 2 techs today' we can't do that. Poor customer service and I never even went in.	
Golden Nails has been my nail place for almost a year so it was surprising to see new management. However B did an AMAZING job on my coffin chrome nails and Nancy was extremely helpful figuring out how I wanted my nails done too. Definitely excited to keep coming back!	
Seriously the best service I have ever gotten at a Tempe nail salon!! I walked in and they helped me right away. Nancy helped me pick the perfect color and was very honest and up front about everything! I wanted something very natural and using the dip method, I love my nails!!	

**Table 5.13:** Yelp summaries generated by FEWSUM, COPYCAT, and MEANSUM.

# Efficient Few-Shot Fine-Tuning for Opinion Summarization

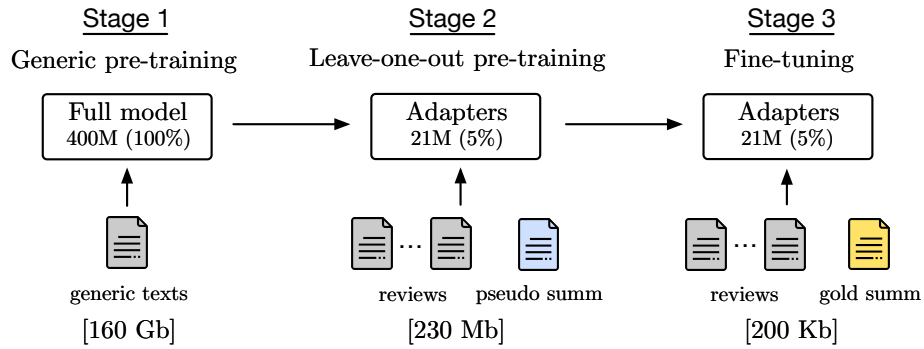
---

In Chapter 5, we discussed few-shot learning and the FEWSUM model specifically. In this chapter, we focus on efficiency in the few-shot learning setting. We make two main improvements over FEWSUM. First, we leverage a pre-trained model with powerful language understanding and generation abilities. Such a model can represent input texts as rich features and generate fluent texts from the start. Second, we replace hand-crafted properties with an efficient adaptation method – *adapters* (Houlsby et al., 2019). Compared to the standard fine-tuning, adapters are more robust to overfitting in low-resource settings (He et al., 2021) and computationally- and memory-efficient (Mahabadi et al., 2021).

Language models are pre-trained on large collections of generic texts and are often not accustomed to target domain specifics. These specifics, such as how users can use products, are hard to learn from a handful of annotated summaries in the fine-tuning phase. This leads to semantic mistakes in generated summaries, for example, *‘The hairdryer is great for water cooling.’* We address this problem by pre-training adapters – small neural networks inserted in Transformer layers – on a synthetic dataset. The dataset is created from customer reviews, where one review is sampled as a summary and others are selected as input. This effectively stores in-domain knowledge to adapters while preserving prior knowledge in the pre-training language model.

We also explore summary personalization, where a user submits a query based on his interests. A query consists of multiple aspects, such as ‘bluetooth,’ ‘price,’ ‘screen resolution,’ and a summarizer should generate a summary reflecting these aspects. As aspect-based summary datasets are not readily available, we propose a simple technique for converting generic datasets to aspect-based ones. We also show that task-specific pre-training is crucial for few-shot query-based summarization. It allows us to learn a model for this more complex task even from a handful of annotated samples.

All in all, we demonstrate significant improvements over FEWSUM in terms of ROUGE scores, input faithfulness, and human preference. Furthermore, our query-based approach yields more coherent summaries with fewer redundancies.



**Figure 6.1:** Illustration of the proposed approach. In Stage 1, all parameters of a large language model are pre-trained on generic texts (we use BART). In Stage 2, we pre-train adapters (5% of the full model’s parameters) on customer reviews using held-out reviews as summaries. In Stage 3, we fine-tune the adapters on a handful of reviews-summary pairs.

## 6.1 Introduction

The lack of sufficiently large annotated datasets led to a variety of unsupervised abstractive models (e.g., COPYCAT in Chapter 4, MEANSUM (Chu & Liu, 2019); DENOISESUM (Amplayo & Lapata, 2020)) that are trained on large collections of unannotated customer reviews. However, as the models are never exposed to actual summaries, they cannot capture their expected characteristics. This results in generated summaries mimicking the informal style of customer reviews and containing hallucinations and unimportant details.

These limitations were addressed in Chapter 5 by the few-shot method – FEWSUM – learning from a handful of human-written summaries. However, while the model is more robust to overfitting, such features require manual, domain-specific engineering and can be sub-optimal for capturing correspondences between texts on the semantic level. In this chapter, we propose a simpler approach – ADASUM – which is based on adapters (Bapna, Arivazhagan, & Firat, 2019; Houlsby et al., 2019). As we explain next, the adapters are pre-trained on customer reviews in a task-specific manner and subsequently fine-tuned on gold summaries.

We utilize a pre-trained model with powerful language understanding and generation abilities in combination with a parameter-efficient fine-tuning method – adapters. As was shown in recent studies, this method is also robust to overfitting in low-resource settings (He et al., 2021). In this way, a large pre-trained model, BART (Lewis, Liu, et al., 2020), in our case, remains frozen, and only small modules (0.6%-5% of the model parameters) are optimized. This effectively retains acquired knowledge in the pre-trained language model (PLM) without specialized training objectives as in RECADAM (S. Chen et al., 2020). However, available annotated data is not sufficient for learning in-domain specifics and results in summaries with subtle semantic mistakes. As explained next, we reduce these semantic mistakes by pre-training adapters on customer reviews.

Language models are pre-trained with generic objectives (e.g., single-document denoising) and rarely on in-domain data, such as customer reviews. Consequently, this makes them less attuned to in-domain specifics as these are hard to learn from a handful of summaries. This often results in subtle semantic mistakes. For instance, in Table 6.1, PASS (Oved & Levy, 2021) incorrectly concludes that *thin material* implies *poor quality*. To address this issue, we learn in-domain specifics from customer reviews. Concretely, we employ a self-supervised pre-training method: For any given product without a human-written summary, we predict one of the given reviews by conditioning on  $N$  other reviews with the highest lexical overlap in the *leave-one-out* fashion. As the standard training of PLMs is storage and memory inefficient (Mahabadi et al., 2021), we pre-train adapters only; see Stage 2 in Fig. 6.1. Afterwards, we fine-tune them on a small number of annotated reviews-summary pairs ( $< 100$  pairs), see Stage 3 in Fig. 6.1. All in all, our method combines the general text generation and understanding abilities of the PLM with in-domain knowledge directly related to the end task.

Well-organized content in summaries is easier to follow and thus improves user experience. However, the lack of annotated data makes it challenging to learn a desired content structure. For example, in Table 6.1, FEWSUM’s summary does not end after a concluding phrase ‘*Other than that, it’s a great top.*’ While the state-of-the-art model (PASS) addresses this issue by ranking multiple generated candidates with a specialized coherence model, we propose a simpler solution – ADAQSUM – that capitalizes on *text planning* (Hua & Wang, 2019; Moryossef, Goldberg, & Dagan, 2019b). Specifically, we allow the model to *plan ahead* by providing an intermediate summary representation in the form of a query consisting of aspect keywords. As we show, this results in more coherent text patterns with fewer redundancies. Moreover, it can be useful for summary personalization, which better reflect user interests.

We evaluate the proposed models in terms of automatic metrics and human efforts. We find that pre-training and fine-tuning of adapters leads to more than 2.0 and 1.3 ROUGE-L points improvement over fine-tuning the entire model on Amazon and Yelp datasets, respectively. We also find that our pre-trained and fine-tuned query-based model improves ROUGE-L scores by more than 2.7 and 0.9 ROUGE-L points over PASS, on Amazon and Yelp datasets, respectively, and is more preferred by humans. We further demonstrate that the query-based model (ADAQSUM) substantially improves coherence and reduces redundancies in generated summaries.

In summary, our contributions are as follows:

- We propose a self-supervised pre-training method to learn in-domain knowledge by adapters that alleviate catastrophic forgetting;
- We propose, to the best of our knowledge, the first aspect-based abstractive opinion summarizer learned from a few annotated samples;
- We substantially increase summary coherence using the query-based approach;

FEWSUM	This tank top is well made, fits well, and is comfortable to wear. The only thing is that it runs a little small, so order a size up from what you normally wear. Other than that, it's a great top. It's well made and it looks like it will last a long time. Love it!
PASS	This is a basic tank. The photo shows it going well past the models hips. However, the material used to make it this long is thin and therefore not good quality. It is also thinner than other tanks on the market but is still comfortable to wear.
ADASUM	This is a basic tank top that fits well and is comfortable to wear. The color is great and the length is long enough to wear with leggings. The quality of the product is good.
REVIEWS	... This is a basic tank ...    ... this tank fits like a normal tank top, not any longer ... I could wear it with leggings ...    ... It is THIN and runs SMALL ... It fits tight and is NOT long like in the picture ...    The tank fit very well and was comfortable to wear. I've bought much higher quality tanks ...    ... it is listed as a 'long' tank top and the photo even shows it going well past the models hips, however I'm short and the tank top is just a normal length.    ... They were a lot thinner than I like ...    Every women should own one in every color. Just feels quality I don't know how else to explain it ...    ... They are long enough that the color peeks out from under my tops. Looks cute.

**Table 6.1:** Generated summaries for an Amazon product by baseline models (FEWSUM and PASS) and our approach (ADASUM). Colored words indicate aspect keywords that were part of the query. The special marker ‘||’ separates truncated reviews.

- We show that self-supervised pre-training significantly improves performance on the query-based task;
- We demonstrate state-of-the-art results on two primary benchmarks in automatic and human evaluation.<sup>1</sup>

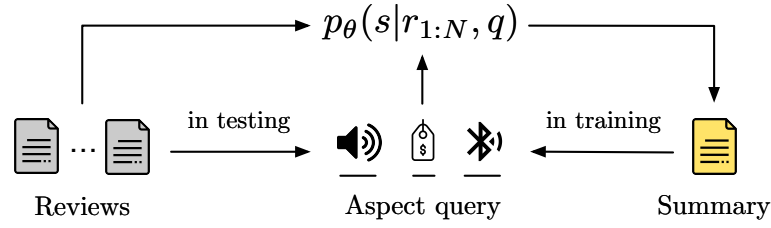
## 6.2 Approach

### 6.2.1 Opinion Summarization Tasks

In this chapter, we consider two tasks of customer review summarization. The first one is *generic summarization* (Chu & Liu, 2019), where the aim is to produce a summary that covers overall opinions in input reviews. Formally, given  $N$  input reviews  $r_{1:N}$ , the task is to predict word-by-word the summary  $s$ :

$$\mathcal{L}(s, r_{1:N}; \theta) = \sum_{t=1}^T \log p_{\theta}(s^t | s^{1:t-1}, r_{1:N}).$$

1. Our code and associated artifacts will be publicly available at <https://github.com/amazon-research/adasum>.



**Figure 6.2:** Illustration of the query-based summarizer that inputs reviews and a text query consisting of aspects, such as ‘volume,’ ‘price,’ and ‘bluetooth.’ The query is automatically created from gold summaries in training and reviews in test time.

In the second task, *query-based summarization*, we assume that the user provides a query  $q$  consisting of aspect keywords, such as ‘bluetooth,’ ‘resolution,’ and ‘battery life.’ In turn, a summarizer should generate a summary reflecting customer opinions in  $r_{1:N}$  about these aspects. Formally, given a pair of input reviews and query  $(r_{1:N}, q)$ , the task is to predict word-by-word the summary  $s$ :

$$\mathcal{L}(s, r_{1:N}, q; \theta) = \sum_{t=1}^T \log p_{\theta}(s^t | s^{1:t-1}, r_{1:N}, q).$$

Unfortunately, abstractive opinion datasets with annotated aspect queries are unavailable in the domain. To mitigate this problem, we follow Ni, Li, and McAuley (2019a) and create queries by extracting fine-grained aspect keywords from available generic summaries. Specifically, we utilize the model proposed by Y. Zhang et al. (2014) to build a fine-grained aspect lexicon from review datasets. Further, we use simple rules to determine which aspects appear in summaries; see an annotated summary in Table 6.2. At test time, we follow the intuition that a summary should reflect common opinions and create a query from  $K$  most frequent aspect keywords in input reviews. The workflow is illustrated in Fig. 6.2.

### 6.2.2 Model

Our model is based on the Transformer (Vaswani et al., 2017) encoder-decoder architecture initialized with BART (Lewis, Liu, et al., 2020). We adopt the same encoder as in (Oved & Levy, 2021; Raffel et al., 2019) where reviews are concatenated before encoding.<sup>2</sup> This allows us to capture product-level features and leverage commonalities across reviews during encoding. For query-based summarization, we concatenate a query and reviews while indicating boundaries with special markers. In this way, the encoder can contextualize aspect keywords and focus on salient review fragments reflecting these aspects.

<sup>2</sup> We experimented with the independent review encoding as in Chapter 4. However, the results were slightly worse.

---

The *cover* offers durable *protection* for the MacBook, the retractable *tilt* stands offer *protection* for the *wrists*. The *keyboard cover* can take some effort to *fit* properly, and *adjustment* to its feel may take time. However, free and fast *shipping* make up for this one potential issue.

---

**Table 6.2:** Automatically annotated Amazon summary with fine-grained aspect keywords (*underlined italic*).

### 6.2.3 Adapters

In training, a large pre-trained model remains frozen and only tiny neural networks called *adapters* (Houlsby et al., 2019) are optimized. These modules are injected into the transformer layers (both encoder and decoder).

Formally, given the input hidden vector  $h$ , the output vector  $\hat{h}$  is calculated as shown below:

$$\hat{h} = f_2(\tanh f_1(h)) + h.$$

The functions  $f_1(\cdot)$  and  $f_2(\cdot)$  are the down- and up- projection layers. At each transformer layer, two adapters are inserted right after the self-attention and the feed-forward layers, respectively. These modules consist of substantially fewer parameters than the language model, usually around 3% - 5%. Recent studies have shown that adapters are less prone to overfitting (He et al., 2021) and are more memory-efficient in training (Mahabadi et al., 2021). Finally, as the pre-trained model remains frozen, it retains all the prior knowledge for text understanding and generation. This effectively alleviates catastrophic forgetting (I. J. Goodfellow, Mirza, Xiao, Courville, & Bengio, 2013; Kemker, McClure, Abitino, Hayes, & Kanan, 2017) without modifying the training objective as in RECADAM (S. Chen et al., 2020; Yu, Liu, & Fung, 2021). We refer to our approaches as ADASUM and ADAQSUM for generic and query-based summarization, respectively.

### 6.2.4 Self-supervised Pre-training

Language models, initially pre-trained on generic text corpora, are often not accustomed to in-domain specifics. Unsurprisingly perhaps, a wide range of product-related specifics cannot be learned from a handful of annotated summaries during fine-tuning. Consequently, this can result in subtle semantic mistakes in generated summaries. We will discuss this problem and provide examples in Sec. 6.5.4. Furthermore, query-based summarization is even more challenging for learning than generic summarization. To be useful in practice, the summarizer

Split	Amazon		Yelp	
	Gold	Synthetic	Gold	Synthetic
Train	84	70,144 / 59,963	90	70,144 / 68,499
Valid	36	7,900 / 6,810	90	7,900 / 7,724
Test	60	-	120	-

**Table 6.3:** Source-target pair numbers for Amazon and Yelp, both gold and synthetic. Each pair has 8 source reviews. Generic and query-based pair statistics are separated by '/'.

should rely on the provided query after fine-tuning. However, a handful of annotated samples might be insufficient to learn this dynamic. We will analyze this problem in Sec. 6.5.1. To alleviate these two problems, we leverage unannotated customer reviews to construct synthetic datasets for pre-training.

*Synthetic In-Domain Pre-Training Dataset.* From a group of product reviews, we randomly sample one review as a *pseudo summary*  $s$  and select  $N$  reviews as input  $(r_{1:N})$ .<sup>3</sup> We select  $N$  input reviews covering the content of the summary  $s$  – that have the highest ROUGE-1 F scores. Following the naming convention we used in Chapter 4, we refer to this as *leave-one-out pre-training* (L1O). To closely resemble query-based summarization, we create aspect queries from pseudo summaries. Specifically, we leverage the aspect lexicon by matching summary keywords; in the same way as was explained in Sec. 6.2.1. In practice, we expect queries to have at least one aspect keyword. Therefore, we remove all pre-training pairs where the pseudo summary has no aspect keywords.

## 6.3 Experimental Setup

### 6.3.1 Data

To create synthetic datasets, we used customer reviews from Amazon (He & McAuley, 2016) and Yelp.<sup>4</sup> Similar to the approach in Chapter 5, we selected 4 categories: *Electronics; Clothing, Shoes and Jewelry; Home and Kitchen; Health and Personal Care*. We pre-processed the datasets by removing all reviews that are shorter than 20 words and longer than 120 words and evened the number of pairs in both datasets. Further, we used Amazon and Yelp gold summaries introduced in Chapter 5, where each product/business has 3 references and is paired with 8 reviews. Gold and synthetic dataset statistics<sup>5</sup> are presented in Table 6.3.

3. We also experimented with selecting pseudo summaries without personal pronouns – written in the formal style (Bražinskas et al., 2020a). However, we did not observe significant improvements.

4. <https://www.yelp.com/dataset>

5. For the query-based setup, we removed all instances where targets had no aspects.



### 6.3.2 Baselines

LEXRANK (Erkan & Radev, 2004) is an unsupervised extractive graph-based model that selects sentences based on graph centrality. Sentences represent nodes in a graph whose edges are weighted with tf-idf.

MEANSUM (Chu & Liu, 2019) is an unsupervised abstractive summarization model which treats a summary as a structured latent state of an auto-encoder trained to reconstruct reviews of a product.

COPYCAT is the state-of-the-art unsupervised abstractive summarizer with hierarchical continuous latent representations to model products and individual reviews, which we introduced in Chapter 4.

FEWSUM is a few-shot framework where lexical features are used to differentiate between customer reviews and summaries. In the fine-tuning phase, features leading to generation of summaries are searched. This model was presented in Chapter 5.

PASS (Oved & Levy, 2021) is based on a pre-trained T5 model (Raffel et al., 2019) that is further fine-tuned on gold summaries. At inference, the model's input is perturbed to generate multiple candidates. These candidates are further ranked by a separate model based on coherence and fluency to select the best one.

We fine-tuned the full BART model (FULL) for a fair comparison, with and without the leave-one-out pre-training. We also employed a number of simple summarization baselines. First, the CLUSTROID review was computed for each group of reviews as follows. We took each review from a group and computed ROUGE-L with respect to all other reviews. The review with the highest ROUGE score was selected as the clustroid review. Second, we sampled a RANDOM review from each group to be used as the summary. And lastly, we constructed the summary by selecting the *leading sentences* (LEAD) from each review of a group.

### 6.3.3 Experimental Details

We used a standard Transformer encoder-decoder (Vaswani et al., 2017), pre-initialized with BART large (Lewis, Liu, et al., 2020), consisting of 400M parameters. We used two adapter sizes – 0.6% and 5% of the full model's parameters. All input reviews were concatenated, following Oved and Levy (2021); Raffel et al. (2019). For parameter optimization, we used ADAM (Kingma & Ba, 2014), and summary generation was performed via the beam search of size 5 and with 3-gram blocking (Paulus et al., 2017). We used ROUGE-L as the stopping criterion on the end task, and perplexity (PPL) for pre-training. The learning rate for most experiments was set to  $5e-5$ . Aspect lexicons for query-based summarization contained 2809 and 4013 fine-grained aspects for Amazon and Yelp, respectively. In pre-training and fine-tuning, we shuffled aspects to break temporal dependencies. For fine-tuning on Yelp, we also

found it useful to exclude summary aspect keywords that do not appear in input reviews. This approximately matched the number of aspect keywords for Amazon and Yelp. At test time, we selected up to 6 and 5 most frequent aspects for Amazon and Yelp, respectively. All computations were performed on an 8-GPU p3.8-xlarge Amazon instance.

## 6.4 Results

### 6.4.1 Automatic Evaluation

	Amazon					Yelp			
	Par.↓	PPL↓	R1↑	R2↑	RL↑	PPL↓	R1↑	R2↑	RL↑
CLUSTROID	-	-	27.16	3.61	16.77	-	28.90	4.90	18.00
LEAD	-	-	27.00	4.92	14.95	-	26.20	4.57	14.32
RANDOM	-	-	25.00	3.82	15.72	-	21.48	2.59	13.87
<i>Unsupervised</i>									
LEXRANK	-	-	27.72	5.06	17.04	-	26.96	4.93	16.13
MEANSUM	25M	-	26.63	4.89	17.11	-	27.50	3.54	16.09
COPYCAT	25M	-	27.85	4.77	18.86	-	28.12	5.89	18.32
<i>Few-shot</i>									
FEWSUM	25M	-	33.56	7.16	21.49	-	37.29	9.92	22.76
PASS	440M	-	37.43	8.02	23.34	-	36.91	8.12	23.09
FULL (100%)	400M	17.87	37.22	9.17	23.51	12.87	37.40	10.27	23.76
FULL (100%) + L1O	400M	16.90	37.67	10.28	24.32	12.40	36.79	11.07	25.03
ADASUM (0.6%)	2.6M	13.45	38.49	9.84	24.37	11.94	37.55	10.11	24.08
ADASUM (0.6%) + L1O	2.6M	12.06	38.94	10.63	24.95	11.23	37.78	11.31	24.04
ADASUM (5%)	21.3M	16.30	38.15	9.18	23.17	12.50	38.12	10.89	24.11
ADASUM (5%) + L1O	21.3M	<b>12.03</b>	<b>39.78</b>	<b>10.80</b>	<b>25.55</b>	<b>11.11</b>	<b>38.82</b>	<b>11.75</b>	<b>25.14</b>

**Table 6.4:** Test set ROUGE F1 scores on gold Amazon and Yelp datasets for generic review summarization. L1O stands for leave-one-out pre-training. We also provide the total number of trainable parameters.

Table 6.4 shows results on the Amazon and Yelp test sets for generic summarization. It shows ROUGE F1 scores (Lin, 2004) as a standard measure of informativeness<sup>6</sup> and perplexity (PPL) as a measure of confusion.

First of all, the results indicate the superiority of adapters over full fine-tuning and state-of-the-art few-shot models on both datasets. As was observed in He et al. (2021), adapters are less prone to overfitting, which is especially beneficial in few-shot settings. Second, we observe a significant improvement in ROUGE scores when pre-trained models are further trained using L1O. This signifies the importance of learning in-domain specifics before fine-tuning. We also observe that adapters are more effective on the Amazon dataset, which

6. For consistency with previous works, we used the same Python package (<https://github.com/google-research/google-research/tree/master/rouge>)

		R1	R2	RL	uniq. 1-gram (%)	uniq. 2-gram (%)
Amazon	ADASUM (5%) + L1O	<b>39.78</b>	<b>10.80</b>	25.55	67.72	80.83
	ADAQSUM (5%) + L1O	38.53	10.52	<b>26.06</b>	<b>69.38</b>	<b>82.57</b>
Yelp	ADASUM (5%) + L1O	<b>38.82</b>	<b>11.75</b>	<b>25.14</b>	62.26	76.55
	ADAQSUM (5%) + L1O	36.79	10.06	23.99	<b>65.74</b>	<b>79.88</b>

**Table 6.5:** Comparison of the query-based and generic summarizers on test sets. Unique n-grams were computed in generated summaries.

is more challenging as indicated by higher perplexity (PPL).<sup>7</sup> We hypothesize that the pre-trained language model (BART) is more accustomed to restaurant- than product-related texts. Moreover, larger adapters (5%) tend to overfit on the small number of annotated instances, and L1O pre-training helps substantially, as indicated both by ROUGE scores and PPL. We provide example generated summaries in the Appendix.

### 6.4.2 Human Evaluation

*Coherence Improvement.* As was observed in (Oved & Levy, 2021), opinion summarizers sometimes generate incoherent summaries. We hypothesized that a query should allow the model to plan ahead of time and thus generate more coherent and less redundant texts. To test the hypothesis, we compared 5% adapter-based models with and without the query; both were pre-trained via L1O. We performed human evaluation in terms of *coherence* and *non-redundancy* via Best-Worst Scaling (BWS) (Louviere et al., 2015; Louviere & Woodworth, 1991). BWS has been shown to produce more reliable results than ranking scales (Kiritchenko & Mohammad, 2016b).

For each Amazon test set entry and criterion, we asked three independent workers on Amazon Mechanical Turk (AMT) to select the best and worst summary. For each criterion, a system’s score is computed as the percentage of times it was selected as best, minus the percentage of times it was selected as worst (Orme, 2009). The scores range from -100 (unanimously worst) to +100 (unanimously best). For more details, please refer to Appendix C.1.

First, the results indicate that the summaries generated by ADAQSUM are substantially more preferred to ADASUM in terms of coherence and non-redundancy. Namely, +13.73 vs -30.91 and -1.96 vs -25.93 for coherence and non-redundancy, respectively. We also computed the percentage of unique n-grams in each generated summary for both datasets, as shown in Table 6.5. The results support that query-based summaries are less redundant. However, similar to findings in Oved and Levy (2021), we observe that more coherent summaries tend to get lower ROUGE scores. Nevertheless, our model outperforms PASS by a margin on both datasets – by 2.72 and 0.9 ROUGE-L points on Amazon and Yelp, respectively.

7. Training sets are of similar sizes, i.e., 84 and 90 summaries on Amazon and Yelp, respectively

	Fluency	Coher.	Non-Red.
PASS	-21.74	<b>+33.33</b>	0.00
LEXRANK	-45.95	-52.38	-58.97
ADAQSUM (5%) + L1O	<b>+26.67</b>	+25.00	<b>+26.67</b>
GOLD	+46.67	+27.78	+55.56

**Table 6.6:** Human evaluation results in terms of the Best-Worst scaling on the Amazon test set.

	Full↑	Partial↑	No↓
FEWSUM	47.56	24.39	28.05
PASS	60.70	31.84	7.46
ADASUM (5%) + L1O	<b>78.97</b>	<b>15.48</b>	<b>5.56</b>
ADAQSUM (5%) + L1O	72.69	20.37	6.94

**Table 6.7:** Input fidelity on the Amazon test set, normalized by sentences.

*Comparison to Baselines.* To understand better how our query-based model compares to other models, we performed an additional human evaluation experiment. We used the following criteria: *coherence*, *non-redundancy*, and *fluency*. As previously, we used the Best-Worst scaling on the Amazon test set. We assigned three AMT workers to each tuple containing summaries from PASS, ADAQSUM (5%) + L1O, LEXRANK, and human annotators (GOLD).

The results in Table 6.6 suggest that summaries produced by our model are more fluent and non-redundant than the ones produced by PASS. In general, PASS produces more diverse and detailed summaries yet with more semantic mistakes that make them harder to understand (hence lower fluency scores). However, summaries by both systems are similarly preferred in terms of coherence. Also, we note that PASS utilizes a separately trained classifier on human-annotated summaries (Fabbri et al., 2021) to rank candidate summaries, while our approach does not.

*Input Content Fidelity.* As was shown in Falke et al. (2019); Tay et al. (2019), the ROUGE metric can be insensitive to hallucinations (Maynez et al., 2020). However, hallucinations can lead to user aversion, and their reduction remains an open problem in summarization. To assess the input fidelity of generated summaries, we performed a human evaluation. Specifically, we used summaries produced by the adapter models (ADASUM (5%) + L1O and ADAQSUM (5%) + L1O), FEWSUM, PASS, and human-written (GOLD). In each task (HIT), we presented both reviews and all summary sentences. We asked three workers to assess how well the content in summary sentences is supported by the reviews. The three following options were available. *Full support*: all the content is reflected in the reviews; *Partial support*: only some content is reflected in the reviews; *No support*: content is not reflected in the reviews. The results, normalized by sentences, are shown in Table 6.7.

	R1	R2	RL	AR
FULL (100%) + Q*	40.52	10.96	25.06	59.84
FULL (100%) + L1O + Q*	42.65	11.53	26.82	96.39
ADAQSUM (5%)*	41.04	11.08	25.46	60.64
ADAQSUM (5%) + L1O*	43.84	13.41	27.31	97.19
ADAQSUM (5%)	38.58	10.10	24.19	69.14
ADAQSUM (5%) + L1O	38.53	10.52	26.06	98.78

**Table 6.8:** Amazon test set ROUGE F1 for query-based summarization. Here, \* indicates that queries were created from gold summaries; AR stands for aspect recall.

First, we observe that FEWSUM hallucinates the most, potentially because it was not initialized with a pre-trained language model. Second, PASS improves input fidelity over FEWSUM yet substantially underperforms our adapter-based models. We also notice a slight decrease in input fidelity when the query is used. This is likely caused by more abstractive summaries generated by ADAQSUM, we discuss it in Sec. 6.5.3.

## 6.5 Analysis

### 6.5.1 Query-based Pre-training

Query-based summarizers should generate summaries reflecting all aspects in user queries to be useful in practice. We investigated how summarizers learn this task in the few-shot regime with and without pre-training. We created test-time queries from gold summaries (indicated by \*) and input reviews. Further, we calculated the aspect recall (AR) score by counting aspect keywords in queries present in generated summaries. The results are shown in Table 6.8.

As indicated by low AR scores, without pre-training, the models miss many aspects in queries. The increase to nearly 100% in AR suggests that pre-training is crucial for the task. The same trend remains when aspect keywords from reviews are used in queries.

### 6.5.2 Catastrophic Forgetting

ROUGE scores in Table 6.4 suggest that L1O pre-training is beneficial for the end task. However, fine-tuning on summaries can lead to the catastrophic forgetting of the acquired in-domain specifics from reviews. Because adapters have fewer parameters to optimize, we hypothesized that they might be more robust to this phenomenon.

To test the hypothesis, we evaluated two models on the pre-training L1O pairs where a review is used as a summary, before and after fine-tuning on human-written summaries. For the first model, we optimized only adapters (5%), both in pre-training and fine-tuning. And in the second case, we optimized the entire model. We used PPL to measure the model’s confusion about the pre-training pseudo summaries, as shown in Table 6.9a.

The results demonstrate that the adapter-based model better preserves information about reviews after they are fine-tuned on summaries, as indicated by lower PPL scores. Our findings are also supported by Yu et al. (2021).

	PPL↓		2-gram	3-gram	4-gram
FULL (100%) + L1O	21.51	FEWSUM	<b>78.63</b>	<b>95.59</b>	<b>98.74</b>
FULL (100%) + L1O + FT	34.87 (+13.36)	PASS	70.72	86.32	93.24
ADASUM (5%) + L1O	19.69	ADASUM (5%) + L1O	55.47	78.24	86.78
ADASUM (5%) + L1O + FT	<b>28.45 (+8.76)</b>	ADAQSUM (5%) + L1O	56.27	79.18	88.48

(a) Catastrophic forgetting evaluation on the Amazon pre-training task’s validation set.

(b) The abtractiveness of generated summaries in terms of novel n-grams on the Amazon test set.

### 6.5.3 Abtractiveness

Abstracting information in reviews is important for practical applications (Carenini & Cheung, 2008b). To investigate how well the models abstract, we computed the number of novel n-grams in generated summaries with respect to input reviews on the Amazon test set. The results in percentages are shown in Table 6.9b.

First, we observe that FEWSUM tends to produce the most abtractive summaries, followed by PASS. Second, ADAQSUM has higher abtractivness than ADASUM. We also observe that abtractiveness is inversely proportional to input faithfulness in Table 6.7, in line with previous studies Dreyer, Liu, Nan, Atluri, and Ravi (2021); Durmus, He, and Diab (2020).

### 6.5.4 Semantic Mistakes

When a pre-trained model (with and without adapters) is fine-tuned on a handful of annotated samples, it often results in summaries with subtle semantic mistakes; see examples in Table 6.10. For instance, a 5% adapter model generates a semantically contradicting fragment: *‘This **dead on arrival battery** is of good quality and holds a charge well.’*

We hypothesize that it is caused by the lack of in-domain knowledge, which we propose to learn via L1O (see Sec. 6.2.4). During a manual investigation, we observed that L1O pre-training substantially reduces semantic mistakes. This is also reflected in higher ROUGE scores in Table 6.4.

ADASUM (5%)	This Thomas the Train costume is cute and functional. The size is perfect for a toddler or 2 year old and the candy pouch is large enough to carry a lot of candy. The costume can be customized with googly eyes, pumpkin patch, spiders, bats, or train tracks to make it more suitable for a child of any age. The fit is comfortable and the <b>fit quality is great</b> . The only complaint I have is <b>the quality of the felt fabric</b> .
ADASUM (5%) + L1O	This Thomas the Train costume is very cute and the size is perfect for a 2-year-old. The hat is thin and flimsy and the face is not 3D sculpted. The candy pouch is a nice feature and it looks like it will grow with the child. Overall, it's a cute costume and will be used for Halloween next year.
ADASUM (5%)	<b>This dead on arrival battery is of good quality and holds a charge well.</b> It is easy to install and is a great value for the money. <b>However, it may not hold a charge as advertised due to the plastic case bulging.</b> Overall, this product is highly recommended.
ADASUM (5%) + L1O	This battery is a great value for the price and works great. It is a good quality battery that can be used to replace a dead battery in an alarm system. The price is great and the quality of the product is good. The shipping was fast and the customer service was excellent.

**Table 6.10:** Adapter-based models (5%) and their generated outputs with and without L1O pre-training. Semantic mistakes and disfluencies are highlighted in **bold**.

## 6.6 Related Work

In terms of in-domain pre-training, we discussed the leave-one-out objective in Chapter 5, which was used to train COPYCAT. Other works in unsupervised opinion summarization (Amplayo & Lapata, 2020; Isonuma, Mori, Bollegala, & Sakata, 2021b) also leverage customer reviews for training. In Chapter 5, we discussed FEWSUM that was pre-trained before fine-tuning. Unlike in our previous work, here we pre-train the adapters instead of the entry model to avoid catastrophic forgetting. Furthermore, FEWSUM is trained from scratch while we rely on a large pre-trained language model. Finally, we avoid hand-crafted properties completely.

In a related work OPINIONDIGEST (Suhara, Wang, Angelidis, & Tan, 2020b), the authors propose to aggregate opinions in a pipeline framework. We approach the problem end-to-end and rely on aspect keywords (e.g., price) instead of opinion phrases (e.g., good location). Controllability using input fragments (e.g., entities) and meta information (e.g., coarse-grained aspects) has received recent attention in various NLP domains (Elsahar et al., 2021; Frermann & Klementiev, 2019; Z. Liu & Chen, 2021; Narayan, Zhao, Maynez, Simoes, & McDonald, 2021). In contrast to SELFSUM (Elsahar et al., 2021), we use aspect keywords instead of generic tokens. Additionally, our setup is few-shot instead of unsupervised and we use a different model architecture.

Planning was tackled in opinion summarization in Amplayo and Lapata (2020). However, their approach is substantially less flexible, as the summary plan consists of an aspect and sentiment classes only. Query-based settings have received recent attention in the news domain (Xu & Lapata, 2020, 2021). Compared to a concurrent work on opinion summarization ACESUM (Amplayo, Angelidis, & Lapata, 2021), our approach does not require a trained aspect induction model, is few-shot instead of self-supervised, and benefits from a large collection of automatically created fine-grained aspects (a couple of thousands) instead of human annotated coarse-grained aspects (up to 18). Concurrently with our work, Poth, Pfeiffer, Rücklé, and Gurevych (2021) support our findings on the benefits of pre-training adapters for other tasks.

## 6.7 Conclusions

In this chapter, we focus on large pre-trained language models and their efficient few-shot adaptation to opinion summarization. We explore adapters – small neural networks inserted in Transformer layers – and how to store in-domain knowledge in them. Before fine-tuning, we pre-train adapters on customer reviews with the leave-one-out objective. In this way, the model learns in-domain specifics, which reduces semantic mistakes in generated summaries. We show that our approach leads to more than 2.0 and 1.3 ROUGE-L points improvement over the entire model’s fine-tuning on the Amazon and Yelp datasets, respectively.

Further, we propose a simple method for few-shot query-based summarization. The queries consist of aspect keywords reflecting potential user interests. We create these queries automatically and show that pre-training is crucial for the task, significantly improving performance. Finally, we demonstrate that the query-based model generates more coherent and less redundant summaries in human evaluation.

## 6.8 Future Work

### 6.8.1 Cross-domain

Opinion summarization is diverse in terms of review domains – restaurants (Yelp), products (Amazon), and films (Rotten Tomatoes) – to name a few. This, in turn, requires summary annotation efforts for each domain. While these domains are different in terms of review content, summaries in different domains share many similarities. These similarities can be beneficial for cross-domain and cross-category generalization. In turn, such generalization could significantly reduce the costs of deploying opinion summarizers for new domains and categories.



As we discussed in Chapter 5, summaries are expected to be written in a formal style, reflect common opinions and sentiment. Notice that these characteristics are not tied to a particular domain. We hypothesize that a competitive opinion summarizer for a target domain can be learned from other domains. Specifically, we believe that such a summarizer can achieve results, in automatic and human evaluation, close to the one learned on the target domain exclusively. This hypothesis is supported by our positive cross-category generalization results for FEWSUM on the Amazon dataset, see Sec. 5.8.3.

One potential direction would be to separately learn two skills – summary content generation and the process of summarization – in the vein of *skill modularization* (Ponti, Sordoni, & Reddy, 2022). As we discussed in Chapter 5, some reviews share many similarities with expected summaries. Furthermore, customer reviews are available in large quantities in most domains. Therefore, we could learn summary content generation for a target domain from reviews exclusively. However, the process of summarization might be hard to learn from reviews. The primary reason is that reviews rarely summarize a particular subset of other reviews. This, in turn, makes it challenging to construct high-quality synthetic datasets. To alleviate this, we could use annotated datasets in other domains, even in other branches, such as news summarization.

### 6.8.2 Query-based Summarization

In this chapter, we presented a query-based model – ADAQSUM – that is fine-tuned on a handful of summaries. While this model was not the center of the work, we believe that this direction has an immense potential in practical settings. Therefore, we would like to reflect on its two core components – the aspect extractor and aspect-based summarizer. The former is responsible for the aspect queries created from summaries and reviews in training and test time, respectively. The latter inputs reviews and a query and produces a summary. We start from the extractor and how it can be improved. At the moment of writing and to the best of our knowledge, there are no fine-grained aspect extractors not requiring supervised data for training. This lead to using an unsupervised approach yielding a fine-grained aspect lexicon. We used the lexicon to create queries, as we explained in Sec. 6.2.1. However, a better aspect extractor could significantly increase the quality of summaries. Such a model could leverage contexts to reduce false positives. Relatively often, users compare reviewed products to other products on the market or the ones they currently own. This can result in extracted aspects irrelevant to the reviewed product. For instance, ‘*battery charge indicator*’ should not be extracted from the sentence ‘*My current earset has a battery charge indicator*’. Furthermore, the aspect extractor could be enhanced to extract aspects implicitly mentioned in text. For instance, the phrase ‘*this restaurant could be better located*’, implicitly mentions ‘*location*’ in

the negative sentiment. Lastly, a heuristic used in test time to create aspect queries – frequent review aspects – can be replaced by a learned model pre-trained to mimic the training time extractor. We used this approach in Chapter 5 by replacing the oracle with the plugin network for test time. We will return to this principle in Chapter 7.

On the summarizer’s side, we could make the following improvements. First, we could account for mistakes in aspect queries, both in training and testing. These mistakes emerge as the extractor can incorrectly extract an aspect (false positive) or miss one (false negative). In turn, this closely resembles the real-world setting, where mistakes are made by the user, intentionally or unintentionally. We could address that by enhancing the summarizer to carefully evaluate the query and act accordingly. For instance, if an aspect is never mentioned in the input reviews, it should be ignored. Also, the summarizer should not generate a text fragments for aspects not present in the query. Second, the user might have only a vague idea about the aspects of interest. For instance, he might provide the keyword ‘*food*’ while be interested in details about various dishes offered by a restaurant, such as pasta, gelato, and pizza. However, the model in its current form will generate a coarse-grained summary fragment for that aspect, explicitly mentioning the keyword ‘*food*’. This could be addressed by allowing the user to specify the granularity of the summary.

GOLD	These transition tights are perfect for children sensitive to the tight sensation other tights have around the foot. The material is soft and durable; they stand up well to both the rough nature of children, and the washing machine. This product does tend to run slightly small, so purchasing one size up is recommended.
FEWSUM	These tights are a great value for the price. The fit is true to size and the quality of the tights is very good. They are well made and will last a long time. They do run a little on the small side, so order a size up.
PASS	These soft, breathable tights are great for transitioning from tap to ballet. They fit snugly around the body and stay in place when worn with ballet shoes. They are well made and well made, and can last longer than other tights available. The colors are beautiful and will definitely be purchasing again.
ADASUM (5%) + L1O	These tights are soft and comfortable and fit well. They are durable and will last a long time. They can be worn with sandals or flip-flops. They do run small and should be ordered one size up to avoid squishing toes. The color is beautiful and the material is soft and durable.
ADAQSUM (5%) + L1O	These tights are soft, comfortable, and durable. The <b>color</b> is beautiful and the <b>fit</b> is perfect for <b>tap</b> and ballet. They <b>fit</b> well and are durable enough to <b>wear</b> with flip-flops to class. They are recommended to order one size up if your child is chubby or slim.
REVIEW 1	These are the perfect tights for my 5-year old. The tights are very well made and have already lasted several washings (hang dry). The color is beautiful, and my daughter loves that she can wear flip-flops to class like the big girls do.
REVIEW 2	my 3 year old fit into these perfectly. I love these tights, they are great for wearing sandals to dance class and then pulling them over her toes to put ballet slippers on. They are nice and soft and the pink color is pretty. Will purchase again.
REVIEW 3	These are my daughters preferred ballet tights. They fit well and don't squish her toes as much as some others. The convertible option is nice as she can wear flip flops to the studio with her tights. I like that they appear to be fairly durable.
REVIEW 4	Bought this for my little one to use for her ballet class. She's almost 4 and this fits perfectly. Transition tights give her the ability to pull up the foot area to around the ankles so that they don't get dirty when not wearing her shoes, but fit well and stay in place when pulled over her feet and used with ballet shoes on.
REVIEW 5	Great soft fabric, runs small a though. U should consider getting one size up to avoid having your daughter, or son if he's into ballet, have little circulation. Don't get me wrong, great product and material.
REVIEW 6	I purchased these tights for my 4 year old because she has a tap class immediately followed by an acro class. Tights fit well and were easy to transition to bare feet after tap. I can't comment on how they hold up, as we only had one week of dance so far.
REVIEW 7	This is my go to tight when my daughter needs new ones. I won't buy anything else. They are super comfy and last forever! She is 7 but very slim and these will fit her for a while but the 8-12 is a better fit for longevity purposes but they only had this size left and I needed them right away.
REVIEW 8	These are great tights. If your girl is chubby on the waist i think you should order a number up. they are a strong tihgts. My daugther has it for more than 4 months and hasn't broken them. She only complains some days about the waist, just when she has had a lot of milk or food recently. she's 3 years old.

**Table 6.11:** Example summaries produced by different systems for an Amazon product. We highlight in **bold** keywords that were part of the query for ADAQSUM (5%) + L1O.

GOLD	The results with this scanner are sporadic at best- there are many problems with the software drivers and the stringent system requirements that come along with it. It does not work with the most recent versions of Windows, and the company charges for tech support calls.
FEWSUM	This is a great scanner. It is easy to use and easy to set up. The software that comes with it does not work with Windows Vista. It would be nice if it was a little more user friendly, but for the price it can't be beat.
PASS	The VuPoint Digital Scanner is very user friendly and easy to use. It can scan over 1200 35 mm negatives and about a dozen slides. It takes time and money to get the scanner to fit in the holders that come with it. The company also charges \$10.00 per phone call for tech support.
ADASUM (5%) + L1O	The VuPoint Digital Scanner is easy to use and does a great job converting negatives to digital format. However, the software is not compatible with newer versions of Windows. The company ArcSoft charges \$10 per phone call for tech support. Overall, this product is not recommended.
ADAQSUM (5%) + L1O	The VuPoint Digital Scanner does a great job of converting <b>photo</b> negatives to digital format. The <b>software</b> is easy to use and easy to install. However, the <b>image</b> bleaches out with too much light. A <b>replacement</b> unit is required. Overall, this product is recommended.
REVIEW 1	I recently bought this film and slide scanner to scan my grandfather's slide collection. It bleaches out the image with too much light. I tried changing the settings to improve the image quality, but had no luck. The company ArcSoft charges \$10.00 per phone call for tech support. You are better off making the investment on a nicer quality scanner.
REVIEW 2	* * Not Recommended * * Purchased as a gift in August. Opened a week ago. Spent the last week trying to get Win Xp to recognise the Vu Point scanner. Many drivers and reloads later all I have is a little black box with a red light and a message from windows that says 'USB Device Not Recognised'.
REVIEW 3	I used the VuPoint Digital Scanner to scan over 1,200 35 mm negatives and about a dozen slides and found this gadget a most user-friendly and efficient tool. I even managed to upload a few black and white negatives from 1963. I recommend the product highly.
REVIEW 4	While the software was good for Windows XP and Vista, I now have Windows 7 and would like to have software for the newer operating system. The company prefers to sell other products rather than update their software. I can't see recommending this product in today's market.
REVIEW 5	While most equipment will work with more modern versions of Windows than were available when manufactured this is not true with this scanner. Requires Windows XP means it won't work with earlier OR LATER. Its on its way back for a refund.
REVIEW 6	I found the VuPoint scanner not acceptable and I am still waiting for a replacement. My contacts with VuPoint were helpful but the equipment still did not produce acceptable images. My contact with the seller has been sporadic, at best, and a replacement unit has not been delivered. I am NOT anxious to deal with these providers again.
REVIEW 7	Product is very easy to use. Does a great job converting my slide and photo negatives to digital format. Touch-up and enhance program gave me just what I needed to clean up and enhance some of the scans, Company was great to work with!!
REVIEW 8	Its not worth the time it takes to get the negative to fit in the holders they give you. I'd much rather buy a hp flat bed scanner that lets you see the final photo image and not just an image of the negative. It takes to much time and isn't worth the money.

**Table 6.12:** Example summaries produced by different systems for an Amazon product. We highlight in **bold** keywords that were part of the query for ADAQSUM (5%) + L1O.

## PART II

# High-Resource Opinion Summarization

# Learning Opinion Summarizers by Selecting Informative Reviews

---

In Chapters 4, 5, and 6, we focused on the summarization of 8 reviews. Also, annotated datasets were scarce – we had less than 100 summaries for fine-tuning. In this chapter, we will discuss a more realistic setting, where a product can have more than 2,000 reviews. Furthermore, in our disposition, we will have more than 33,000 summaries. These summaries were written by the professional reviewers of consumer products and comprise the largest dataset in opinion summarization – AMASUM. We will explore how to efficiently deal with such a large input. Specifically, we will discuss a review selector that is memory and computationally efficient. In training, the selector assesses a large collection of reviews using ‘cheap’ lexical features and selects only a small subset of summary-relevant ones. Only these reviews are encoded by an ‘expensive’ deep encoder and subsequently summarized. We train the model end-to-end using amortized inference and policy gradient methods. Our experiments demonstrate the improved quality of summaries and reduced hallucinations over alternative review selection methods and baselines.

## 7.1 Introduction

As we already discussed in previous chapters, data scarcity is one of the central challenges in opinion summarization. Besides the absence of large annotated resources, most datasets contain summaries for only a handful of reviews (8-10). While this simplified setting allows researchers develop new opinion summarizers, it does not reflect the real-world scenarios. Specifically, in the real-world scenarios, a product can have hundreds or even thousands of reviews.

In this chapter, we make an important step forward by introducing the largest multi-document opinion summarization dataset – AMASUM. It consists of verdicts, pros and cons for more than 31,000 summarized Amazon products, as shown in Table 7.1. The summaries were written by professional product reviewers guiding online audience to make better purchasing decisions. In turn, each product is linked up to two thousands of reviews. This, however, makes

<b>Verdict</b>	If you like the idea of a <b>glass feeder</b> , this is the one to get. It has <b>a lot to offer for the price</b> .
<b>Pros</b>	<ul style="list-style-type: none"> <li>Has a <b>large opening</b> that makes it <b>easy to get in and out</b> of the feeder</li> <li>Has a <b>nice design</b> that's <b>easy to clean</b></li> </ul>
<b>Cons</b>	<ul style="list-style-type: none"> <li>The <b>lid is a little flimsy</b>, and it's <b>not as durable as some of the other models</b></li> </ul>
<b>Reviews</b>	<p>... looks just as nice as the <b>glass feeders</b>    ... Very happy with the <b>value, quality and function</b> ...    ... <b>the cheapest most flexible "jar"</b> I've ever seen ...    ... <b>Nice large opening</b> so it's easy to pour the sugar water    ... This feeder has a nice <b>large opening</b> ...    ... this is the <b>perfect design</b> and size ...    <b>The hummingbirds liked it and had no trouble feeding or perching...</b>    ... The main compartment is <b>easy to clean</b>...    ... <b>The top is a little flimsy</b> ...    ... <b>it fell out of the hanger it broke for good</b> ... <b>there are so many other nice ones out there that have glass "jar's" or at least sturdier plastic</b> ...    ... <b>The tray is easy to clean</b> ...</p>

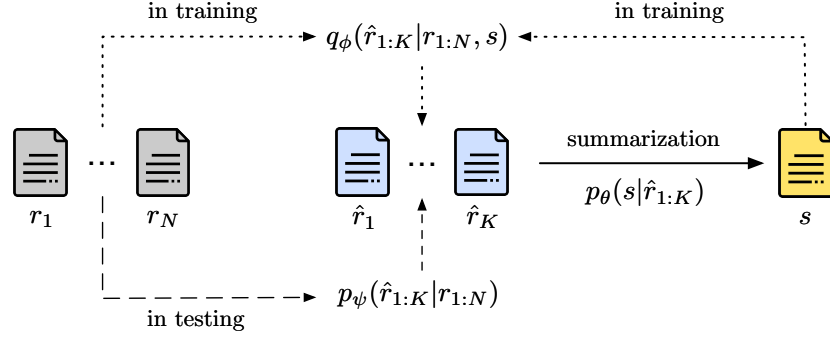
**Table 7.1:** Example summary generated by SELSUM with colored alignment to the input reviews. The reviews are truncated and delimited with ‘||’.

it virtually impossible to train a conventional encoder-decoder model using standard hardware. Moreover, not all reviews cover the summary content. Thus, training to predict summaries based on random review subsets results in hallucinations, as we will empirically demonstrate in Sec. 7.5.2. This calls for specialized methods selecting smaller subsets of relevant reviews that are fed to a summarizer. We explore this direction by introducing SELSUM that jointly learns to **select** and **summarize** review subsets using *amortized variational inference* and *policy gradient optimization* (Deng, Kim, Chiu, Guo, & Rush, 2018; Kingma & Welling, 2013; Mnih & Gregor, 2014), as depicted in Fig. 7.1.

To select relevant review subsets in training, we utilize the summary to pre-compute lexical features. Then we score review relevance with a tiny neural selector that has only 0.1% of the deep encoder’s parameters. These simple features, as opposed to deep encoder representations, allow us to select reviews from large collections without a significant computational burden. Subsequently, only selected reviews are encoded by an ‘expensive’ encoder, in order to predict the summary. To select quality review subsets in test time, when the summary is not available, we approximate the summary relevance using another neural selector. In our experiments, we show the importance of accurate review selection, affecting the summarizer in training and its output in testing. Furthermore, we show that our model outperforms alternatives in terms of ROUGE scores and content fidelity. All in all, our contributions can be summarized as follows<sup>1</sup>:

- We provide the largest dataset for multi-document opinion summarization;
- We propose an end-to-end model selecting and summarizing reviews;
- We empirically demonstrate superiority of our model to alternatives.

1. The codebase and dataset are available at <https://github.com/abrazinskas/SelSum>.



**Figure 7.1:** The SELSUM model is trained to select and summarize a subset of relevant reviews  $\hat{r}_{1:K}$  from a full set  $r_{1:N}$  using the approximate posterior  $q_\phi(\hat{r}_{1:K} | r_{1:N}, s)$ . To yield review subsets in test time, we fit and use a parametrized prior  $p_\psi(\hat{r}_{1:K} | r_{1:N})$ .

## 7.2 Dataset

The dataset (AMASUM) is based on summaries for consumer products written by professional reviewers, in English. We focused on four main professional product review platforms: `bestreviews.com` (BR); `cnet.com`; `pmag.co.uk` (PM); `runrepeat.com` (RR). The former three mostly offer content for electronic consumer products, while the last one for sport shoes. These summaries provide a quick-glance overview of a product to help users make informed purchases. Unlike customer reviewers on public platforms, such as Amazon, professional reviewers concentrate on quality writing and deliberately utilize many sources of information. These sources include reading customer reviews on public platforms, making online research, asking expert users for an opinion, and testing products themselves. In general, the summaries come in two forms. The first ones are verdicts, usually a few sentences emphasizing the most important points about a product. The second ones are pros and cons, where the most important positive and negative details about a product are presented. These tend to be more detailed, and focus on fine-grained product aspects, such as Bluetooth connectivity, resolution, and CPU clock speed.

As content providers compete for online users, the summaries **are** what the user wants as opposed to what researchers **believe** the user wants. This is in contrast to crowd-sourcing where researchers bias the worker writing process with assumptions about what constitutes a good summary. The assumptions are rarely verified by a marketing research or user testing. In turn, this has lead to a large variance of summary styles and composition even in the same domain (Angelidis & Lapata, 2018; Chu & Liu, 2019).



### 7.2.1 Content Extraction

We wrote HTML scraping programs for each platform and extracted segments containing verdicts, and pros and cons. Further, from advertisement links we extracted Amazon standard identification numbers (ASINs) which allowed us to identify what Amazon products are reviewed and link summaries to the Amazon product catalog.

We used various paid services to obtain Amazon reviews and product metadata. We fetched verified reviews for all products, and utilized unverified ones only for unpopular products (< 50 reviews). We also utilized a publicly available Amazon review dataset (Ni, Li, & McAuley, 2019b).

### 7.2.2 Filtering

We removed all reviews that have less than 10 and more than 120 words. We also removed all unpopular products that have less than 10 reviews. Further, we removed all summaries that have less than 5 words, and all instances that have either verdict or pros or cons missing. The overall statistics comparing our final dataset to available alternatives are shown in Table 7.2. Our dataset is substantially larger than the alternatives, both in terms of number of summaries and their associated reviews.

	Ent	Rev/Ent	Summaries (R)	Type	Domain
AMASUM (This chapter)	31,483	326	33,324 (1.06)	Abs.	Products
SPACE (Angelidis et al., 2020)	50	100	1,050 (3)	Abs.	Hotels
COPYCAT	60	8	180 (3)	Abs.	Products
FEWSUM	60	8	180 (3)	Abs.	Businesses
MEANSUM (Chu & Liu, 2019)	200	8	200 (1)	Abs.	Businesses
OPOSUM (Angelidis & Lapata, 2018)	60	10	180 (3)	Ext.	Products

**Table 7.2:** Statistics comparing our dataset to alternatives; R stands for the number of references. For our dataset, we show the average number of reviews and references per entity. We count verdicts, pros and cons of a product as one summary.

### 7.2.3 Summary Statistics

	Verdict			Pros			Cons		
	Len	R1	R2	Len	R1	R2	Len	R1	R2
BR (27,329)	20.60	82.40	34.45	37.34	79.12	29.75	16.27	82.19	33.58
CNET (2,717)	29.74	81.05	34.72	32.08	77.85	30.04	25.11	75.16	25.84
PM (1,756)	30.23	76.08	28.28	20.78	65.53	16.09	14.33	62.08	13.81
RR (1,522)	77.86	60.45	13.12	120.04	59.44	13.47	43.36	63.11	16.02
All (33,324)	24.47	80.95	33.18	39.82	77.40	28.31	18.12	79.69	31.10

**Table 7.3:** Summary statistics of the dataset. The number of data points is in parentheses.

We analyzed summaries from different platforms in terms of their lengths and ROUGE recall with respect to reviews, as shown in Table 7.3. First of all, verdicts tend to be shorter than pros and cons, and concentrate on fewer aspects. They also exhibit higher word overlap to user reviews as indicated by higher ROUGE scores. We also observed that pros and cons tend to concentrate on specific product features, which can often be found in product meta information (product description, the bullet list of features). Cons tend to be shorter than pros, we believe, primarily because most summarized products are rated highly (4.32/5.0 on average).

#### 7.2.4 Aspect-based Metric

In addition to standard unweighted word-overlap metrics commonly used to analyze datasets (Fabbri et al., 2019; Grusky, Naaman, & Artzi, 2018), we also leveraged an aspect specific metric. Similar to Ni et al. (2019b), we applied a parser (Y. Zhang et al., 2014) to the training set to yield (*aspect*, *opinion*, *polarity*) tuples. From the tuples, we created a lexicon, which contains fine-grained aspect keywords, such as battery life, screen, resolution, etc. In addition, to reduce noise, we manually cleaned the lexicon from aspect unrelated keywords, resulting in 2,810 entries. Further, we used the lexicon to automatically tag aspect keywords in text. Lastly, we computed *aspect precision* (AP) and *aspect recall* (AR) scores by comparing two sequences. These scores were used as features in SELSUM, we will discuss it in Sec. 7.3.2.

### 7.3 Approach

As summaries are written mostly for popular products, with more than 320 reviews on average, it is computationally challenging to encode and attend all the available ones to decode the summary. To alleviate this problem, we propose to condition the decoder on a smaller subset of reviews. However, as not all reviews provide a good content coverage of the summary. Thus, training on random subsets leads to hallucinations, as we will show in Sec. 7.5.2. Instead, we propose to learn a review selector, which chooses reviews guided by the summary. We frame this as a latent variable modeling problem (the selection is latent) and rely on the variational inference framework to train the selector, see Sec. 7.3.2. The selector (the approximate posterior) is a neural module assessing the review-summary relevance using pre-computed lexical features, thus, efficiently selecting from large review collections. Further, the selected reviews are decoded to the summary, as illustrated in Fig. 7.1. To select reviews in test time, we train a review selector that does not rely on the summary, as presented in Sec. 7.3.3.

### 7.3.1 Probabilistic Framing

Let  $\{r_{1:N}^i, s^i\}_{i=1}^M$  be reviews-summary pairs, and let  $\hat{r}_{1:K}$  be a reduced subset of reviews, where  $K < N$ , and each variable follows a categorical distribution. As review subsets  $\hat{r}_{1:K}$  are unknown in advance, they are latent variables in our model, and both the full set  $r_{1:N}$  and the summary  $s$  are observed variables. To maximize the log-likelihood shown in Eq. G.1, we have to marginalize over all possible review subsets.

$$\log p_\theta(s|r_{1:N}) = \log \mathbb{E}_{\hat{r}_{1:K} \sim p(\hat{r}_{1:K}|r_{1:N})} [p_\theta(s|\hat{r}_{1:K})] \quad (\text{G.1})$$

Unfortunately, the marginalization is intractable, and thus we leverage the Jensen's inequality (Boyd & Vandenberghe, 2004) to obtain the lower bound as shown in Eq. G.2, which, in turn, is approximated via Monte Carlo (MC).

$$\log \mathbb{E}_{\hat{r}_{1:K} \sim p(\hat{r}_{1:K}|r_{1:N})} [p_\theta(s|\hat{r}_{1:K})] \geq \mathbb{E}_{\hat{r}_{1:K} \sim p(\hat{r}_{1:K}|r_{1:N})} [\log p_\theta(s|\hat{r}_{1:K})] \quad (\text{G.2})$$

Here the latent subset  $\hat{r}_{1:K}$  is sampled from a prior categorical distribution agnostic of the summary. From the theoretical perspective, it can lead to a large gap between the log-likelihood and the lower bound, contributing to poor performance (Deng et al., 2018). From the practical perspective, it can result in the input reviews not covering the summary content, thus forcing the decoder in training to predict 'novel' content. Consequently, this leads to hallucinations (Maynez et al., 2020) in test time, as we empirically demonstrate in Sec. 7.5.2.

### 7.3.2 Model

To address the previously mentioned problems, we leverage *amortized inference* reducing the gap (Cremer, Li, & Duvenaud, 2018; Kingma & Welling, 2013). And re-formulate the lower bound using weighted sampling as shown in Eq. G.3. For more details on amortized inference, please refer to Sec. 2.5.

$$\begin{aligned} \log \mathbb{E}_{\hat{r}_{1:K} \sim p(\hat{r}_{1:K}|r_{1:N})} [p_\theta(s|\hat{r}_{1:K})] &\geq \\ \mathbb{E}_{\hat{r}_{1:K} \sim q_\phi(\hat{r}_{1:K}|r_{1:N}, s)} [\log p_\theta(s|\hat{r}_{1:K})] - \mathbb{D}_{\text{KL}} [q_\phi(\hat{r}_{1:K}|r_{1:N}, s) || p(\hat{r}_{1:K}|r_{1:N})] \end{aligned} \quad (\text{G.3})$$

The first term, known as *reconstruction*, quantifies the summary prediction quality with review subsets selected by the approximate posterior  $q_\phi(\hat{r}_{1:K}|r_{1:N}, s)$ . Unlike the prior, it selects reviews relevant to the summary  $s$ , thus providing a better content coverage of the summary. Hence, it reduces the amount of 'novel' content the decoder needs to predict. As we empirically demonstrate in Sec. 7.5.2, this results in summaries with substantially fewer hallucinations in test time. The second term, the Kullback-Leibler divergence (KLD), serves as a regularizer

preventing the posterior from deviating from the prior. We did not find it useful – presumably because the latent space of our model (i.e. the choice of reviews to summarize) has already very limited capacity – and do not use the KLD term in training. Instead, after training, we fit a rich prior (see Sec. 7.3.3).

### Approximate Posterior

The distribution assigns a probability to every possible subset of reviews  $\hat{r}_{1:K}$ . However, this would require us to consider  $N!/(N-K)!K!$  possible combinations to normalize the distribution (Koller & Friedman, 2009). To make it computationally feasible, we assume a local, left-to-right factorization (Larochelle & Murray, 2011), reducing the complexity to  $\mathcal{O}(KN)$ :

$$q_\phi(\hat{r}_{1:K}|r_{1:N}, s) = \prod_{k=1}^K q_\phi(\hat{r}_k|r_{1:N}, \hat{r}_{1:k-1}, s). \quad (\text{G.4})$$

Technically, each local distribution can be computed by softmax normalizing scores produced by the *inference network*  $f_\phi(\hat{r}_k, r_{1:N}, s)$ . To represent  $(\hat{r}_k, r_{1:N}, s)$  input tuples, we use pre-computed lexical features, such as ROUGE scores for  $(\hat{r}_k, s)$  and  $(\hat{r}_k, r_{1:N})$ , and aspect-coverage metrics (see Sec. 7.2.4). This, in turn, allows us to learn feature inter-dependencies and score large collections of reviews in a fast and memory efficient-manner.

To avoid duplicate reviews, we assume that  $\hat{r}_k$  can be any review in the full collection  $r_{1:N}$  except previously selected ones in the partial subset  $\hat{r}_{1:k-1}$ . To accommodate that, we ‘block’ scores for all previously selected reviews  $\hat{r}_{1:k-1}$  as  $f_\phi(\hat{r}_k, r_{1:N}, s) = -\inf \forall \hat{r}_k \in \hat{r}_{1:k-1}$ . In practice, we compute logits once for  $r_{1:N}$ , and then perform a progressive distribution re-normalization by ‘blocking’ logits for previously selected reviews.

### Reconstruction

In training, we optimize parameters only for the reconstruction term in Eq. G.3. However, this optimization is not straightforward as it requires backpropagation through categorical samples  $\hat{r}_{1:K}$  to compute a gradient estimate. Furthermore, it is not possible to apply the re-parametrization trick (Kingma & Welling, 2013) for categorical variables. On the other hand, the Gumbel-Softmax trick (Jang, Gu, & Poole, 2017), in its standard form, would require encoding and backpropagating through all possible review subsets, making it computationally infeasible. Instead, we used REINFORCE (Williams, 1992) that considers only a sampled subset for gradient estimation,<sup>2</sup> as shown in Eq. G.5. The notation is simplified to avoid clutter.

$$\nabla_\phi \mathbb{E}_{\hat{r}_{1:K} \sim q_\phi(\hat{r}_k|r_{1:N}, \hat{r}_{1:k-1}, s)} [\log p_\theta(s|\hat{r}_{1:K})] = \mathbb{E}_{\hat{r}_{1:K} \sim q_\phi} [(\log p_\theta(s|\hat{r}_{1:K}) - \beta(s)) \nabla_\phi \log q_\phi] \quad (\text{G.5})$$

2. We provide further discussion contrasting REINFORCE and the Gumbel-Softmax trick in Appendix D.1.

Here  $\beta(s)$  corresponds to a baseline reducing the gradient variance (Greensmith et al., 2004). Specifically, we used an MC estimate of Eq. G.2 by randomly sampling review subsets. Moreover, we were separately updating the posterior and summarizer, in the spirit of stochastic inference (Hoffman, Blei, Wang, & Paisley, 2013). In turn, this made it computationally possible to further reduce the variance by estimating Eq. G.5 with more samples.

### 7.3.3 Fitting a Prior

The selector used in training (i.e the approximate posterior  $q_\phi(\hat{r}_{1:K}|r_{1:N}, s)$ ) cannot be used in test time, as it has a look-ahead to the summary  $s$ . Instead we need a prior  $p_\psi(\hat{r}_{1:K}|r_{1:N})$ . Since we have not used any prior in training (i.e. ignored the KLD term, Eq. G.3), we, similarly in spirit to Razavi, Oord, and Vinyals (2019), fit a parameterized prior after training the summarizer, and then use the prior as the test-time review selector.

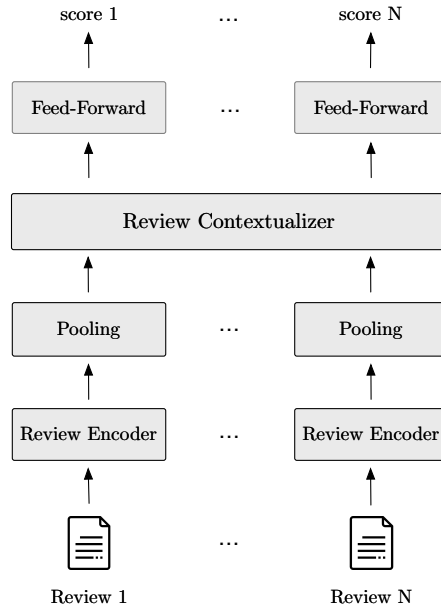
Intuitively, the fitted prior tries to mimic the predictions of the approximate posterior without having access to  $s$ . We care only about the mode of the distribution, so, to simplify the task, we select the most likely review subset from the posterior to train the test time selector and frame it as a binary prediction task. Let  $\{r_{1:N}^i, s^i\}_{i=1}^M$  be reviews-summary pairs where we utilize  $q_\phi(\hat{r}_{1:K}|r_{1:N}, s)$  to create  $\{r_{1:N}^i, d_{1:N}^i\}_{i=1}^M$  pairs. Here,  $d_j$  is a binary tag indicating whether the review  $r_j$  was selected by the posterior. This dataset is then used to train the score function  $f_\psi(r_k; r_{1:N})$ . In test time, we select  $K$  reviews with the highest scores.

On the high level, we score reviews with a binary classifier that inputs review contextualized representations as illustrated in Fig. 7.2. Below we describe details behind the prior. First, we initialized with a fine-tuned review encoder from the summarizer that was trained using a review selector (i.e. SELSUM or R1 TOP-K). The encoder produces contextualized word representations for each review independently. The word representations are obtained from the last Transformer layer. Then, we computed the weighted average of these representations to get the review representation. Further, we passed the review representations through another encoder that contextualizes them by attending representations of other reviews in the collection. Finally, we projected the outputs to scores.

## 7.4 Experimental Setup

### 7.4.1 Data Preprocessing

In our experiments, we used a preprocessed version of the dataset described in Sec. 7.2. First, we set the full review set size  $N$  to 100 maximum reviews, and the review subset size  $K$  was set to 10 entries. Further, we split the dataset to 26660, 3302, and 3362 summaries for training, validation, and testing, respectively. For our models training, verdicts, pros, and cons were joined to one sequence with a separator symbol indicating boundaries.



**Figure 7.2:** Architecture of the prior score function.

### 7.4.2 Baselines

**LEXRANK** (Erkan & Radev, 2004) is an unsupervised extractive graph-based model that selects sentences based on graph centrality. Sentences represent nodes in a graph whose edges are weighted with tf-idf.

**MEANSUM** (Chu & Liu, 2019) is an unsupervised abstractive summarization model which treats a summary as a structured latent state of an auto-encoder trained to reconstruct reviews of a product.

**COPYCAT** is an unsupervised abstractive summarizer with hierarchical continuous latent representations to model products and individual reviews. We discussed it in Chapter 4.

**RANDOM**: here we split all  $N$  reviews by sentences, and randomly selected 3, 7, 4 sentences for verdicts, pros, and cons, respectively.

**EXTSUM**: we created an extractive summarizer trained on our data. First, we used the same ROUGE greedy heuristic as in Y. Liu and Lapata (2019) to sequentially select summarizing verdict, pro, and con sentences from the full set of reviews using the actual gold summary (ORACLE). Further, we trained a model, with the same architecture as the prior in Sec. 7.3.3, to predict sentence classes. We will discuss it in detail in Sec. 7.4.4.

### 7.4.3 Alternative Review Selectors

To better understand the role of review selection, we trained the same encoder-decoder summarizer as in SELSUM but with two alternative selectors.

**RANDOM REVIEWS:** We trained and tested on random review subsets (RANDSEL). Here, review subsets were re-sampled at each training epoch.

**ROUGE-1 TOP-K:** We produced review subsets based on review-summary ROUGE-1 recall scores (R1 TOP-K) for training.<sup>3</sup> Specifically, we computed the scores for each pair, and then selected  $K$  reviews with highest scores to form the subset. To select reviews in test time, we trained a selector as in Sec. 7.3.3.

### 7.4.4 Experimental details

**SUMMARIZER:** We used the Transformer encoder-decoder architecture (Vaswani et al., 2017) initialized with base BART (Lewis, Liu, et al., 2020), 140M parameters in total. Reviews were independently encoded and concatenated states of product reviews were attended by the decoder to predict the summary as in Chapter 5. We used trainable length embeddings, and BPE (Sennrich et al., 2016) vocabulary of 51,200 subwords. Subword embeddings were shared across the encoder and decoder for regularization (Press & Wolf, 2017). For summary generation, we used beam search with the size of 5 and 3-gram blocking (Paulus et al., 2017). Parameter optimization was performed using ADAM (Kingma & Ba, 2014) with 5,000 warm-up steps. We trained SELSUM, R1 TOP-K, and RANDSEL for 8, 8, and 9 epochs, respectively. All with the learning rate of  $3e-05$ .

**POSTERIOR:** For the inference network in Sec. 7.3.2, we used a simple two-layer feed-forward, 250 hidden dimensions, with the tanh non-linearity and layer normalization before a linear transformation to scores. The model consisted of 95k parameters.

We used 23 static features by treating verdicts and pros and cons as separate summaries. For instance, ROUGE-1 and -2 scores between each review and the summary, and each review and other reviews in the full set. Similar to Ni et al. (2019b), we tagged fine-grained aspect words to compute precision and recall scores between reviews and the summary, and used them as features. We will discuss details regarding featured in Sec. 7.6.1. Lastly, we used 3 samples for the expectation estimation in Eq. G.5 and 3 samples to compute the baseline.

**PRIOR:** For the parametrized prior in Sec. 7.3.3, we used fine-tuned encoders on the end-task from both R1 TOP-K and SELSUM. For the contextualizer we used a cold-start Transformer encoder with 2 layers and 8-head attention mechanisms. For score networks, we used 2 hidden layer feed-forward networks with the ReLU non-linearities and 100 hidden dimensions. Dropouts at each layer were set to 0.10. In total, the model had 97M parameters.

---

3. We tried but were not able to obtain better results by turning the scores into a distribution and sampling from it, so we used the deterministic strategy in the main experiments.

**PROS AND CONS CLASSIFICATION:** COPYCAT and MEANSUM are not specifically designed for pros and cons generation. Therefore, we used a separately trained classifier to split each summary to pros and cons.

**EXTRACTIVE SUMMARIZER:** Our extractive summarizer had the same architecture as the prior in Sec. 7.3.3. We independently encoded sentences from reviews, contextualized them, and computed their distributions for 4 classes. In training, we considered up to 550 sentences, where only up to 16 have positive labels (4, 8, 4 for verdicts, pros, cons, respectively) marked by ORACLE. However, this resulted in label imbalance, where, in training, the model is incentivized to ignore positive labels (Li et al., 2020). However, in test time, we care about positive instances only. To counter this problem, we scaled each positive class loss by 50. In this way, the model is forced to prioritize the positive classes more. At test time, we sequentially selected top-k summarizing sentences for verdicts, pros, and cons. To make each sentence selected either for verdict, pros, or cons, we were sequentially excluding selected sentences from the pool of candidates.

**AUTOMATIC EVALUATION:** We separately evaluated verdicts, pros, and cons with the standard ROUGE package (Lin, 2004)<sup>4</sup>, and report F1 scores.

**HUMAN EVALUATION:** To assess content support, we randomly sampled 50 products, generated summaries, and hired 3 workers on Amazon Mechanical Turk (AMT) for each HIT. To ensure high quality submissions, we used qualification tasks and filters. More details can be found in Appendix D.2.

**HARDWARE:** All experiments were conducted on 4 x GeForce RTX 2080 Ti.

## 7.5 Results

### 7.5.1 Automatic Evaluation

The results in Table 7.4 suggest that the supervised models substantially outperform the unsupervised ones. Also, all supervised abstractive summarizers outperform EXTSUM, suggesting recombining information from reviews into fluent text is beneficial. Among the summarizers with the review selectors, SELSUM yields the best results on verdicts and cons. Although, we noticed that SELSUM generates shorter pros than R1 TOP-K, which may harm its scores (Fan et al., 2018)<sup>5</sup>. Further, when random reviews were used both in training and testing (RANDSEL), the results are substantially lower. On the other hand, when review subsets were produced by SELSUM and summarized by RANDSEL (marked with '\*'), we observed a substantial increase

4. We used a wrapper over the package <https://github.com/pltrdy/files2rouge>.

5. R1 TOP-K and SELSUM generate 31.95 and 27.14 words on average, respectively.



	Verdict			Pros			Cons		
	R1	R2	RL	R1	R2	RL	R1	R2	RL
ORACLE	38.14	11.76	31.50	37.22	10.53	33.50	34.09	10.75	29.66
RANDOM	13.12	0.82	10.85	14.29	1.04	13.02	9.91	0.72	8.77
LEXRANK	15.12	1.84	12.60	14.12	1.50	12.81	8.28	0.82	7.24
MEANSUM	13.78	0.93	11.70	10.44	0.63	9.55	5.95	0.45	5.29
COPYCAT	17.05	1.78	14.50	15.12	1.48	13.85	6.81	0.82	5.89
EXTSUM	18.74	3.01	15.74	19.06	2.47	17.49	11.63	1.19	10.44
RANDSEL	23.25	4.75	17.82	20.26	3.60	18.52	13.59	2.32	11.86
RANDSEL*	23.95	5.16	18.49	21.06	3.94	19.31	13.78	2.35	12.10
R1 TOP-K	23.43	4.94	18.52	<b>22.01</b>	3.94	<b>19.84</b>	14.93	2.57	12.96
SELSUM	<b>24.33</b>	<b>5.29</b>	<b>18.84</b>	21.29	<b>4.00</b>	19.39	<b>14.96</b>	<b>2.60</b>	<b>13.07</b>

**Table 7.4:** Test set ROUGE F1 scores on verdict, pros and cons. The last block shows review selection variants, where RANDSEL\* was trained on random review subsets but tested on SELSUM-selected subsets.

	Verdict			Pros			Cons		
	Full↑	Partial↓	No↓	Full↑	Partial↓	No↓	Full↑	Partial↓	No↓
RANDSEL	28.96	45.90	25.14	38.62	29.10	32.28	14.92	14.60	70.48
RANDSEL*	50.79	31.75	17.46	50.62	22.96	26.42	16.84	<b>13.75</b>	69.42
R1 TOP-K	55.21	31.77	13.02	56.07	26.61	17.31	33.33	27.78	38.89
SELSUM	<b>66.08</b>	<b>25.15</b>	<b>8.77</b>	<b>70.21</b>	<b>17.99</b>	<b>11.80</b>	<b>38.41</b>	29.21	<b>32.38</b>

**Table 7.5:** Human evaluated content support. Percentages are based on summary sentences. RANDSEL\* was trained on random review subsets but tested on SELSUM selected subsets.

in all the scores. This suggests the importance of deliberate review selection in test time. In general, all models yield higher scores on pros than cons, which is expected as most reviews are positive (on average 4.32/5) and it is harder for the model to find negative points in input reviews. We provide an example summary generated by SELSUM in Table 7.8.

### 7.5.2 Content Support

Generating input faithful summaries is crucial for practical applications, however, it remains an open problem in summarization (Fabbri et al., 2021; Maynez et al., 2020; Wang, Cho, & Lewis, 2020a). Moreover, ROUGE scores were shown not always be reliable for the content support assessment (Bražinskas et al., 2020b; Tay et al., 2019). Therefore, we evaluated generated summary sentences via AMT, as we did in Chapter 4, using the following options.

- *Full support*: all the content is reflected in the reviews;
- *Partial support*: only some content is reflected in the reviews;
- *No support*: content is not reflected in the reviews.

First, we observed that random reviews in training and testing (RANDSEL) lead to summaries with a significant amount of hallucinations. Further, when RANDSEL summarizes reviews chosen by SELSUM’s selector (‘the prior’) – indicated by ‘\*’ – the content support is still substantially lower than with SELSUM. This demonstrates that having a selection component is necessary not only at test time but also in training; without it, the model does not learn to be faithful to input reviews. Lastly, SELSUM generates substantially more input faithful summaries than R1 TOP-K.

### 7.5.3 Posterior-Selected Review Subsets

We performed extra experiments to understand why SELSUM model performs better than R1 TOP-K. Recall, their difference is only in the review selector used in training. SELSUM learns a neural model as the posterior, whereas R1 TOP-K relies on a ROUGE-1 heuristic. We hypothesize that SELSUM exploits more expressive features (beyond ROUGE-1) to select reviews that are more relevant to the summary, helping SELSUM to learn a stronger model, less prone to hallucinations.

In order to validate this, in Table 7.6 we show their results on the test set but in the training regime, i.e. with reviews selected while accessing the actual gold summary. As in training, R1 TOP-K uses the ROUGE-1 heuristic, while SELSUM relies on the learned posterior. Naturally, both methods obtain stronger scores in this artificial set-up (Table 7.6 vs. Table 7.4). What is more interesting is that SELSUM is considerably stronger than R1 TOP-K, suggesting that the SELSUM’s selection component indeed chooses more relevant reviews.

	Verdict	Pros	Cons
	RL	RL	RL
R1 TOP-K	19.38	<b>21.09</b>	13.26
SELSUM	<b>20.44</b>	20.79	<b>14.40</b>

**Table 7.6:** Test set ROUGE F1 scores when review selection is guided by the gold summary.

Lastly, to rank each feature ‘importance’, we estimated the mutual information (MI) (Kraskov, Stögbauer, & Grassberger, 2004; Ross, 2014) between the posterior input features and the binary decision to select a review, as in Sec. 7.3.3. We found that besides review-vs-summary ROUGE-1 and -2 scores, the posterior uses fine-grained aspect features, and review-vs-all-reviews ROUGE scores (quantifying the uniqueness of each review). We will provide a detailed analysis of feature importance in Sec. 7.6.1.

Feature	MI	Feature	MI
R2-R( $\hat{r}$ , $pc$ )	0.0634	R1-P( $\hat{r}$ , $v$ )	0.0208
R1-R( $\hat{r}$ , $pc$ )	0.0564	AP( $\hat{r}_k$ , $r_{-k}$ )	0.0190
R2-P( $\hat{r}$ , $pc$ )	0.0523	AR( $\hat{r}_k$ , $r_{-k}$ )	0.0173
R1-R( $\hat{r}$ , $v$ )	0.0489	LD( $\hat{r}$ , $pc$ )	0.0167
R2-R( $\hat{r}$ , $v$ )	0.0449	AP( $\hat{r}$ , $v$ )	0.0151
R2-P( $\hat{r}_k$ , $r_{-k}$ )	0.0411	LD( $\hat{r}$ , $v$ )	0.0146
R2-P( $\hat{r}$ , $v$ )	0.0405	R1-P( $\hat{r}_k$ , $r_{-k}$ )	0.0138
AR( $\hat{r}$ , $pc$ )	0.0353	AR( $\hat{r}$ , $v$ )	0.0135
R1-R( $\hat{r}_k$ , $r_{-k}$ )	0.0346	AD( $\hat{r}$ )	0.0106
AP( $\hat{r}$ , $pc$ )	0.0331	AD( $v$ )	0.0005
R2-R( $\hat{r}_k$ , $r_{-k}$ )	0.0313	AD( $pc$ )	0.0003
R1-P( $\hat{r}$ , $pc$ )	0.0266	-	-

**Table 7.7:** Full list of features sorted by their mutual information to a review being selected to the subset binary variable.

## 7.6 Analysis

### 7.6.1 Posterior Features

In total, we used 23 continuous features computed for each tuple  $(s, \hat{r}_k, \hat{r}_{1:N})$ . We can group features into three categories. The first ones were computed for a sequence standalone. The second ones were computed as  $f(\hat{r}_k, s)$  where  $\hat{r}_k$  is the current review (hypothesis) and  $s$  is the summary (reference). The last ones were computed with respect to other reviews as  $f(\hat{r}_k, r_{-k})$ , where  $r_{-k}$  (reference) are all reviews except  $\hat{r}_k$  (hypothesis). We also treated verdicts ( $v$ ) and pros and cons ( $pc$ ) as separate sequences.

Aspect precision, recall, and density are calculated by leveraging the lexicon presented in Appendix 7.2.4. Additionally, aspect density (AD) was computed as the number of unigram aspect keywords, divided by the number of unigrams in a sequence. Finally, length difference  $LD(\cdot, \cdot)$  was computed as the difference of two normalized lengths. The Normalization was performed by the maximum sequence length division.

To gain a deeper insight into the SELSUM posterior’s inner-workings, we analyzed feature importance for including a review in the subset. Same as in Sec. 7.3.3, we used the trained posterior to create a binary tagging dataset. Further, we estimated the mutual information (MI) (Kraskov et al., 2004; Ross, 2014) between the posterior input features and the binary decision variable. This allowed us to identify the dependency strength between each feature and the variable.

As features are computed separately for verdicts and pros and cons, we observed that features for pros and cons ( $pc$ ) have higher MI than for verdicts ( $v$ ), which suggests that review selection is guided by pros and cons more than by verdicts. Second, fine-grained aspect keyword based scores (AP and AR) also have high MI for  $pc$ . This is unsurprising, as pros and cons are

often more detailed, making them less predictable based on the prefix, thus the model favours reviews with matching aspect keywords. Lastly, the ROUGE scores computed against other reviews in the collection  $r_{-k}$  have high MI. This indicates reliance on global statistics computed based on the whole set of reviews.

### 7.6.2 Error Analysis

Human written pros and cons, besides summarizing customer opinion expressed in reviews, can contain details that can often be found in the product meta data. For example, users rarely mention that the same product comes in different colors. Consequently, the decoder is trained to predict such phrases based on the prefix instead of the input reviews, as shown in Example 7.9. We further observed that cons are harder, in general, to align to reviews. This is likely caused by the fact that most professionally reviewed products are often highly rated by users (4.32/5.0 on average). Thus, reviews contain fewer negative points. This is reflected in lower ROUGE scores for all systems in Sec. 7.5.1.

Human-written summaries sometimes contain customer opinion quantification expressed in phrases, such as ‘some users’ and ‘a few customers’. We observed this to be challenging for the summarizer to generate accurately as shown in Example 7.10. Especially, it applies to cons that summarize opinions of a small number of users. Logically, such reviews are hard to retrieve from a large collection in training, consequently, the model learns to rely on local statistics (the prefix). Overall, quantification of user opinions adds an additional layer of complexity for the decoder as besides generating summaries that are content supported in terms of opinions, it needs to quantify the them correctly. This is an interesting future direction for abstractive opinion summarization. Lastly, we observed that online users, in their reviews, sometimes compare the product to other products on the market. This, in turn, can confuse the model and make it generate the summary that contains fragments describing another product. Occasionally, we observed such mistakes in the output.

## 7.7 Related Work

In the context of opinion summarization, our model is related to unsupervised approach: COPYCAT (Chapter 5), MEANSUM (Chu & Liu, 2019), DENOISESUM (Amplayo & Lapata, 2020), OPINIONDIGEST (Suhara et al., 2020b), and CONDASUM (Amplayo & Lapata, 2021). Also, it is related to few-shot approaches: FEWSUM (Chapter 5), ADASUM (Chapter 6), and PASS (Oved & Levy, 2021).

Our work is also related to the extractive-abstractive summarization model (Y.-C. Chen & Bansal, 2018) that selects salient sentences from an input document using reinforcement learning. They assume one-to-one mapping between extracted and summary sentences for news. In opinion summarization, however, we often need to fuse user opinions expressed in

multiple reviews. Lastly, unlike their model, our selector and summarizer are trained jointly to predict the summary using a differentiable loss. Also, our model is related to the unsupervised paraphrasing MARGE model (Lewis, Ghazvininejad, et al., 2020), where the decoder has a modified attention mechanism accounting for the target-source document similarity. However, in their approach, the actual selection of relevant documents is performed offline via heuristics. This, in turn, makes it non-differentiable and over-reliant on the modified attention mechanism. We, however, learn the selector (posterior) jointly with summarizer, and select reviews in the online regime.

An alternative to review subsets selection are more memory and computationally efficient attention mechanisms (Beltagy et al., 2020; Pasunuru, Liu, Bansal, Ravi, & Dreyer, 2021). However, it is unclear what relationship exists between attention weights and model outputs (Jain & Wallace, 2019), thus, making it harder to offer evidence for generated summaries. In our case, the summarizer relies only on a selected subset and generates summaries faithful to its content.

In general, in news summarization, which is a more mature branch, large datasets are commonly obtained from online resources (Fabbri et al., 2019; Grusky et al., 2018; Hermann et al., 2015; Narayan et al., 2018; Sandhaus, 2008). The most relevant dataset is MULTINEWS Fabbri et al. (2019), where journalist-written summaries are linked to multiple news articles. The most similar opinion summarization dataset SPACE (Angelidis et al., 2020) contains 1050 summaries produced for 50 hotels by crowd-sourcing.

## 7.8 Conclusions

In this chapter, we introduced the largest multi-document abstractive dataset for opinion summarization – AMASUM. The dataset consists of verdicts, pros and cons, written by professional writers for more than 31,000 Amazon products. Each product is linked to more than 320 customer reviews, on average. The dataset resembles a realistic summarization scenario and comes with exciting challenges.

As product can have hundreds of reviews, standard encoding-decoding is computationally challenging. To alleviate this problem, we introduced SELSUM – a summarizer with an integrated component selecting smaller review subsets. Only selected reviews are passed to the ‘expensive’ deep encoder. The model is computationally efficient, scaling to large collections, and is trained end-to-end. We found that the ‘naive’ selection of random reviews leads to content infidelity (aka hallucinations). In contrast, SELSUM yields summaries with better ROUGE scores and that are better supported by input reviews.

## 7.9 Future Work

### 7.9.1 Meta Data

In our investigation, we observed that summary writers often utilize details in product meta data. This is particularly evident for pros and cons often covering fine-grained aspects and specific details. Considering this, a model can make more accurate predictions of summaries if meta data is provided along with reviews. In turn, this can reduce hallucinations in test time. Textual meta data, such as product descriptions and feature bullet-point lists, could be encoded with the review encoder and used as an additional input to the decoder. However, as descriptions and bullet-point features are often lengthy, this could require more efficient attention mechanisms (Beltagy et al., 2020; Zaheer et al., 2017) or more fine-grained selectors, which we describe in the next section.

Besides textual meta data, products have one or more images that reflect various aspects, such as size and color. This potentially useful information for summary predictions might not be present in neither textual meta data nor customer reviews. In order to leverage images, one might utilize a separate image encoder, similar to Im et al. (2021). Subsequently, pass the encoded images to the decoder to make more accurate summary predictions.

### 7.9.2 Content Selector

In its current form, the selector selects whole reviews which are subsequently encoded and passed on the decoder. However, reviews often contain both informative and uninformative content for summary prediction. This suggests that more a fine-grained content selection could be beneficial. Specifically, the selector could be selecting review sentences instead of whole reviews. In turn, this could significantly reduce the total input length and thus the computational and memory burden. Furthermore, sentence subsets could provide a better content coverage of summaries and thus reduce hallucinations in test time. We will return to this discussion in Sec. 8.2.1.

Summaries	
VERDICT	A comprehensive study guide for those who are new to the ASWB exam.
PROS	Offers a variety of practice questions to help you get the most out of the exam. Offers an easy-to-understand overview of the test and how it works.
CONS	The practice questions are not as detailed as the actual exam, and some questions may not be relevant to the actual questions.
Reviews	
<p>This review guide claims to reflect the 2018 blueprint of the ASWB exam. However, all ethics questions refer to a 2006 version of the NASW Code of Ethics. The code of ethics is a substantial part of the exam and many of the questions and answer explanations do not reflect what will be on the test. It is not worth the money.</p>	
<p>I passed my LMSW test on the first try with this as the only study material!!! I would definitely recommend the book for its content and practice test. I will say that the actual test is very different from this practice test in the book as the real tests involves more questions what do you do FIRST and what do you do NEXT? This guide makes it pretty easy to narrow down to two answers whereas the actual test is not that easy. ...</p>	
<p>I used this book as my primary study material for the LMSW licensing exam. While it is a thick book with a lot of information, it was helpful in preparing for the exam. There were some questions on the exam that were not in the book, but I still passed the exam with the information I studied from the book. While the exams are not all the same, I can not guarantee the same results for everyone who uses this book, but I do not have any negative reviews. ...</p>	
<p>I just graduated with my MSW in May, and studied with this book as well as the pocket prep app for about a month. This book is a great comprehensive overview of material that we have all learned, and was great for reviewing. The practice questions were also helpful in figuring out HOW the exam wants you to answer....</p>	
<p>I read through this book and utilized the practice exam at the end. While this book does go over some foundational content which is applicable to the exam, overall the content is redundant and irrelevant. The practice exam in the back of this book is extremely different than the practice exam offered through the ASWB or the actual exam itself ...</p>	
<p>Yes, I did it. I have an LCSW(not to be confused with the LCSW-clinical license as in MA the 'C' stands for certified). With this comprehensive and thorough study guide i was able to pass my exam the first time. I felt so prepared after using this. I would say, pair this study guide with an online prep app, as the exam is computer based and using a phone app you will get into the patterning needed to succeed in the exam.</p>	
<p>This study guide for the ASWB exam is great. It has a lot of review material and a practice test. There is also a code to put on a phone/tablet. I used this and passed on the first time. I also used the BSW guide for that exam and passed on the first time. A must have for anyone looking to pass the Master's Exam.</p>	
<p>This was a super purchase! It offers excellent tips and strategies to prepping for this challenging exam. It conditioned me to understand the method of the questions and not just knowledge. I just passed the first time! This was incredible because I trained in the UK and not USA. This study kit prepared me to pass what better review can one give?</p>	
<p>I passed!! This was a great study guide for me. I was intending to read the whole thing but it was a lot so I went through the table of contents and highlighted the sections I wanted to study. It also helped to read it and write it down for memorization. The practice test was hard but it really tests you on your knowledge so don't take it until you are ready. I used this and a few other practice exams.</p>	
<p>I cannot attest to results as of yet. But I can say that the book has a very organized layout and presents information about how the test is setup which provides great insight for one's approach to testing.</p>	

**Table 7.8:** Example summary generated by SELSUM with color highlighted content alignment. Some reviews were truncated to fit the page.

Summaries	
VERDICT	If you're looking for a set of glass containers that are both BPA-free and dishwasher safe, this is the one to get.
PROS	Glass containers come in a variety of sizes and colors, so you can find the right size for your needs. The lids are easy to open and close. BPA and phthalate-free.
CONS	The containers are on the smaller side, which can make them difficult to store in the microwave.
Reviews	
<p>These containers are fantastic. The lips snap on very securely but are extremely easy to remove. I do wash the lids by hand b/c they are top rack dishwasher safe only but that's not a big deal for me b/c the glass can go through the dishwasher. Plus I am saving dishwashing time anyway b/c I previously stored leftovers in Tupperware that I could not heat in the microwave so would have to transfer to a different dish before heating. Now it's a 1 stop shop. Variety of sizes are great as well.</p>	
<p>These containers are excellent!! One of the things I love about them is that I can fill them all the way to the top and it won't spill out when I put the lid on. There are a variety of sizes included and I use these every day. I only wish they were etched on the bottom with the volume that each bowl can hold. But in any case they are worth every penny. Glass containers don't discolor in the microwave and you don't have to worry about consuming plastic. ...</p>	
<p>Pro: Glass containers can go into the microwave to reheat leftovers without cooking food oils and colors into plastic. Don't use the lids in the microwave. And DO follow the instructions and wash before using, definitely! Slightly con: The rubber gasket will separate from the lid and stick to the glass container if you snap the lid on while it's slightly wet. Maybe if it's completely dry as well! ...</p>	
<p>Great quality! Nice, secure fitting lids. It's so easy to know what is being stored in the bowls. I have not used them in the freezer. I have only used them in the microwave and refrigerator. We have used every single bowl at one time or another! It's great to have different sizes to accommodate different portions. I would like to get a couple of even larger sizes if I can find them! This glass won't crack as my plastic containers did (unless I happen to drop!). ...</p>	
<p>These are the perfect size for everything, but I'm sad that the blue rubber isn't staying on the lids at all. We are avoiding washing them in dishwasher so they don't get worse. Kinda bummed. UPDATE 6/3/2018 We received the new and improved 1790 Glass Container Set &amp; the modifications made by the manufacturer were a home run! The lids fit tightly and evenly on the containers that using them is a snap ...</p>	
<p>These are such great dishes! I don't eat a lot so they are perfect for single serving cooking and storage. Love the way the lids clip in place, making a very good seal to keep the food fresh. Baking and cleaning are so easy, just the way I like things, nice and easy...</p>	
<p>Some of the flaps on the lids are a little hard to close, but I am guessing that has more to do with the fact that they are new more than anything else. Overall, this is a good, quality set at a great price...</p>	
<p>I should have read a little closer and counted the actual dishes in the photo - its a NINE piece set unless you count the lids - which you can't store food in. They might as well call it a 27 piece set because of the nine lid seals, which is realistically another 'piece'...</p>	
<p>Great glass based meal containers. The caps are plastic, and the air seal is a rubber-like material. Works as intended, however the seal part can be separated from the cap and has a tendency to adhere more onto to the glass over the cap...</p>	
<p>This product is durable and easy to clean. I love that it's BPA free and oven, microwave, freezer, and dishwasher safe. It's air tight and I haven't had anything leak even when putting liquid in. For the price you get, they're great storage containers with versatility in different temperatures...</p>	

**Table 7.9:** Example summary generated by SELSUM with color highlighted errors. As indicated in red, pros can contain details that one would expect to find in product meta data instead of customer reviews. In orange, we indicate a logical mistake. In cyan we indicate a contradiction. Some reviews were truncated to fit the page.



Summaries	
VERDICT	If you're looking for a reliable, water-resistant, and weather-resistant G-Shock watch, this is the one to get.
PROS	Solar powered. Water-resistant. Includes atomic clock and countdown timer. Includes a stopwatch and 5 alarms.
CONS	Some owners say the watch is too small for their needs. A few owners say it doesn't have the features of other models.
Reviews	
<p>It will become your everyday watch and you will enjoy it. It does everything in the description and then some. It sits nicely on your wrist without looking too big or too small, just right. It has stopwatch capabilities along with timer capabilities. Its a good digital watch and from what I've seen it can take a beating and still beep on the hour for you. Of course there's a couple accessories you can get for it too, including a screen protector and a brace to protect it. If you're looking for a good watch that you can drag through the mud and still check the time, look no further!</p>	
<p>This Casio G-Shock adjusted itself to the correct time, day of the week and date as soon as I unboxed it and light struck the solar charger. Very easy to set up, unlike some Casio products that require long sequences of button pushing. It meets my needs: updates the time and date automatically via an atomic clock signal, is solar powered, is water resistant, and displays the important information at a glance. Glad I paid a little more for this model than some of the Casios that have thick operating manuals and lots of button pushing to adjust. Well worth the money, and Amazon did a great job with prompt delivery of the correct product on top condition.</p>	
<p>OK - so it won't give you weather, or headings, or your email... but come on, it sets itself to the freakin' atomic clock EVERY SINGLE DAY. ...</p>	
<p>As a purchaser of Casio' G-Shock watches for almost four decades, I have depended on their toughness in rugged environments. This latest update has all the whistles and bells of the bigger versions, but more compact. The atomic-solar G-Shocks seem to last ten years with accurate time and no problems with the battery. ...</p>	
<p>I like the atomic clock sync and multi-time zone options. The UTC time needs to be accurate and easy to get to for celestial navigation. This watch has all that plus many other features and, of course, it's nearly impossible to break and self-charging. This is the perfect watch for someone that is outdoors or on the ocean for extended periods. ...</p>	
<p>This is one of the best value for the money G-Shocks out there. A homage to the original square G-Shock, this watch combines retro styling with modern technology like automatic time setting. Every feature works flawlessly, and it looks and feels good on the wrist. If you're strapped for cash go for the 5600 series, but if you can swing it, this is the entry level G-Shock that will leave you wanting. ...</p>	
<p>It's cheap, reliable, durable, and fashionable. It's so light it feels like you're not wearing a wrist watch. It tells the time, date, day of the week. It adjusts itself to the atomic clock nearest to you. It's solar powered. You don't have to adjust it or buy a new battery for it (for around a decade!). It has a stopwatch and 5 alarms although I don't see myself using them. ...</p>	
<p>I love these watches. It is flippin' solar powered with multi-band atomic correction! You can't beat that for maintenance-free operation. ...</p>	
<p>If your interested in a no flash, timeless, simplistic design then this watch is for you. This G-Shock is a step up from Casio classic DW-6500. It add solar charging, world time, 5 alarms, and the ability to sync with 6 different stations around the world. ...</p>	
<p>This model is considered a must have by most G-Shock aficionados. It's the classic G-Square model, but updated with solar power and atomic clock time sync features. It also has the World Time feature. ...</p>	

**Table 7.10:** Example summary generated by SELSUM with highlighted errors. In this summary, the system incorrectly generated cons with quantifiers. Some reviews were truncated to fit the page.

## Conclusions and Future Work

---

### 8.1 Conclusions

In the thesis, we discussed two settings of opinion summarization: low- and high-resource. Due to their specifics, these settings call for different learning approaches. In the low-resource setting, we utilize unsupervised learning based on unannotated customer reviews. We also leverage a handful of human-written summaries to fine-tune models with specialized techniques. In the high-resource setting, we use conventional supervised learning as substantially more annotated data is available. Another difference between the settings is the number of associated reviews per each summary. While each summary is linked to less than ten reviews in the low-resource setting, it can exceed thousands in the high-resource setting. As encoding and attending so many reviews is computationally challenging using standard deep encoder-decoder models, it calls for scalable methods. We proposed a model selecting a small subset of informative reviews that are subsequently summarized.

The presented works validate the five hypotheses stated in Sec. 1.3. Below, we conclude and link each work with a hypothesis.

*Hypothesis 1: Customer reviews without human-written summaries provide a sufficiently strong signal to train an abstractive opinion summarizer that generates fluent, coherent, and input faithful summaries.*

We started the discussion of the low-resource setting in Chapter 4 with the unsupervised model – COPYCAT. It is framed as a hierarchical latent variable model with separate latent variables representing individual reviews and a product. We train the model end-to-end on customer reviews using variational auto-encoders (Kingma & Welling, 2013) via an unsupervised objective – *leave-one-out*. Here, one review of a product is sampled as a summary and the other ones as input. In training, the model leverages commonalities between reviews, such as the brand name mentions, product features, and common opinions. We demonstrate that unannotated customer reviews provide a sufficiently strong signal to train the summarizer and subsequently generate high-quality summaries. In automatic and human evaluation, we demonstrate that our approach outperforms the state-of-the-art MEANSUM model by generating more fluent, coherent, and input faithful summaries.

*Hypothesis II: A handful of human-written summaries is sufficient to learn key summary characteristics and subsequently improve writing style and informativeness of generated summaries.*

As COPYCAT is never exposed to human-written summaries, it can, unsurprisingly perhaps, generate summaries with review-like fragments. These fragments are often uninformative and written in the informal writing style. One could leverage annotated summaries to learn these characteristics. However, the naive fine-tuning of the full model results in rapid overfitting and consequently poor summaries. In this light, in Chapter 5, we propose a few-shot summarizer – FEWSUM.

We explicitly model summary characteristics and refer to them as *properties* (Ficler & Goldberg, 2017). Properties capture differences between reviews and summaries, and we compute them automatically using heuristics. For instance, we observe that summaries inform only about what is discussed in input reviews while reviews often have novel content (e.g., unique personal experiences). We leverage heuristics, such as ROUGE scores, to compute property values and pass them as an additional input to the decoder. We pre-train the model on customer reviews and then use human-written summaries to learn their associated property values using a trainable network. The network is used in inference to yield property values for the decoder to generate summaries. First, we demonstrate that the explicit modelling of summary characteristics results in improved robustness towards overfitting in the low-resource setting. Second, in automatic and human evaluation, we show that the approach results in substantially more informative summaries written in a formal writing style than the ones generated by COPYCAT.

*Hypothesis III: In-domain and task-oriented knowledge can be efficiently stored in small neural modules and subsequently leveraged to generate summaries with accurate product specifics.*

In Chapter 6, we improve over FEWSUM by utilizing large pre-trained language models with powerful text understanding and generation abilities and an efficient fine-tuning method – *adapters* (Houlsby et al., 2019). The adapters are tiny neural networks inserted into the encoder and decoder layers, consisting of a small fraction of the language model’s parameters. However, large language models, such as BART (Lewis, Liu, et al., 2020), are pre-trained on generic texts and are rarely accustomed to product specifics. After the naive fine-tuning on a handful of annotated samples, generated summaries can contain subtle semantic mistakes. For example, ‘*This hair dryer is great for water cooling.*’ To address this problem, we pre-train the adapters on customer reviews via leave-one-out before fine-tuning them on gold samples. In both stages, only the adapters are optimized, thus leading to computational and memory savings (Mahabadi et al., 2021). Furthermore, we investigate aspect-based summarization, where a model should generate opinions only about particular aspects mentioned in the query. We conclude that a handful of gold samples is insufficient for learning this more complex task

with the naive fine-tuning method. Similarly, we pre-train the adapters on customer reviews in a query-based task-oriented manner before fine-tuning. This allows us to store task-oriented information to adapters and successfully learn the task. In automatic and human evaluation, we demonstrate that our approaches lead to substantial improvements over FEWSUM in terms of ROUGE scores and input faithfulness.

*Hypothesis IV: High-quality and large scale abstractive summarization dataset can be created from online resources.*

Human-written summary production is expensive, as it requires writers to read multiple reviews. This has led to annotated data scarcity, where most available datasets have up to 300 summaries, each based on up to 10 reviews. However, in more realistic settings, each product can have hundreds or even thousands of reviews to summarize. Therefore, in Chapter 7, we propose a large annotated dataset, called AMASUM, with more than 33,000 summaries. Each summary is associated with up to a couple of thousand reviews. These summaries were collected from professional product review websites that link their content to Amazon pages.

*Hypothesis V: Summarization can be done efficiently by learning to select the subsets of informative reviews and summarizing only the content in these reviews.*

This dataset presents a challenge as many reviews per product need to be encoded and attended. In turn, this is computationally and memory expensive using deep neural models (Beltagy et al., 2020). We address this problem by introducing SELSUM that consists of a review selector and a summarizer. In training, the selector selects a subset of summary-relevant reviews that are passed to the summarizer. The selector relies on pre-computed lexical features and only selected review subsets are encoded using an ‘expensive’ deep encoder. This approach has an insignificant computational overhead and is trained end-to-end using amortized inference and policy gradient methods. Further, we train a test time selector taking as input only reviews and use it to generate summaries in test time. We demonstrate that this approach results in high-quality summaries with more input faithful content than alternative approaches.

## 8.2 Future Work

Abstractive opinion summarization is a relatively new branch of summarization.<sup>1</sup> It presents exciting new avenues for research. We believe that the ultimate goal is to make it useful for online users and society in general. This, in turn, requires improvements over existing models. Some of the necessary improvements are related to machine learning in general, while others are to opinion summarization specifically. In this section, we iterate over open problems and future directions.

---

1. More than 80% of the papers were published within the last three years

### 8.2.1 Efficient Multi-document Modelling

In realistic settings, products and services can have thousands of reviews to summarize. In turn, this calls for efficient modelling methods of input texts. The naive concatenation of all reviews as one string and subsequent encoding-decoding via a standard Transformer is not feasible on modern hardware. The central problem resides in the computational and memory cost of the encoder's self-attention and the decoder's cross-attention that is proportional to the input's length. Broadly, there are two types of solutions for this problem: *content selection* and *efficient attention*. In Chapter 7, we explored review subset selection from large review collections, which falls under the former category. The other alternative is to use more efficient attention mechanisms. In essence, we modify attention patterns over sequences to make them sparser (Beltagy et al., 2020; Zaheer et al., 2020). However, while this reduces both computational and memory burdens, it can also reduce contextualization capabilities. More specialized methods could be explored to find the optimal balance between computational cost and contextualization.

### 8.2.2 Summary Personalization

In Chapters 4, 5, and 7, we discussed generic summarization. Models in these chapters yield summaries agnostic to individual user interests and preferences. However, a generic summary can hardly fit needs of diverse users. For example, a user might want a summary reflecting his interest in terms of product aspects, such as 'price,' 'bluetooth,' and 'connectivity.' However, the progress in this direction is hindered by the lack of annotated datasets with personalized summaries. In Chapter 6, we explored a simple method for converting generic summary datasets to query-based, and the first few-shot query-based opinion summarizer. Here query creation was based on a lexicon constructed by an aspect extractor. This direction could be further explored by improving the aspect extraction phase and finding ways to reduce the noise in extracted aspects or even consider end-to-end settings. Furthermore, one could model user profiles to produce summaries that are personalized. These profiles could include a purchasing history, topics of interest, and reviews that the user wrote.

### 8.2.3 Automatic Evaluation of Human Preference

From the global perspective, our central goal is to learn summarizers yielding summaries appealing to the user. And ideally, we would like to optimize for human preference, such as *informativness* and *coherence*. As such metrics are not readily available, we use proxy metrics, such as ROUGE. Besides not being able to capture all aspects of human judgment, ROUGE is not always well-correlated with input faithfulness as we observed and was reported in a number of studies (Fabbri et al., 2021; Maynez et al., 2020). At the moment of writing, there are no reliable ways of automatically performing input faithfulness evaluation for opinion summarization except via human evaluation. Other automatically learned metrics, such as

based on question-answering (Scialom, Lamprier, Piwowarski, & Staiano, 2019; Wang, Cho, & Lewis, 2020b) and pre-trained models BERScore (T. Zhang, Kishore, Wu, Weinberger, & Artzi, 2018) can be promising. However, they have not been successfully applied to the domain yet.

In turn, the progress in opinion summarization depends on our ability to evaluate models for human judgment. While we leverage human evaluation studies in all works, it is a sub-optimal procedure. First, it is expensive to hire workers to perform an evaluation task. Second, it is time consuming to design the actual task, build interfaces, train workers, and conduct the experiment. This significantly increases development cycles and thus slows down the progress. An alternative is to directly learn a metric of human judgment (Stiennon et al., 2020). While this would require some annotated datasets for learning, the metric could be reused in different experiments.

#### 8.2.4 Input Faithfulness

As we discussed in Sec. 1.2.3, generation of input faithful summaries is an important open problem. In Chapter 6, we showed that input faithfulness can be improved by utilizing pre-trained language models and better fine-tuning methods. Specifically, ADASUM has only 5.56% of summary sentences not supported by input reviews while FEWSUM (Chapter 5) has 28.05%. Nevertheless, input faithfulness needs to be further improved to make models applicable to industrial settings. One could consider more advanced content planning techniques (Narayan et al., 2021), such as entity chains as well as summary re-ranking methods.

#### 8.2.5 Explainable Opinion Summarization

By design, when a summary is generated, it often abstracts information present in customer reviews. For example, it is not practical to enumerate all dishes in a summary but instead abstract by an umbrella term – ‘food’. However, a user might want to ‘trace-back’ opinions that contributed to the generated content. In turn, this calls for specialized approaches that produce summaries with evidence. This direction could be further explored using *select-and-summarize* approach (Shen et al., 2019), where first we select review fragments and then summarize them.

# Unsupervised Summarization

---

## A.1 Human Evaluation Setup

To perform the human evaluation experiments described in Sections 4.7.1 and 4.7.2 we combined both tasks into single Human Intelligence Tasks (HITS). Example interface can be found in Appendix A.4. The workers needed to mark sentences as described in Section 4.7.2, and then proceed to the task in Section 4.7.1. We explicitly asked them to re-read the reviews before each task.

For worker requirements we set 98% approval rate, 1000+ HITS, Location: USA, UK, Canada, and the maximum score on a qualification test that we designed. The test was asking if the workers are native English speakers, and verifying that they correctly understand the instructions of both tasks by completing a mini version of the actual HIT.

## A.2 Full Human Evaluation Instructions

- **Fluency:** The summary sentences should be grammatically correct, easy to read and understand.
- **Coherence:** The summary should be well structured and well organized. The summary should not just be a heap of related information, but should build from sentence to sentence to a coherent body of information about a topic.
- **Non-redundancy:** There should be no unnecessary repetition in the summary. Unnecessary repetition might take the form of whole sentences that are repeated, or repeated facts, or the repeated use of a noun or noun phrase (e.g., "Bill Clinton") when a pronoun ("he") would suffice.
- **Opinion consensus:** The summary should reflect common opinions expressed in the reviews. For example, if many reviewers complain about a musty smell in the hotel's rooms, the summary should include this information.
- **Overall:** Based on your own criteria (judgment) please select the best and the worst summary of the reviews.

### A.3 Amazon Summaries Creation

First, we sampled 15 products from each of the Amazon review categories: *Electronics; Clothing, Shoes and Jewelry; Home and Kitchen; Health and Personal Care*. Then, we selected 8 reviews from each product to be summaries. We used the same requirements for workers as for human evaluation in A.1. We assigned 3 workers to each product, and instructed them to read the reviews and produce a summary text. We followed the instructions provided in (Chu & Liu, 2019), and used the following points in our instructions:

- The summary should reflect common opinions about the product expressed in the reviews. Try to preserve the common sentiment of the opinions and their details (e.g. what exactly the users like or dislike). For example, if most reviews are negative about the sound quality, then also write negatively about it. Please make the summary coherent and fluent in terms of sentence and information structure. Iterate over the written summary multiple times to improve it, and re-read the reviews whenever necessary.
- Please write your summary as if it were a review itself, e.g. 'This place is expensive' instead of 'Users thought this place was expensive'. Keep the length of the summary reasonably close to the average length of the reviews.
- Please try to write the summary using your own words instead of copying text directly from the reviews. Using the exact words from the reviews is allowed, but do not copy more than 5 consecutive words from a review .

### A.4 Human Interface Examples

We provide HTML examples of interfaces that were used for human evaluation experiments on the Amazon Mechanical Turk platform. A similar ones were used for data annotation.



**Best-Worst Summary Selection Task's Instructions** (Click to collapse)

In this task you will be presented with a number of summaries produced by different systems based on user reviews. Your task is to **select the best and worst summary** based on the **criteria** listed below.

Please **read** the reviews below and try to get an overall idea of **opinions** expressed in them.

Please read the **criteria descriptions** and **summaries** carefully, and whenever is necessary **re-read** the reviews.

This task contains validation instances (for which answers are known) that will be used for an automatic quality assessment of submissions. Therefore, please **read the reviews and summaries carefully**.

**Reviews**

**Review 1**  
\${rev\_1}

**Review 2**  
\${rev\_2}

**Review 3**  
\${rev\_3}

**Review 4**  
\${rev\_4}

**1. Informativeness**

How much **USEFUL information** about the product does the summary provide? E.g., if many reviewers complain about a size mismatch, the summary should include this information.

**Best**

☐ 1   ☐ 2   ☐ 3   ☐ 4

**Worst**

☐ 1   ☐ 2   ☐ 3   ☐ 4

**Summaries**

**Summary 1**  
\${copycat\_summ}

**Summary 2**  
\${lexrank\_summ}

**Summary 3**  
\${our\_summ}

**Summary 4**  
\${gold\_summ}

**Figure A.1:** Best-Worst Scaling interface parts. Where \$\$ indicate placeholders for actual data (reviews and summaries).

# Few-shot Learning for Opinion Summarization

---

### B.1 Best-Worst Scaling Details

We performed human evaluation based on the Amazon and Yelp test sets using the AMT platform. We assigned workers to each tuple containing summaries from COPYCAT, our model, LEXRANK, and human annotators. Due to dataset size differences, we assigned 5 and 3 workers to each tuple in the Amazon and Yelp test sets, respectively. We presented the associated reviews in a random order and asked the workers to judge summaries using the Best-Worst scaling (BWS) (Louviere et al., 2015; Louviere & Woodworth, 1991) that is known to produce more reliable results than ranking scales (Kiritchenko & Mohammad, 2016a). The judgment criteria are presented below, where *non-redundancy* and *coherence* were taken from Dang (2005). *Fluency*: the summary sentences should be grammatically correct, easy to read and understand; *Coherence*: the summary should be well structured and well organized; *Non-redundancy*: there should be no unnecessary repetition in the summary; *Informativeness*: how much useful information about the product does the summary provide?; *Sentiment*: how well the sentiment of the summary agrees with the overall sentiment of the original reviews?

### B.2 Human Evaluation Setup

To perform the human evaluation experiments described in Sec. 5.7, we hired workers with 98% approval rate, 1000+ HITS, Location: USA, UK, Canada, and the maximum score on a qualification test that we had designed. The test was asking if the workers were native English speakers, and was verifying that they correctly understood the instructions of both the best-worst scaling and content support tasks.

## B.3 Summary Annotation

For summary annotation, we reused 60 Amazon products introduced in Chapter 4 and sampled 100 businesses from Yelp. We assigned 3 Mechanical Turk workers to each product/business, and instructed them to read the reviews and produce a summary text. We used the following instructions:

- The summary should reflect user common opinions expressed in the reviews. Try to preserve the common sentiment of the opinions and their details (e.g. what exactly the users like or dislike). For example, if most reviews are negative about the sound quality, then also write negatively about it.
- Please make the summary coherent and fluent in terms of sentence and information structure. Iterate over the written summary multiple times to improve it, and re-read the reviews whenever necessary.
- The summary should not look like a review, please write formally.
- Keep the length of the summary reasonably close to the average length of the reviews.
- Please try to write the summary using your own words instead of copying text directly from the reviews. Using the exact words from the reviews is allowed but do not copy more than 5 consecutive words from a review.

# Efficient Few-shot Fine-tuning

---

### C.1 Best-Worst Scaling Details

We performed human evaluation based on the Amazon test set using the AMT platform. We assigned 3 workers to each tuple containing summaries from different systems. We showed summaries and asked to select the best and worst one based on the criterion presented below.

1. *Fluency*: the summary sentences should be grammatically correct, easy to read and understand;
2. *Coherence*: the summary should be well structured and well organized;
3. *Non-redundancy*: there should be no unnecessary repetition in the summary.

### C.2 Human Evaluation Setup

To performed the human evaluation experiments described in Sec. 6.4.2, we hired workers with 98% approval rate, 1000+ HITS, Location: USA and the maximum score on a qualification test that we had designed. The test asked if the workers were native English speakers, and verified that they correctly understood the instructions of both the best-worst scaling and content support tasks. We paid the workers an approximate amount of \$12 per hour.

# Learning Opinion Summarizers by Selecting Informative Reviews

---

## D.1 REINFORCE vs Gumbel-Softmax

In our experiments, we used REINFORCE (Williams, 1992) instead of a straight-trough Gumbel-Softmax estimator (Jang et al., 2017), which is a popular alternative. In our case, we need to sample without replacement each  $\hat{r}_k$  from the collection  $r_{1:N}$ . The Gumbel-Softmax, requires to relax the system for ‘soft’ samples used in the backward pass. In this way, the system is updated considering all possible assignments to the categorical variable, even though only one was sampled in the forward pass. For instance, one could encode all reviews  $r_{1:N}$  and weigh their word contextualized representations to obtain each  $\hat{r}_k$ . However, this is a computationally expensive and memory demanding operation. On the other hand, REINFORCE does not require this relaxation, and the encoder is exposed only to one possible assignment to each  $\hat{r}_k$ , both in the forward and backward pass.

## D.2 Human Evaluation Setup

To perform the human evaluation experiments described in Sec. 7.5.2, we hired workers with 98% approval rate, 1000+ HITS, from the USA and UK, and the maximum score on a qualification test that we had designed. We paid them 17.25 \$ per hour, on average. The task was a minimal version of the actual HIT, where we could test that workers correctly understood the instructions. Also, we asked them if they were native English speakers.

---

## Bibliography

---

- Amplayo, R. K., Angelidis, S., & Lapata, M. (2021). Aspect-controllable opinion summarization. *arXiv preprint arXiv:2109.03171*.
- Amplayo, R. K., & Lapata, M. (2020, July). Unsupervised opinion summarization with noising and denoising. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 1934–1945). Online: Association for Computational Linguistics. Retrieved from <https://www.aclweb.org/anthology/2020.acl-main.175> doi: 10.18653/v1/2020.acl-main.175
- Amplayo, R. K., & Lapata, M. (2021, April). Informative and controllable opinion summarization. In *Proceedings of the 16th conference of the european chapter of the association for computational linguistics: Main volume* (pp. 2662–2672). Online: Association for Computational Linguistics. Retrieved from <https://aclanthology.org/2021.eacl-main.229>
- Angelidis, S., Amplayo, R. K., Suhara, Y., Wang, X., & Lapata, M. (2020). Extractive opinion summarization in quantized transformer spaces. In *In transactions of the association for computational linguistics (tacl)*.
- Angelidis, S., & Lapata, M. (2018, October-November). Summarizing opinions: Aspect extraction meets sentiment prediction and they are both weakly supervised. In *Proceedings of the 2018 conference on empirical methods in natural language processing* (pp. 3675–3686). Brussels, Belgium: Association for Computational Linguistics. Retrieved from <https://www.aclweb.org/anthology/D18-1403> doi: 10.18653/v1/D18-1403
- Bahdanau, D., Cho, K., & Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *Proceedings of international conference on learning representations (iclr)*.
- Banerjee, S., & Lavie, A. (2005, June). METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization* (pp. 65–72). Ann Arbor, Michigan: Association for Computational Linguistics. Retrieved from <https://aclanthology.org/W05-0909>
- Bapna, A., Arivazhagan, N., & Firat, O. (2019). Simple, scalable adaptation for neural machine translation. *arXiv preprint arXiv:1909.08478*.

- Barzilay, R., McKeown, K. R., & Elhadad, M. (1999, June). Information fusion in the context of multi-document summarization. In *Proceedings of the 37th annual meeting of the association for computational linguistics* (pp. 550–557). College Park, Maryland, USA: Association for Computational Linguistics. Retrieved from <https://aclanthology.org/P99-1071> doi: 10.3115/1034678.1034760
- Beltagy, I., Peters, M. E., & Cohan, A. (2020). Longformer: The long-document transformer. *arXiv:2004.05150*.
- Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8), 1798–1828.
- Besag, J. (1975). Statistical analysis of non-lattice data. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 24(3), 179–195.
- Bishop, C. M. (2006). *Pattern recognition* (Vol. 128) (No. 9).
- Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan), 993–1022.
- Blitzer, J., Dredze, M., & Pereira, F. (2007). Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th annual meeting of the association of computational linguistics* (pp. 440–447).
- Bowman, S. R., Vilnis, L., Vinyals, O., Dai, A., Jozefowicz, R., & Bengio, S. (2016, August). Generating sentences from a continuous space. In *Proceedings of the 20th SIGNLL conference on computational natural language learning* (pp. 10–21). Berlin, Germany: Association for Computational Linguistics. Retrieved from <https://aclanthology.org/K16-1002> doi: 10.18653/v1/K16-1002
- Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.
- Bražinskas, A., Lapata, M., & Titov, I. (2020a, November). Few-shot learning for opinion summarization. In *Proceedings of the 2020 conference on empirical methods in natural language processing (emnlp)* (pp. 4119–4135). Online: Association for Computational Linguistics. Retrieved from <https://www.aclweb.org/anthology/2020.emnlp-main.337> doi: 10.18653/v1/2020.emnlp-main.337
- Bražinskas, A., Lapata, M., & Titov, I. (2020b, July). Unsupervised opinion summarization as copycat-review generation. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 5151–5169). Online: Association for Computational Linguistics. Retrieved from <https://www.aclweb.org/anthology/2020.acl-main.461> doi: 10.18653/v1/2020.acl-main.461

- Brazinskas, A., Lapata, M., & Titov, I. (2021). Learning opinion summarizers by selecting informative reviews. In *Emnlp*.
- Bražinskas, A., Lapata, M., & Titov, I. (2021, November). Learning opinion summarizers by selecting informative reviews. In *Proceedings of the 2021 conference on empirical methods in natural language processing* (pp. 9424–9442). Online and Punta Cana, Dominican Republic: Association for Computational Linguistics. Retrieved from <https://aclanthology.org/2021.emnlp-main.743> doi: 10.18653/v1/2021.emnlp-main.743
- Bražinskas, A., Nallapati, R., Bansal, M., & Dreyer, M. (2022). Efficient few-shot fine-tuning for opinion summarization. In *Findings of the conference on empirical methods in natural language processing (emnlp)*.
- Carenini, G., & Cheung, J. C. K. (2008a). Extractive vs. nlg-based abstractive summarization of evaluative text: The effect of corpus controversiality. In *Proceedings of the fifth international natural language generation conference* (pp. 33–41).
- Carenini, G., & Cheung, J. C. K. (2008b, June). Extractive vs. NLG-based abstractive summarization of evaluative text: The effect of corpus controversiality. In *Proceedings of the fifth international natural language generation conference* (pp. 33–41). Salt Fork, Ohio, USA: Association for Computational Linguistics. Retrieved from <https://aclanthology.org/W08-1106>
- Chen, S., Hou, Y., Cui, Y., Che, W., Liu, T., & Yu, X. (2020, November). Recall and learn: Fine-tuning deep pretrained language models with less forgetting. In *Proceedings of the 2020 conference on empirical methods in natural language processing (emnlp)* (pp. 7870–7881). Online: Association for Computational Linguistics. Retrieved from <https://aclanthology.org/2020.emnlp-main.634> doi: 10.18653/v1/2020.emnlp-main.634
- Chen, Y.-C., & Bansal, M. (2018). Fast abstractive summarization with reinforce-selected sentence rewriting. In *Proceedings of acl*.
- Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014, October). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1724–1734). Doha, Qatar: Association for Computational Linguistics. Retrieved from <https://aclanthology.org/D14-1179> doi: 10.3115/v1/D14-1179
- Chu, E., & Liu, P. (2019). Meansum: a neural model for unsupervised multi-document abstractive summarization. In *Proceedings of international conference on machine learning (icml)* (pp. 1223–1232).



- Cremer, C., Li, X., & Duvenaud, D. (2018). Inference suboptimality in variational autoencoders. In *International conference on machine learning* (pp. 1078–1086).
- Dang, H. T. (2005). Overview of duc 2005. In *Proceedings of the document understanding conference* (Vol. 2005, pp. 1–12).
- Deng, Y., Kim, Y., Chiu, J., Guo, D., & Rush, A. M. (2018). Latent alignment and variational attention. In *Proceedings of the 32nd international conference on neural information processing systems* (p. 9735–9747). Red Hook, NY, USA: Curran Associates Inc.
- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019, June). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the north American chapter of the association for computational linguistics: Human language technologies, volume 1 (long and short papers)* (pp. 4171–4186). Minneapolis, Minnesota: Association for Computational Linguistics. Retrieved from <https://aclanthology.org/N19-1423> doi: 10.18653/v1/N19-1423
- Di Fabrizio, G., Stent, A., & Gaizauskas, R. (2014). A hybrid approach to multi-document summarization of opinions in reviews. In (pp. 54–63).
- Dreyer, M., Liu, M., Nan, F., Atluri, S., & Ravi, S. (2021). Analyzing the abstractiveness-factuality tradeoff with nonlinear abstractiveness constraints. *CoRR*, *abs/2108.02859*. Retrieved from <https://arxiv.org/abs/2108.02859>
- Durmus, E., He, H., & Diab, M. (2020, July). FEQA: A question answering evaluation framework for faithfulness assessment in abstractive summarization. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 5055–5070). Online: Association for Computational Linguistics. Retrieved from <https://aclanthology.org/2020.acl-main.454> doi: 10.18653/v1/2020.acl-main.454
- Elsahar, H., Coavoux, M., Rozen, J., & Gallé, M. (2021, April). Self-supervised and controlled multi-document opinion summarization. In *Proceedings of the 16th conference of the european chapter of the association for computational linguistics: Main volume* (pp. 1646–1662). Online: Association for Computational Linguistics. Retrieved from <https://aclanthology.org/2021.eacl-main.141>
- Erkan, G., & Radev, D. R. (2004). Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, *22*, 457–479.
- Fabbri, A., Kryściński, W., McCann, B., Xiong, C., Socher, R., & Radev, D. (2021). Summeval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics*, *9*, 391–409.

- Fabbri, A., Li, I., She, T., Li, S., & Radev, D. (2019, July). Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th annual meeting of the association for computational linguistics* (pp. 1074–1084). Florence, Italy: Association for Computational Linguistics. Retrieved from <https://www.aclweb.org/anthology/P19-1102> doi: 10.18653/v1/P19-1102
- Falke, T., Ribeiro, L. F. R., Utama, P. A., Dagan, I., & Gurevych, I. (2019, July). Ranking generated summaries by correctness: An interesting but challenging application for natural language inference. In *Proceedings of the 57th annual meeting of the association for computational linguistics* (pp. 2214–2220). Florence, Italy: Association for Computational Linguistics. Retrieved from <https://www.aclweb.org/anthology/P19-1213> doi: 10.18653/v1/P19-1213
- Fan, A., Grangier, D., & Auli, M. (2018, July). Controllable abstractive summarization. In *Proceedings of the 2nd workshop on neural machine translation and generation* (pp. 45–54). Melbourne, Australia: Association for Computational Linguistics. Retrieved from <https://www.aclweb.org/anthology/W18-2706> doi: 10.18653/v1/W18-2706
- Ficler, J., & Goldberg, Y. (2017, September). Controlling linguistic style aspects in neural language generation. In *Proceedings of the workshop on stylistic variation* (pp. 94–104). Copenhagen, Denmark: Association for Computational Linguistics.
- Finn, C., Abbeel, P., & Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th international conference on machine learning-volume 70* (pp. 1126–1135).
- Frermann, L., & Klementiev, A. (2019, July). Inducing document structure for aspect-based summarization. In *Proceedings of the 57th annual meeting of the association for computational linguistics* (pp. 6263–6273). Florence, Italy: Association for Computational Linguistics. Retrieved from <https://aclanthology.org/P19-1630> doi: 10.18653/v1/P19-1630
- Fu, H., Li, C., Liu, X., Gao, J., Celikyilmaz, A., & Carin, L. (2019, June). Cyclical annealing schedule: A simple approach to mitigating KL vanishing. 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)* (pp. 240–250). Minneapolis, Minnesota: Association for Computational Linguistics. Retrieved from <https://aclanthology.org/N19-1021> doi: 10.18653/v1/N19-1021

- Ganesan, K., Zhai, C., & Han, J. (2010, August). Opinosis: A graph based approach to abstractive summarization of highly redundant opinions. In *Proceedings of the 23rd international conference on computational linguistics (coling 2010)* (pp. 340–348). Beijing, China: Coling 2010 Organizing Committee. Retrieved from <https://www.aclweb.org/anthology/C10-1039>
- Gelman, A., & Hill, J. (2006). *Data analysis using regression and multilevel/hierarchical models*. Cambridge university press.
- Gerani, S., Mehdad, Y., Carenini, G., Ng, R. T., & Nejat, B. (2014, October). Abstractive summarization of product reviews using discourse structure. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1602–1613). Doha, Qatar: Association for Computational Linguistics. Retrieved from <https://aclanthology.org/D14-1168> doi: 10.3115/v1/D14-1168
- Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics* (pp. 249–256).
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.
- Goodfellow, I. J., Mirza, M., Xiao, D., Courville, A., & Bengio, Y. (2013). An empirical investigation of catastrophic forgetting in gradient-based neural networks. In *Proceedings of international conference on learning representations (iclr)*.
- Greensmith, E., Bartlett, P. L., & Baxter, J. (2004). Variance reduction techniques for gradient estimates in reinforcement learning. *Journal of Machine Learning Research*, 5(9).
- Griffiths, T. L., & Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National academy of Sciences*, 101(suppl 1), 5228–5235.
- Grusky, M., Naaman, M., & Artzi, Y. (2018, June). Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In *Proceedings of the 2018 conference of the north American chapter of the association for computational linguistics: Human language technologies, volume 1 (long papers)* (pp. 708–719). New Orleans, Louisiana: Association for Computational Linguistics. Retrieved from <https://www.aclweb.org/anthology/N18-1065> doi: 10.18653/v1/N18-1065
- He, R., Liu, L., Ye, H., Tan, Q., Ding, B., Cheng, L., ... Si, L. (2021, August). On the effectiveness of adapter-based tuning for pretrained language model adaptation. 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)* (pp. 2208–2222). Online: Association for Computational Linguistics. Retrieved from <https://aclanthology.org/2021.acl-long.172> doi: 10.18653/v1/2021.acl-long.172

- He, R., & McAuley, J. (2016). Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web* (pp. 507–517).
- Hermann, K. M., Kocisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., & Blunsom, P. (2015). Teaching machines to read and comprehend. In *Advances in neural information processing systems*.
- Hoang, A., Bosselut, A., Celikyilmaz, A., & Choi, Y. (2019). Efficient adaptation of pretrained transformers for abstractive summarization. *arXiv preprint arXiv:1906.00138*.
- Hoffman, M. D., Blei, D. M., Wang, C., & Paisley, J. (2013). Stochastic variational inference. *Journal of Machine Learning Research*, 14(5).
- Holtzman, A., Buys, J., Forbes, M., & Choi, Y. (2019). The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.
- Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., . . . Gelly, S. (2019). Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th international conference on machine learning*.
- Hu, M., & Liu, B. (2004). Mining and summarizing customer reviews. In *Proceedings of the tenth acm sigkdd international conference on knowledge discovery and data mining* (pp. 168–177).
- Hua, X., & Wang, L. (2019, November). Sentence-level content planning and style specification for neural text generation. 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)* (pp. 591–602). Hong Kong, China: Association for Computational Linguistics. Retrieved from <https://aclanthology.org/D19-1055> doi: 10.18653/v1/D19-1055
- Huszár, F. (2015). How (not) to train your generative model: Scheduled sampling, likelihood, adversary? *arXiv preprint arXiv:1511.05101*.
- Im, J., Kim, M., Lee, H., Cho, H., & Chung, S. (2021, August). Self-supervised multimodal opinion summarization. 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)* (pp. 388–403). Online: Association for Computational Linguistics. Retrieved from <https://aclanthology.org/2021.acl-long.33> doi: 10.18653/v1/2021.acl-long.33

- Iso, H., Wang, X., Suhara, Y., Angelidis, S., & Tan, W.-C. (2021, November). Convex Aggregation for Opinion Summarization. In *Findings of the association for computational linguistics: Emnlp 2021* (pp. 3885–3903). Punta Cana, Dominican Republic: Association for Computational Linguistics. Retrieved from <https://aclanthology.org/2021.findings-emnlp.328> doi: 10.18653/v1/2021.findings-emnlp.328
- Isonuma, M., Fujino, T., Mori, J., Matsuo, Y., & Sakata, I. (2017, September). Extractive summarization using multi-task learning with document classification. In *Proceedings of the 2017 conference on empirical methods in natural language processing* (pp. 2101–2110). Copenhagen, Denmark: Association for Computational Linguistics. Retrieved from <https://www.aclweb.org/anthology/D17-1223> doi: 10.18653/v1/D17-1223
- Isonuma, M., Mori, J., Bollegala, D., & Sakata, I. (2021a). Unsupervised abstractive opinion summarization by generating sentences with tree-structured topic guidance. *Transactions of the Association for Computational Linguistics*, 9, 945–961.
- Isonuma, M., Mori, J., Bollegala, D., & Sakata, I. (2021b). Unsupervised abstractive opinion summarization by generating sentences with tree-structured topic guidance. *Transactions of the Association for Computational Linguistics*, 9, 945–961.
- Isonuma, M., Mori, J., & Sakata, I. (2019, July). Unsupervised neural single-document summarization of reviews via learning latent discourse structure and its ranking. In *Proceedings of the 57th annual meeting of the association for computational linguistics* (pp. 2142–2152). Florence, Italy: Association for Computational Linguistics. Retrieved from <https://www.aclweb.org/anthology/P19-1206> doi: 10.18653/v1/P19-1206
- Jain, S., & Wallace, B. C. (2019, June). Attention is not Explanation. 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)* (pp. 3543–3556). Minneapolis, Minnesota: Association for Computational Linguistics. Retrieved from <https://aclanthology.org/N19-1357> doi: 10.18653/v1/N19-1357
- Jang, E., Gu, S., & Poole, B. (2017). Categorical reparameterization with gumbel-softmax. In *Proceedings of international conference on learning representations (iclr)*.
- Ke, W., Gao, J., Shen, H., & Cheng, X. (2022). Consistsum: Unsupervised opinion summarization with the consistency of aspect, sentiment and semantic. In *Proceedings of the fifteenth acm international conference on web search and data mining* (p. 467–475). New York, NY, USA: Association for Computing Machinery. Retrieved from <https://doi.org/10.1145/3488560.3498463> doi: 10.1145/3488560.3498463

- Kemker, R., McClure, M., Abitino, A., Hayes, T., & Kanan, C. (2017). Measuring catastrophic forgetting in neural networks. *arXiv preprint arXiv:1708.02072*.
- Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.
- Kiritchenko, S., & Mohammad, S. M. (2016a). Capturing reliable fine-grained sentiment associations by crowdsourcing and best–worst scaling. In *Proceedings of the 2016 conference of the north american chapter of the association for computational linguistics: Human language technologies* (pp. 811–817).
- Kiritchenko, S., & Mohammad, S. M. (2016b, June). Capturing reliable fine-grained sentiment associations by crowdsourcing and best–worst scaling. In *Proceedings of the 2016 conference of the north American chapter of the association for computational linguistics: Human language technologies* (pp. 811–817). San Diego, California: Association for Computational Linguistics. Retrieved from <https://www.aclweb.org/anthology/N16-1095> doi: 10.18653/v1/N16-1095
- Kiros, R., Zhu, Y., Salakhutdinov, R. R., Zemel, R., Urtasun, R., Torralba, A., & Fidler, S. (2015). Skip-thought vectors. In *Advances in neural information processing systems* (pp. 3294–3302).
- Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., ... Herbst, E. (2007, June). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions* (pp. 177–180). Prague, Czech Republic: Association for Computational Linguistics. Retrieved from <https://www.aclweb.org/anthology/P07-2045>
- Koller, D., & Friedman, N. (2009). *Probabilistic graphical models: principles and techniques*. MIT press.
- Kraskov, A., Stögbauer, H., & Grassberger, P. (2004). Estimating mutual information. *Physical review E*, 69(6), 066138.
- Larochelle, H., & Murray, I. (2011). The neural autoregressive distribution estimator. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics* (pp. 29–37).

- Lebanoff, L., Song, K., & Liu, F. (2018, October-November). Adapting the neural encoder-decoder framework from single to multi-document summarization. In *Proceedings of the 2018 conference on empirical methods in natural language processing* (pp. 4131–4141). Brussels, Belgium: Association for Computational Linguistics. Retrieved from <https://www.aclweb.org/anthology/D18-1446> doi: 10.18653/v1/D18-1446
- Lewis, M., Ghazvininejad, M., Ghosh, G., Aghajanyan, A., Wang, S., & Zettlemoyer, L. (2020). Pre-training via paraphrasing. In *Proceedings of advances in neural information processing systems*.
- Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., ... Zettlemoyer, L. (2020, July). 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* (pp. 7871–7880). Online: Association for Computational Linguistics. Retrieved from <https://www.aclweb.org/anthology/2020.acl-main.703> doi: 10.18653/v1/2020.acl-main.703
- Li, X., Sun, X., Meng, Y., Liang, J., Wu, F., & Li, J. (2020, July). Dice loss for data-imbalanced NLP tasks. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 465–476). Online: Association for Computational Linguistics. Retrieved from <https://www.aclweb.org/anthology/2020.acl-main.45> doi: 10.18653/v1/2020.acl-main.45
- Lin, C.-Y. (2004, July). ROUGE: A package for automatic evaluation of summaries. In *Text summarization branches out* (pp. 74–81). Barcelona, Spain: Association for Computational Linguistics. Retrieved from <https://www.aclweb.org/anthology/W04-1013>
- Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1), 1–167.
- Liu, B., Hsu, W., Ma, Y., et al. (1998). Integrating classification and association rule mining. In *Kdd* (Vol. 98, pp. 80–86).
- Liu, P. J., Saleh, M., Pot, E., Goodrich, B., Sepassi, R., Kaiser, L., & Shazeer, N. (2018). Generating wikipedia by summarizing long sequences. In *Proceedings of international conference on learning representations (iclr)*.
- Liu, Y., & Lapata, M. (2019, November). Text summarization with pretrained encoders. 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)* (pp. 3730–3740). Hong Kong, China: Association for Computational Linguistics. Retrieved from <https://www.aclweb.org/anthology/D19-1387> doi: 10.18653/v1/D19-1387

- Liu, Z., & Chen, N. (2021, November). Controllable neural dialogue summarization with personal named entity planning. In *Proceedings of the 2021 conference on empirical methods in natural language processing* (pp. 92–106). Online and Punta Cana, Dominican Republic: Association for Computational Linguistics. Retrieved from <https://aclanthology.org/2021.emnlp-main.8> doi: 10.18653/v1/2021.emnlp-main.8
- Louviere, J. J., Flynn, T. N., & Marley, A. A. J. (2015). *Best-worst scaling: Theory, methods and applications*. Cambridge University Press.
- Louviere, J. J., & Woodworth, G. G. (1991). Best-worst scaling: A model for the largest difference judgments. *University of Alberta: Working Paper*.
- Lu, Y., Zhai, C., & Sundaresan, N. (2009). Rated aspect summarization of short comments. In *Proceedings of the 18th international conference on world wide web* (pp. 131–140).
- MacKay, D. J. (2003). *Information theory, inference and learning algorithms*. Cambridge university press.
- Mahabadi, R. K., Henderson, J., & Ruder, S. (2021). Compacter: Efficient low-rank hypercomplex adapter layers. *arXiv preprint arXiv:2106.04647*.
- Mann, W. C., & Thompson, S. A. (1988). Rhetorical structure theory: Toward a functional theory of text organization. *Text-interdisciplinary Journal for the Study of Discourse*, 8(3), 243–281.
- Manning, C., & Schütze, H. (1999). *Foundations of statistical natural language processing*. MIT press.
- Maynez, J., Narayan, S., Bohnet, B., & McDonald, R. (2020, July). On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 1906–1919). Online: Association for Computational Linguistics. Retrieved from <https://www.aclweb.org/anthology/2020.acl-main.173> doi: 10.18653/v1/2020.acl-main.173
- Medhat, W., Hassan, A., & Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams engineering journal*, 5(4), 1093–1113.
- Mei, Q., Ling, X., Wondra, M., Su, H., & Zhai, C. (2007). Topic sentiment mixture: modeling facets and opinions in weblogs. In *Proceedings of the 16th international conference on world wide web* (pp. 171–180).
- Miao, Y., & Blunsom, P. (2016, November). Language as a latent variable: Discrete generative models for sentence compression. In *Proceedings of the 2016 conference on empirical methods in natural language processing* (pp. 319–328). Austin, Texas: Association for Computational Linguistics. Retrieved from <https://aclanthology.org/D16-1031> doi: 10.18653/v1/D16-1031



- Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., & Miller, K. J. (1990). Introduction to wordnet: An on-line lexical database. *International journal of lexicography*, 3(4), 235–244.
- Mnih, A., & Gregor, K. (2014). Neural variational inference and learning in belief networks. In *International conference on machine learning* (pp. 1791–1799).
- Morin, F., & Bengio, Y. (2005). Hierarchical probabilistic neural network language model. *Aistats*, 5, 246–252.
- Moryossef, A., Goldberg, Y., & Dagan, I. (2019a). Step-by-step: Separating planning from realization in neural data-to-text generation. 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)* (pp. 2267–2277). Association for Computational Linguistics. Retrieved from <https://www.aclweb.org/anthology/N19-1236>
- Moryossef, A., Goldberg, Y., & Dagan, I. (2019b). Step-by-step: Separating planning from realization in neural data-to-text generation. 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)* (pp. 2267–2277). Association for Computational Linguistics. Retrieved from <https://www.aclweb.org/anthology/N19-1236>
- Murray, G., Hoque, E., & Carenini, G. (2017). Chapter 11 - opinion summarization and visualization. In F. A. Pozzi, E. Fersini, E. Messina, & B. Liu (Eds.), *Sentiment analysis in social networks* (p. 171-187). Boston: Morgan Kaufmann. Retrieved from <https://www.sciencedirect.com/science/article/pii/B9780128044124000115> doi: <https://doi.org/10.1016/B978-0-12-804412-4.00011-5>
- Nallapati, R., Zhou, B., dos Santos, C., Guçehre, Ç., & Xiang, B. (2016, August). Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of the 20th SIGNLL conference on computational natural language learning* (pp. 280–290). Berlin, Germany: Association for Computational Linguistics. Retrieved from <https://www.aclweb.org/anthology/K16-1028> doi: 10.18653/v1/K16-1028
- Narayan, S., Cohen, S. B., & Lapata, M. (2018, October-November). Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 conference on empirical methods in natural language processing* (pp. 1797–1807). Brussels, Belgium: Association for Computational Linguistics. Retrieved from <https://www.aclweb.org/anthology/D18-1206> doi: 10.18653/v1/D18-1206

- Narayan, S., Zhao, Y., Maynez, J., Simoes, G., & McDonald, R. (2021). Planning with entity chains for abstractive summarization. *arXiv preprint arXiv:2104.07606*.
- Nguyen, T. N. A., Shen, M., & Hovsepian, K. (2021, August). Unsupervised class-specific abstractive summarization of customer reviews. In *Proceedings of the 4th workshop on e-commerce and nlp* (pp. 88–100). Online: Association for Computational Linguistics. Retrieved from <https://aclanthology.org/2021.ecnlp-1.11> doi: 10.18653/v1/2021.ecnlp-1.11
- Ni, J., Li, J., & McAuley, J. (2019a, November). Justifying recommendations using distantly-labeled reviews and fine-grained aspects. 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)* (pp. 188–197). Hong Kong, China: Association for Computational Linguistics. Retrieved from <https://aclanthology.org/D19-1018> doi: 10.18653/v1/D19-1018
- Ni, J., Li, J., & McAuley, J. (2019b, November). Justifying recommendations using distantly-labeled reviews and fine-grained aspects. 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)* (pp. 188–197). Hong Kong, China: Association for Computational Linguistics. Retrieved from <https://www.aclweb.org/anthology/D19-1018> doi: 10.18653/v1/D19-1018
- Orme, B. (2009). Maxdiff analysis: Simple counting, individual-level logit, and hb. *Sequim, WA: Sawtooth Software*.
- Oved, N., & Levy, R. (2021). Pass: Perturb-and-select summarizer for product reviews. In *Acl/ijcnlp*.
- Page, L., Brin, S., Motwani, R., & Winograd, T. (1999). *The pagerank citation ranking: Bringing order to the web*. (Tech. Rep.). Stanford InfoLab.
- Pang, B., & Lee, L. (2004, July). A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd annual meeting of the association for computational linguistics (ACL-04)* (pp. 271–278). Barcelona, Spain. Retrieved from <https://aclanthology.org/P04-1035> doi: 10.3115/1218955.1218990
- Pang, B., Lee, L., & Vaithyanathan, S. (2002, July). Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of the 2002 conference on empirical methods in natural language processing (EMNLP 2002)* (pp. 79–86). Association for Computational Linguistics. Retrieved from <https://aclanthology.org/W02-1011> doi: 10.3115/1118693.1118704

- Pasunuru, R., Liu, M., Bansal, M., Ravi, S., & Dreyer, M. (2021, June). Efficiently summarizing text and graph encodings of multi-document clusters. In *Proceedings of the 2021 conference of the north american chapter of the association for computational linguistics: Human language technologies* (pp. 4768–4779). Online: Association for Computational Linguistics. Retrieved from <https://aclanthology.org/2021.naacl-main.380> doi: 10.18653/v1/2021.naacl-main.380
- Paulus, R., Xiong, C., & Socher, R. (2017). A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304*.
- Pearl, J., & Mackenzie, D. (2018). *The book of why: the new science of cause and effect*. Basic books.
- Pereyra, G., Tucker, G., Chorowski, J., Kaiser, Ł., & Hinton, G. (2017). Regularizing neural networks by penalizing confident output distributions. *arXiv preprint arXiv:1701.06548*.
- Ponti, E. M., Sordoni, A., & Reddy, S. (2022). Combining modular skills in multitask learning. *arXiv preprint arXiv:2202.13914*.
- Poth, C., Pfeiffer, J., Rücklé, A., & Gurevych, I. (2021). What to pre-train on? efficient intermediate task selection. *arXiv preprint arXiv:2104.08247*.
- Press, O., & Wolf, L. (2017, April). Using the output embedding to improve language models. In *Proceedings of the 15th conference of the European chapter of the association for computational linguistics: Volume 2, short papers* (pp. 157–163). Valencia, Spain: Association for Computational Linguistics. Retrieved from <https://www.aclweb.org/anthology/E17-2025>
- Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., ... Liu, P. J. (2019). Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.
- Razavi, A., Oord, A. v. d., & Vinyals, O. (2019). Generating diverse high-fidelity images with vq-vae-2. In *Advances in neural information processing systems*.
- Rennie, S. J., Marcheret, E., Mroueh, Y., Ross, J., & Goel, V. (2017). Self-critical sequence training for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7008–7024).
- Ross, B. C. (2014). Mutual information between discrete and continuous data sets. *PloS one*, 9(2), e87357.
- Rush, A. M., Chopra, S., & Weston, J. (2015, September). A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 conference on empirical methods in natural language processing* (pp. 379–389). Lisbon, Portugal: Association for Computational Linguistics. Retrieved from <https://www.aclweb.org/anthology/D15-1044> doi: 10.18653/v1/D15-1044

- Sandhaus, E. (2008). The new york times annotated corpus. *Linguistic Data Consortium, Philadelphia*.
- Scialom, T., Lamprier, S., Piwowarski, B., & Staiano, J. (2019, November). Answers unite! unsupervised metrics for reinforced summarization models. 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)* (pp. 3246–3256). Hong Kong, China: Association for Computational Linguistics. Retrieved from <https://aclanthology.org/D19-1320> doi: 10.18653/v1/D19-1320
- See, A., Liu, P. J., & Manning, C. D. (2017, July). Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 1073–1083). Vancouver, Canada: Association for Computational Linguistics. Retrieved from <https://aclanthology.org/P17-1099> doi: 10.18653/v1/P17-1099
- Sennrich, R., Haddow, B., & Birch, A. (2016, August). Neural machine translation of rare words with subword units. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 1715–1725). Berlin, Germany: Association for Computational Linguistics. Retrieved from <https://www.aclweb.org/anthology/P16-1162> doi: 10.18653/v1/P16-1162
- Shen, X., Suzuki, J., Inui, K., Su, H., Klakow, D., & Sekine, S. (2019, November). Select and attend: Towards controllable content selection in text generation. 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)* (pp. 579–590). Hong Kong, China: Association for Computational Linguistics. Retrieved from <https://aclanthology.org/D19-1054> doi: 10.18653/v1/D19-1054
- Stiennon, N., Ouyang, L., Wu, J., Ziegler, D., Lowe, R., Voss, C., ... Christiano, P. F. (2020). Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33, 3008–3021.
- Suhara, Y., Wang, X., Angelidis, S., & Tan, W.-C. (2020a). Opiniondigest: A simple framework for opinion summarization. *Proceedings of Association for Computational Linguistics (ACL)*.
- Suhara, Y., Wang, X., Angelidis, S., & Tan, W.-C. (2020b). OpinionDigest: A simple framework for opinion summarization. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 5789–5798). Retrieved from <https://www.aclweb.org/anthology/2020.acl-main.513> doi: 10.18653/v1/2020.acl-main.513
- Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems* (pp. 3104–3112).

- Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.
- Tay, W., Joshi, A., Zhang, X., Karimi, S., & Wan, S. (2019, 4–6 December). Red-faced ROUGE: Examining the suitability of ROUGE for opinion summary evaluation. In *Proceedings of the the 17th annual workshop of the australasian language technology association* (pp. 52–60). Sydney, Australia: Australasian Language Technology Association. Retrieved from <https://www.aclweb.org/anthology/U19-1008>
- Titov, I., & McDonald, R. (2008). Modeling online reviews with multi-grain topic models. In *Proceedings of the 17th international conference on world wide web* (pp. 111–120).
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., . . . Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998–6008).
- Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al. (2016). Matching networks for one shot learning. In *Advances in neural information processing systems* (pp. 3630–3638).
- Vogler, N., Li, S., Xu, Y., Mi, Y., & Berg-Kirkpatrick, T. (2022). An unsupervised masking objective for abstractive multi-document news summarization. *arXiv preprint arXiv:2201.02321*.
- Wang, A., Cho, K., & Lewis, M. (2020a). Asking and answering questions to evaluate the factual consistency of summaries. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Retrieved from <http://dx.doi.org/10.18653/v1/2020.acl-main.450> doi: 10.18653/v1/2020.acl-main.450
- Wang, A., Cho, K., & Lewis, M. (2020b, July). Asking and answering questions to evaluate the factual consistency of summaries. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 5008–5020). Online: Association for Computational Linguistics. Retrieved from <https://aclanthology.org/2020.acl-main.450> doi: 10.18653/v1/2020.acl-main.450
- Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4), 229–256.
- Williams, R. J., & Zipser, D. (1989). A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2), 270–280.
- Xu, Y., & Lapata, M. (2020). Coarse-to-fine query focused multi-document summarization. In *Proceedings of the 2020 conference on empirical methods in natural language processing (emnlp)* (pp. 3632–3645).
- Xu, Y., & Lapata, M. (2021). Text summarization with latent queries. *arXiv preprint arXiv:2106.00104*.

- Yu, T., Liu, Z., & Fung, P. (2021). Adaptsum: Towards low-resource domain adaptation for abstractive summarization. In *arxiv preprint arxiv:2103.11332*.
- Zaheer, M., Guruganesh, G., Dubey, K. A., Ainslie, J., Alberti, C., Ontanon, S., . . . others (2020). Big bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems*, 33, 17283–17297.
- Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R. R., & Smola, A. J. (2017). Deep sets. In *Advances in neural information processing systems* (pp. 3391–3401).
- Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., & Artzi, Y. (2018). Bertscore: Evaluating text generation with bert.
- Zhang, Y., Lai, G., Zhang, M., Zhang, Y., Liu, Y., & Ma, S. (2014). Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proceedings of the 37th international acm sigir conference on research & development in information retrieval* (pp. 83–92).