THE UNIVERSITY
of EDINBURGH

Opinion Summarization of Multiple Reviews: Data Synthesis and Modeling

Reinald Kim Amplayo

Doctor of Philosophy

Institute for Language, Cognition and Computation
School of Informatics
University of Edinburgh

2022

Abstract

The proliferation of online reviews has accelerated research on opinion mining, where the ultimate goal is to glean information from reviews which help users make decisions more efficiently. While opinion mining has assumed several facets in the literature (e.g., sentiment analysis, aspect extraction, etc.), *opinion summarization*, or the task of automatically creating a textual summary of opinions found in multiple reviews, aims to help users access content and improves their decision making. This thesis focuses on different methods to generate opinion summaries given multiple reviews about a target entity (e.g., a product or service). The task is challenging due to the absence of large-scale datasets for supervised training, which is paramount to the recent success of neural-based systems. In this thesis, we propose several methods to synthesize these datasets, thereby making supervised training for opinion summarization feasible.

Firstly, we introduce a two-step process that creates *synthetic datasets* for opinion summarization. Given a corpus of reviews, we first sample a review and pretend it is a (pseudo-)summary. Then, we procure a list of reviews to pair with the summary. We obtain these reviews by generating noisy versions of the summary. We propose a summarization model which learns to denoise the input reviews and generate the summary, motivated by how humans write opinion summaries by removing divergent opinions from reviews. Extensive evaluation shows that our model brings substantial improvements over unsupervised abstractive and extractive baselines.

To further reflect the diversity of opinions in naturally-occurring reviews, we incorporate *content planning* during synthetic dataset creation. For each pseudo-summary sampled from the corpus, we automatically induce its content plan in the form of aspect and sentiment distributions. We then sample reviews from the corpus using Dirichlet distributions parameterized by the content plan, and controlling the variance accordingly. Experimental results show that our approach outperforms competitive models in generating opinion summaries that capture opinion consensus.

In opinion summarization, the notion of salience in reviews largely depends on user interest, therefore a generic summary may not satisfy the needs of all users, limiting their ability to make decisions. Therefore, we extend opinion summarization to generating aspect-controllable summaries. Using a synthetic training dataset enriched with *aspect controllers* of different granularity, we fine-tune a pre-trained language model which allows the creation of generic and aspect-specific summaries by modifying aspect controllers during inference. Experiments show that our model achieves state of the art and is able to generate personalized summaries.

Acknowledgements

First and foremost, I would like to thank my principal supervisor, Mirella Lapata, for her unlimited support throughout my PhD. She has provided me knowledge and wisdom that are necessary to complete my degree, and most importantly, she has given me emotional and mental support in difficult and depressing times, especially during the pandemic. This thesis would not be possible without her constant guidance and feedback.

I am also very grateful to have good people around me in the School of Informatics, and ILCC in particular. I especially would like to thank Hao and Ratish for being helpful and kind officemates, Arthur, Bailin, and Bowen for the occasional lunch-outs, and finally to all the intelligent folks in Mirella's cohort for their insightful comments to improve my work: Jiangming, Jonathan, Laura, Nelly, Stefanos, Tom H, Tom S, and Yang. Without everyone, I could not imagine myself surviving this long and difficult process.

Thank you to all my Filipino and Korean friends who have stayed in touch in the past four years. *Maraming salamat* to Tobeng and high school friends, Ralph/Jake and university friends, Ja, and Ranel. *Kamsahamnida* to Haeju, Kyungjae, Cheoneum, Hyeon-gu, and Juae for the fun times after conferences. A special shout-out to Taeuk, who has always responded quickly to all my messages, which may sometimes be annoying, despite being busy with his own life.

A huge thank you to my distant relatives living in the UK, to the Brind family (Aunt Sal, Paul, Chloe, and Charlotte) for their hospitality and for treating me as their son, and to Rodie who constantly checked up on me.

Last but not least, I want to thank my mom Merlita, my brother Raymarc, and my sister Murielle for their unconditional love and support to my endeavors. All things would not have been possible without them.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Reinald Kim Amplayo*)

To the best mom in the world.

Table of Contents

Chapter 1

Introduction

Since the invention of the Internet, the amount of online information has increased exponentially, which in turn has changed how we access and utilize information. Most of us now read online news articles instead of newspapers, use search engines instead of public libraries to find information, and express and share our opinions through social media, personal blogs, and product reviews. This thesis focuses on product reviews, which are written by customers and contain their personal opinions about a certain product, service, or entity. The past few years have seen the proliferation of such online reviews. Yelp, a business review website, has surpassed 200 million reviews in 2019, which is two times its number of reviews in 2016.[1] The same trend can be observed with Amazon product reviews, where we have seen a 90 million increase in total reviews between 2014 and 2018 (McAuley et al., 2015; Ni et al., 2019). Several studies have shown that consumers use online reviews mainly to consult the opinions and experiences of other consumers for decision making (Chatterjee, 2001), favoring products with higher ratings and those with more reviews (Chevalier and Mayzlin, 2006; Hu et al., 2008; Duan et al., 2008).

However, the volume of information found in online reviews has exceeded the processing capacity of users. This results to information overload (Malhotra, 1984), which can have dysfunctional consequences such as stress and anxiety (Eppler and Mengis, 2008), as well as economical consequences, as consumers are intimidated by the sheer amount of reviews and decide not to buy anything in the end (Soto-Acosta et al., 2014). Due to this, online users tend to rely on a very limited number of reviews in making purchase decisions, which leads to bad decision making (Kwon et al., 2015).

This problem has given rise to a new field of study in the Natural Language Process-

[1] https://www.yelp-press.com/company/fast-facts/default.aspx

ing (NLP) community called *opinion mining* (Pang and Lee, 2008; Liu, 2012), which aims to automatically analyze user opinions and sentiments towards entities such as products and services, which are largely found in online reviews. Opinion mining can be divided into the following four tasks (see Figure 1.1 where we use a set of TV reviews as a running example):

1. **Opinion Extraction** (Dave et al., 2003): Given a review about an entity (e.g., a movie or a hotel), identify the sentences in the review that are opinionated. The task can also be formulated as subjectivity classification (Pang and Lee, 2004), i.e., classify whether a sentence is subjective or objective. For example, in Figure 1.1, the first sentence of the review does not include information that positively or negatively describes the television. All the other sentences express opinions about the television.

2. **Sentiment Classification** (Pang et al., 2002): Given an opinionated text, classify its sentiment. The sentiment can be a polarity (e.g., positive or negative), transforming the task into polarity classification (Pang et al., 2002), or a score in a scale (e.g., from 1 to 5 stars), transforming it into review rating prediction (Pang and Lee, 2005). Figure 1.1 shows example opinions that are classified as positive, negative, and three stars on a five-star scale. This task can be also combined with subjectivity classification as a three-way sentiment classification problem, where non-subjective sentences are classified as *neutral* (Socher et al., 2013).

3. **Aspect Detection** (Popescu and Etzioni, 2005): Given an opinionated text about an entity, determine the aspects of the entity (e.g., picture and sound quality of televisions) the opinion is about. In our running example in Figure 1.1, the opinionated sentences are classified with their respective aspects. This task can be combined with sentiment analysis as an aspect-based sentiment classification problem (Thet et al., 2010). Another reformulation of the task is aspect extraction (Mukherjee and Liu, 2012), where the goal is to extract spans in a text that describe the target aspect.

4. **Opinion Summarization** (Hu and Liu, 2004b): Given a set of reviews about an entity, produce a "summary" of opinions found in the reviews.

It can be argued that among the four tasks above, opinion summarization is the ultimate goal of opinion mining. In fact, opinion mining is often viewed as a pipeline

Input	Output
Task: Opinion Extraction	
TV Review	
1: I bought this TV for my parents.	`not opinion`
2: The color and definition are excellent.	`opinion`
3: It doesn't recognize new ports.	`opinion`
4: The speakers sound okay.	`opinion`
Task: Sentiment Classification	
The color and definition are excellent.	`positive`
It doesn't recognize new ports.	`negative`
The speakers sound okay.	★★★☆☆
Task: Aspect Detection	
The color and definition are excellent.	`picture quality`
It doesn't recognize new ports.	`connectivity`
The speakers sound okay.	`sound quality`
Task: Opinion Summarization	

Review 1:
I bought this TV for my parents. The color and definition are excellent. It doesn't recognize new ports. The speakers sound okay.
Review 2:
Overall, it was okay. Picture quality is perfect, but everything else is below average. I find the sound quality very loud and poor. There are limited ports available as well.
Review 3:
I got my TV last week, very fast delivery. I find the picture and sound quality to be good. I hope there are HDMI and USB ports available though.
Review 4:
...
	(a) **Aspect-Sentiment Table**
	`connectivity`: 1.3/5
	`picture quality`: 4.7/5
	`sound quality`: 2.4/5
	(b) **Clustered Opinions**
	`picture quality`
	- The color and definition are excellent.
	- Picture quality is perfect.
	- *(40 more sentences)*
	`connectivity`
	- It doesn't recognize new ports.
	- No HDMI ports!
	- *(25 more sentences)*
	(c) **Textual Summary**
	The TV has an excellent picture quality. The speakers can be loud for some and the ports may be lacking.

Figure 1.1: Example input and output data for four opinion mining subtasks. For opinion summaries, we provide three different kinds of output formats.

of tasks leading to opinion summarization (Hu and Liu, 2004b; Gamon et al., 2005; Popescu and Etzioni, 2005; Angelidis and Lapata, 2018b). In other words, opinion mining systems first extract opinionated text from reviews, then classify the sentiment and aspect of this text, and finally summarize the results into a summary. Opinion summaries have assumed various formats throughout the history of opinion mining. Earlier work has viewed opinion summaries as aspect-sentiment tables (see (a) in Figure 1.1; Titov and McDonald, 2008; Wu et al., 2016a; Amplayo and Song, 2017) and clusters of a ranked list of opinions (see (b) in Figure 1.1; Hu and Liu, 2004b; Blair-Goldensohn et al., 2008; Lerman et al., 2009). More recent work generates natural language text (Carenini et al., 2006; Ganesan et al., 2010; Wang and Ling, 2016; Angelidis and Lapata, 2018b) to represent an opinion summary. In fact, it can be argued that textual summaries are the most effective solution for opinion mining, especially when it comes to making decisions. Non-textual opinion summaries are usually treated as data visualization tools to analyze reviews, while textual summaries are easily understandable outputs that can be presented to users (Murray et al., 2017; Moussa et al., 2018). Textual summaries provide information that is both brief and comprehensible, unlike other forms of opinion summaries which are either insufficient (e.g., a ratings table that does not explain why the product is rated as such) or overloaded information (e.g., a large list of opinionated segments that users can hardly finish reading). In this thesis, we focus on developing opinion summarization models that generate textual summaries that help users in decision making.

Opinionated text aside, the summarization of news articles has been the object of intense study (Nenkova and McKeown, 2012; Rush et al., 2015; See et al., 2017; Liu and Lapata, 2019b), with most recent work developing effective methods to train deep neural networks (Sutskever et al., 2014; Bahdanau et al., 2014; Vinyals et al., 2015) to generate coherent and grammatical summaries. Training is often in the form of supervised learning, i.e., large-scale datasets consisting of document-summary pairs. The objective of the neural network is then to maximize the likelihood of the model to generate the summary as output given the input document. Luckily for news articles, such pairs are widely available on the web (e.g., CNN and DailyMail provide summaries to their articles which are often produced by the writers themselves; Hermann et al., 2015).

Unfortunately, despite the proliferation of online reviews, opinion summaries are neither freely available nor easy to obtain. In order for a person to effectively write opinion summaries, they would need to digest the content of multiple reviews, which

they have not written themselves. Since the number of products and their corresponding reviews can be thousands, this manual annotation task is very time consuming and cannot be streamlined and thus can be considered practically impossible. For this reason, neural approaches have relied on either unsupervised architectures (Chu and Liu, 2019; Coavoux et al., 2019; Suhara et al., 2020) which have severe limitations in terms of generative power and can be hard to train, or on transfer learning methods (Zhang et al., 2018b; Chen and Shuai, 2021) that require models to learn domain-specific knowledge which can be diverse in our setting (we refer readers to Chapter 2.2 for a detailed literature review). Finally, opinion summarization is a *personalized* task. Different users care for different aspects of products. This would require different summaries for users with contrasting preferences.

In this thesis, we present a framework to create synthetic supervision for training neural methods that are effective for opinion summarization. We propose different ways to create synthetic training datasets consisting of review-summary pairs that are linguistically motivated, resemble real-world data, and work for different review domains (movies, products, restaurants, and hotels), and different kinds of opinion summaries (generic, aspect-specific, and personalized). Finally, our proposed synthetic dataset creation methods rely only on review data, which are freely available on a large-scale.

1.1 Thesis Statement

This thesis investigates a series of hypotheses relating to the summarization of opinions in multiple reviews, which we test through designing models and evaluating them through extensive experiments.

> **DATA SYNTHESIS HYPOTHESIS**:
> Training datasets for opinion summarization can be synthesized from information freely available in online reviews with minimal domain knowledge.

We show that in opinion summarization, the absence of training data does not hinder supervised learning. We propose ways of creating synthetic datasets following how real-world data is constructed and how humans write summaries. These datasets can then be used to train neural network models effectively to *generate* opinion summaries.

> **CONTENT PLANNING HYPOTHESIS**:
> Opinions across reviews vary widely and can be conflicting. Modeling

these variations through the use of content plans allows for generation of summaries that capture opinion consensus.

Reviews about a certain entity do not necessarily agree with each other; some users may like the sound quality of a television, but others may not. We propose to model opinion variation in reviews using content plans in the form of aspect and sentiment distributions. We show that incorporating content planning in opinion summarization not only yields output of higher quality, but also allows the creation of naturalistic synthetic datasets.

ASPECT CONTROL HYPOTHESIS:
Opinion summaries may vary across different users and their preferences. Aspect controllers of different granularity enable summarization systems to produce personalized summaries.

It is generally assumed that a single generic summary is sufficient and the majority of users have the *same information need*. This thesis argues that this assumption does not apply to opinion summarization. We design a summarization framework that can generate personalized opinion summaries with very little input from the users (essentially a few keywords indicating the aspects of interest).

1.2 Contributions

In this thesis, we propose several novel solutions for creating synthetic datasets for opinion summarization, as well as several models for the aforementioned task. We summarize our main contributions below:

Synthetic Dataset Creation We propose a step-by-step framework to create synthetic datasets for training supervised neural opinion summarization models. Given a corpus of reviews, we construct pseudo-pairs consisting of a set of reviews and their corresponding opinion summary on a large scale. We propose three different methods for synthetic dataset creation representing different facets of the opinion summarization task. Firstly, we use noise generating functions to create synthetic datasets motivated by how humans generate summaries (Chapter 3). Secondly, we incorporate content planning to control opinion variance among reviews in order to create more naturalistic datasets (Chapter 4). And finally, we leverage aspect controllers of different granularity to create synthetic datasets and train aspect-controllable opinion summarization models (Chapter 5).

Content Plans for Opinion Summaries To the best of our knowledge, we are the first to introduce content planning for opinion summarization, a major component of traditional natural language generation systems (McKeown, 1985). We use aspect and sentiment distributions as content plans for a single review. These distributions can be induced automatically using learned aspect and sentiment embeddings. Unlike traditional discrete content plans for other domains and tasks, our content plan is not discrete and is in the form of latent multinomial distributions, which makes its variance easily controllable.

Opinion Summarization We present two neural architectures for the generation of opinion summaries. Firstly, our DENOISESUM model is trained to remove non-salient opinions (i.e., noise) from the reviews and generate a consensus summary of opinions. We introduce three modules in DENOISESUM: (a) explicit denoising guides how the model removes noise from the input, (b) partial copy enables to copy information from the input only when necessary, and (c) a discriminator helps the decoder generate topically consistent text. Secondly, our PLANSUM model incorporates content plans to guide generation towards more salient content. In PLANSUM, we propose two new components that help train the model: (a) injective fusion aggregates tokens that retain information regarding token frequency, and (b) LM-based label smoothing extends label smoothing (Szegedy et al., 2016) by incorporating pre-trained language models such as BERT (Devlin et al., 2019)

Aspect-Controllable Summarization We learn to generate personalized summaries via controlling the various aspects they discuss. That is, we use aspect queries to represent user preferences and generate a personalized summary. Our model, ACESUM, levereges pre-trained language models (Devlin et al., 2019) to generate fluent summaries, as well as automatically induced aspect controllers (e.g., aspect keywords, sentences, and codes) to translate aspect queries into machine-understandable sequences. The model is trained using a synthetic dataset tailored specifically to allow for controllability.

1.3 Thesis Outline

Chapter 2 provides background to the neural network models used in this thesis and the task of opinion summarization. We first discuss LSTMs and Transformers, two

neural models that are widely used for sequence-to-sequence tasks (Sutskever et al., 2014; Hochreiter and Schmidhuber, 1997; Vaswani et al., 2017). We then define the opinion summarization task and review related literature. This chapter aims to familiarize readers with basic concepts and terms used throughout this thesis.

Chapter 3 describes our proposed framework to create synthetic datasets for training in a supervised manner neural models for opinion summarization. We only require a freely available corpus of reviews, in order to construct pseudo-summary and review pairs. We also introduce a synthetic dataset creation method that generates noisy versions of the pseudo-summary and treats these as pseudo-reviews. Finally, we use the synthetic dataset to train our model, which aims to remove the noise from the reviews and recreate the summary. Experimental results on two benchmarks show that our model brings substantial improvements over purely unsupervised methods.

Chapter 4 presents another method to create synthetic datasets that capture opinion variance found among reviews in real-world data. We incorporate content plans of the summary during the dataset creation process, which are used to effectively sample naturalistic reviews from the corpus. We also propose a summarization model that repurposes these content plans as additional input to guide the generation of summaries with salient content. Extensive experiments show that, when trained on our synthetic dataset, our approach outperforms competitive models in generating opinion summaries that capture opinion consensus.

Chapter 5 focuses on the task of aspect-controllable opinion summarization, which introduces personalization in the summaries. The aim is to generate a customized opinion summary based on aspect queries (e.g., describing the location and room of a hotel). We create a synthetic training dataset enriched with aspect controllers which are induced by a multi-instance learning model that predicts the aspects of a document at different levels of granularity. We fine-tune a pre-trained model using our synthetic dataset and generate aspect-specific summaries by modifying the aspect controllers. Experiments on two benchmarks show that our model outperforms the previous state of the art and generates personalized summaries by controlling the number of aspects discussed in them.

Chapter 6 summarizes our work and discusses interesting directions for future work.

1.4 Published Work

Portions of this thesis have been previously published in Amplayo and Lapata (2021) (Chapter 2), Amplayo and Lapata (2020) (Chapter 3), Amplayo et al. (2021b) (Chapter 4), and Amplayo et al. (2021a) (Chapter 5).

Chapter 2

Background

The methods we present in this thesis use neural network models following their recent success on various Natural Language Processing (NLP) tasks. In this chapter, we first provide background for two commonly used neural network models, Recurrent Neural Networks (RNNs; Elman, 1990; Hochreiter and Schmidhuber, 1997) and Transformers (Vaswani et al., 2017). We also introduce the encoder-decoder architecture (Sutskever et al., 2014), and the concept of language model pretraining (Ramachandran et al., 2017), a widely used method to improve the performance of NLP tasks. Finally, we formally define the opinion summarization problem, discuss related work that uses three kinds of approaches to generate summaries, and describe various opinion summarization datasets, and how system summaries are evaluated.

2.1 Neural Networks

Given input text in the form of a sequence of tokens[1] $W = [w_1, w_2, ..., w_n]$, neural network models learn rich textual representations in the form of vectors (Hinton et al., 1986). This usually starts by transforming W into dense and low-dimensional vectors $X = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n]$ called token embeddings (Mikolov et al., 2013; Pennington et al., 2014), where $\mathbf{x}_i \in \mathbb{R}^d$. Token embeddings X are then fed into neural networks that learn richer contextualized representations based on relationships among different embeddings. Training neural networks is usually done with backpropagation (Rumelhart et al., 1986), which computes the gradient of the objective function with respect to the

[1]Tokens can refer to both words or *subwords* (Sennrich et al., 2016), which are substrings of long and/or uncommon words split into commonly used strings to limit the vocabulary size. For example, the word "subword" may be split into its subwords "sub" and "words."

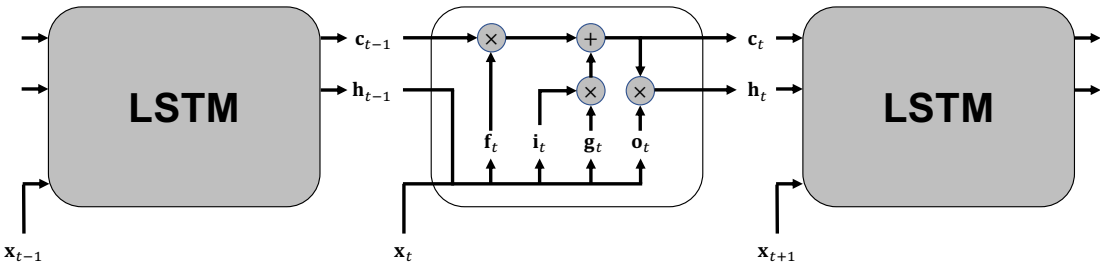


Figure 2.1: Long Short-Term Memory (LSTM) network with three time steps. At each time step t, input $\mathbf{x}_t$ is processed using previous hidden vector $\mathbf{h}_{t-1}$ and memory cell $\mathbf{c}_{t-1}$. The LSTM uses three gates $\mathbf{f}_t$, $\mathbf{i}_t$, and $\mathbf{o}_t$ to control the flow of information from one time step to the other. The operations $\otimes$ and $\oplus$ refer to broadcast multiplication and addition.

weights in the network, and updates them accordingly. In this section, we describe different types of neural network models commonly used in NLP.

2.1.1 Recurrent Neural Networks

Given a sequence of input token vectors $X = [\mathbf{x}_1, ..., \mathbf{x}_n]$, Recurrent Neural Networks (RNNs) model the input in a sequential manner. That is, an RNN includes a hidden state vector $\mathbf{h} \in \mathbb{R}^d$ that is updated each time it receives a vector from the input. Formally, at time step t, RNNs take the tth vector $\mathbf{x}_t$ as input and compute a hidden state vector $\mathbf{h}_t$ using a non-linear function of $\mathbf{x}_t$ and the previous value of the hidden state vector $\mathbf{h}_{t-1}$:

$$\mathbf{h}_t = f(\mathbf{x}_t, \mathbf{h}_{t-1}) \tag{2.1}$$

The function f is usually a linear combination of the input (i.e., Elman network, Elman, 1990):

$$\mathbf{h}_t = \sigma(\mathbf{W}_1\mathbf{x}_t + \mathbf{W}_2\mathbf{h}_{t-1} + \mathbf{b}) \tag{2.2}$$

where σ is a non-linear activation function, $\mathbf{W}_1 \in \mathbb{R}^{d\times d}$ and $\mathbf{W}_2 \in \mathbb{R}^{d\times d}$ are weight matrices, and $\mathbf{b} \in \mathbb{R}^d$ is a bias term. The output is a list of hidden state vectors $H = [\mathbf{h}_1, ..., \mathbf{h}_n]$, where $\mathbf{h}_t$ is the updated representation of the tth word contextualized using previous words.

RNNs are prone to the vanishing gradient problem (Kolen and Kremer, 2001), since gradients become extremely small over time during backpropagation. To miti-

gate this problem, a special kind of RNNs called Long Short-Term Memory Networks (LSTMs; Hochreiter and Schmidhuber, 1997) is widely used. LSTMs introduce multiple gating mechanisms and a memory cell to alleviate the vanishing gradient problem. Formally, let $\mathbf{c}_t \in \mathbb{R}^d$ be the memory cell at time step t; $\mathbf{c}_t$ is calculated using three gates: (1) $\mathbf{i}_t$ controls the amount of information to put into the memory from the input, (2) $\mathbf{f}_t$ controls the amount of information to forget and remove from the memory, and (3) $\mathbf{o}_t$ controls the amount of information to output into the hidden state vector:

$$\begin{bmatrix} \mathbf{g}_t \\ \mathbf{i}_t \\ \mathbf{f}_t \\ \mathbf{o}_t \end{bmatrix} = \begin{bmatrix} \tanh \\ \text{sigm} \\ \text{sigm} \\ \text{sigm} \end{bmatrix} \mathbf{W}[\mathbf{x}_t; \mathbf{h}_{t-1}] + \mathbf{b} \tag{2.3}$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{g}_t \tag{2.4}$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t) \tag{2.5}$$

where tanh and sigm are hyporbolic tangent and sigmoid operators, $\odot$ is element-wise multiplication, and $[\cdot;\cdot]$ is vector concatenation. Weight matrix $\mathbf{W} \in \mathbb{R}^{4d \times 2d}$ and bias vector $\mathbf{b} \in \mathbb{R}^{4d}$ are the parameters used to obtain these gates, and $\mathbf{g}_t$ is a vector that represents information from the current input $\mathbf{x}_t$. We illustrate LSTMs in Figure 2.1.

2.1.2 Transformers

The temporal nature of RNNs constrains the processing of input vectors to be sequential, i.e., the next input must wait for the current input to be processed. This makes parallelization, a trait that is essential to train neural networks efficiently in GPUs/TPUs, very difficult. To remove this constraint, Vaswani et al. (2017) introduce a new architecture called Transformer that replaces RNNs for text modeling. Transformer makes use of a self-attention mechanism in lieu of the recurrent structure, where each word is updated by aggregating information from all other words simultaneously.

Formally, given input token vectors $X = [\mathbf{x}_1, ..., \mathbf{x}_n]$, the Transformer is a stack of L identical layers, where each layer has two sub-layers:

$$\tilde{H}^{(l)} = \text{MultiHeadAttention}(H^{(l-1)}) \tag{2.6}$$

$$H^{(l)} = \text{FeedForward}(\tilde{H}^{(l)}) \tag{2.7}$$

where $H^{(l)} = [\mathbf{h}_1^{(l)}, ..., \mathbf{h}_n^{(l)}]$ is the set of hidden state vectors at lth layer and $H^{(0)} = \text{PositionalEmbeddings}(X)$. We discuss all these modules in depth in the next sections.

Positional Embeddings To recuperate the lack of recurrence when modeling text, Transformer uses positional embeddings to explicitly distinguish tokens with different positions in the text:

$$\mathbf{h}_t^{(0)} = \mathbf{x}_t + \mathbf{p}_t \tag{2.8}$$

where $\mathbf{p}_t \in \mathbb{R}^d$ is the positional embedding for the word in the tth position in the input text. These embeddings are represented using sine and cosine functions with different frequencies to distinguish different input positions:

$$\mathbf{p}_t[2i] = \sin(t/10000^{2i/d}) \tag{2.9}$$

$$\mathbf{p}_t[2i+1] = \cos(t/10000^{2i/d}) \tag{2.10}$$

where $\mathbf{p}_t[i]$ represents the ith dimension of the vector. Each dimension of $\mathbf{p}_t$ corresponds to a sinusoid, thus all positional embeddings are distinguishable from each other. This characteristic is also useful to easily learn to attend by relative positions of the word, since any one positional embedding can be linearly transformed to another.

Multi-head Attention To contextualize token representations using information from other tokens, Transformer uses a multi-head attention mechanism that jointly calculates a weighted sum over all tokens to update representations for each token. For each layer-wise hidden vector $\mathbf{h}_t^{(l)}$, a single-head attention can be calculated as follows:

$$\begin{bmatrix} \mathbf{q}_t \\ \mathbf{k}_t \\ \mathbf{v}_t \end{bmatrix} = \begin{bmatrix} \mathbf{W}_q \\ \mathbf{W}_k \\ \mathbf{W}_v \end{bmatrix} \mathbf{h}_t^{(l)} \tag{2.11}$$

$$\mathbf{head}_t = \text{softmax}(\frac{\mathbf{q}_t \mathbf{k}_t^\top}{\sqrt{d_k}})\mathbf{v}_t \tag{2.12}$$

where $\mathbf{W}_q \in \mathbb{R}^{d_k \times d}$, $\mathbf{W}_k \in \mathbb{R}^{d_k \times d}$, and $\mathbf{W}_v \in \mathbb{R}^{d_k \times d}$ are weights to transform $\mathbf{h}_t$ and obtain the query, key, and value vectors $\mathbf{q}_t$, $\mathbf{k}_t$, and $\mathbf{v}_t$, respectively, and $\mathbf{head}_t \in \mathbb{R}^{d_k}$ is the resulting head-specific vector. A scaling factor of $\sqrt{d_k}$ is used to ensure that the dot-product between the query and key vectors does not grow large in magnitude.

Multi-head attention performs single-head attention K times, producing K head vectors. These head vectors are concatenated and transformed into a single vector:

$$\mathbf{mhead}_t = \mathbf{W}_o[\mathbf{head}_t^1; ...; \mathbf{head}_t^K] \tag{2.13}$$

where $\mathbf{W}_o \in \mathbb{R}^{d_k K \times d}$ and $[\cdot;\cdot]$ is the concatenation operation. Performing attention on multiple heads allows Transformer to jointly attend to information from different representation subspaces (Vaswani et al., 2017).

Feed-forward Networks Multi-head attention is followed by a two-layer feed-forward network, which is two linear transformations with a ReLU (Glorot et al., 2011a) activation function in between:

$$\mathbf{ffn}_t = \mathbf{W}_2 \max(0, \mathbf{W}_1 \mathbf{h}_t + \mathbf{b}_1) + \mathbf{b}_2 \tag{2.14}$$

where $\mathbf{W}_1 \in \mathbb{R}^{d_{ff} \times d}$ and $\mathbf{W}_2 \in \mathbb{R}^{d \times d_{ff}}$ are transformation weights, $\mathbf{b}_1 \in \mathbb{R}^{d_{ff}}$ and $\mathbf{b}_2 \in \mathbb{R}^d$ are biases, and $\max(\cdot)$ is the element-wise maximum operation

Residual Connections and Layer Normalization Finally, Transformer uses residual connections (He et al., 2016) and layer normalization (LayerNorm; Ba et al., 2016) at the end of each sub-layer to make optimization easier and more efficient.

Let $f(\cdot)$ be a sub-layer of a Transformer (i.e., either a multi-head attention or a feed-forward network). Given input $\mathbf{h}$, residual connections add the input to the resulting value when passed to function f:

$$f_{add}(\mathbf{h}) = \mathbf{h} + f(\mathbf{h}) \tag{2.15}$$

where $f_{add}(\cdot)$ is the same function but with a residual connection.

Residual connections are followed by LayerNorm (i.e., layer normalization), where the mean and variance for all dimensions of vector $\mathbf{h}$ are first calculated and then are used to normalize the vector:

$$\mu = \frac{1}{d} \sum_{i=1}^{d} f_{add}(\mathbf{h})[i] \tag{2.16}$$

$$\sigma^2 = \frac{1}{d} \sum_{i=1}^{d} (f_{add}(\mathbf{h})[i] - \mu)^2 \tag{2.17}$$

$$f_{add+norm}(\mathbf{h}) = \frac{f_{add}(\mathbf{h}) - \mu}{\sqrt{\sigma^2 + \varepsilon}} \tag{2.18}$$

where μ and σ^2 are mean and variance vectors, d is the dimension of $f_{add}(\cdot)$, and ε is a small value to prevent division by zero. Figure 2.2 shows an illustration of a Transformer.

2.2 Opinion Summarization

Let C denote a corpus of reviews on a set of entities $E = \{e_1, e_2, ..., e_{|E|}\}$ (e.g., movies, products, hotels, or restaurants). For each entity $e \in E$, the corpus contains a set of reviews $R_e = \{r_1, r_2, ..., r_{|R_e|}\}$ with one or more opinions about entity e, where $r_i =$

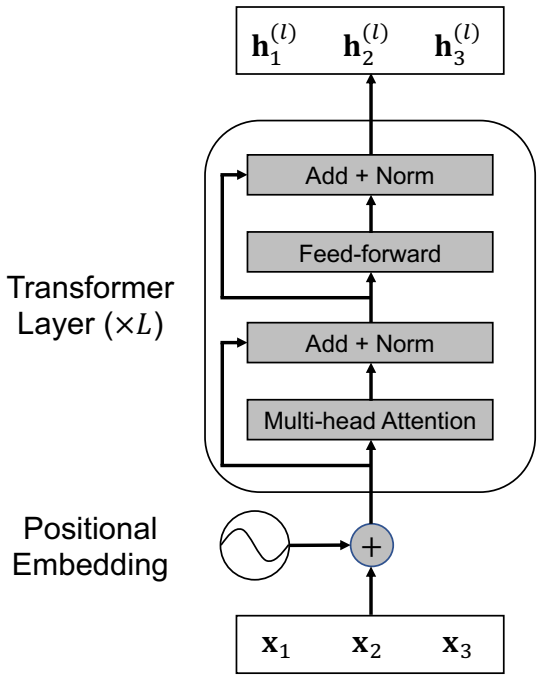


Figure 2.2: A Transformer encoder with L layers. The input token embeddings $\mathbf{x}_i$ and their positional embeddings are first added together. The resulting vectors go through multiple layers of multi-head attention and feed-forward modules to obtain hidden vectors $\mathbf{h}_i^{(l)}$ at lth layer.

$[w_1, ..., w_{|r_i|}]$ is a sequence of tokens. The opinions in these reviews may pertain to at least one of given set of *aspects* $A_e = \{a_1, a_2, ..., a_{|A_e|}\}$ for entity e (e.g., the plot/acting of a movie, the room/food of a hotel). Finally, we assume that for each review r_i, there exists a user-annotated rating s_i, which suggests the overall *sentiment* of r_i and can be either binary (e.g., positive or negative) or on a scale (e.g., from 1 to 5 stars).

The goal of opinion summarization is to generate an opinion summary y_e for each set of reviews R_e of entity $e \in E$. The opinion summary should include information from the reviews that subjectively describes one or more of the aspects. Moreover, this information should be salient, i.e., it should be informative and are mentioned across multiple reviews. There are two types of opinion summary: (a) a general summary that contains salient opinions about *all* aspects of an entity, and (b) an aspect-specific summary that focuses on opinions about *specific* aspects of interest specified by a query $Q = \{q_1, q_2, ..., q_{|A_e|}\}$; here, q_j is an indicator function which designates whether the aspect should be mentioned in the summary or not. The query is usually derived from a user query, e.g., when a user requests for a summary about aspects a_1 and a_2, then q_1 and q_2 is set to 1. Figure 2.3 illustrates example opinion summaries for a bar, where

positive opinions regarding the beer are considered salient, while negative opinions about the same aspect are not salient, even though they are mentioned in the input.

Opinion summarization is an instance of multi-document summarization (MDS; Radev et al., 2000). Early MDS methods were mostly *extractive*, selecting parts of the input to form the summary. These include greedy selection methods (Carbonell and Goldstein, 1998; Nenkova and Vanderwende, 2005), which sequentially select sentences based on various criteria such as importance and redundancy. Graph-based methods such as LexRank (Erkan and Radev, 2004) and TextRank (Mihalcea and Tarau, 2004) are also widely used. These methods create weighted graphs from sentences and calculate importance scores using a PageRank algorithm.[2] Finally, centroid-based clustering methods (Radev et al., 2000, 2004) are also popular, which collate sentences into multiple clusters and select at least one representative sentence from the clusters to form the summary. More recent methods leverage advances in neural networks and deep learning (Mikolov et al., 2013; Sutskever et al., 2014; Bahdanau et al., 2014) to obtain richer dense representations for extractive methods (Cao et al., 2015; Li et al., 2017), as well as to build *abstractive* methods (Liu et al., 2018a; Zhang et al., 2018b; Liu and Lapata, 2019a) that learn how to generate summaries from scratch. The next two sections formally define the extractive and abstractive opinion summarization problems and describe previous methods proposed in the literature.

2.2.1 Extractive Opinion Summarization

Let $S = \{s_1, s_2, ..., s_{|S|}\}$ be the set of all sentences found in input set of reviews R. The goal of extractive opinion summarization is to select a subset of sentences $U = \{u_1, u_2, ..., u_M\}$ that is representative of the opinions found in the reviews. The concatenation of these sentences forms the output opinion summary y, which ideally contains only salient opinions and is concise, maximally informative, and minimally redundant. There are two main modules of extractive summarization: (1) the sentence representation module that transforms sentences into features, and (2) the sentence selection module that uses the feature-based sentence representation to extract the most important ones and create the output summary.

[2]The key difference between LexRank and TextRank is their edge weighting functions, where the former assigns edge weights based on word and phrase similarity, while the latter assumes all edges to be of the same unit weight.

Bar Reviews

1. ... The staff were very friendly and I found the bar a bit like home. ... their beer is quality. ... Their jalapeno pale ale!? Hello deliciousness ...

2. Great beer to try! Fun flavors like jalapeno pale ale. The staff inside is nice and friendly. ...

3. Had the extra pale ale and loved it. ... The vibe was ideal for a long night of serious causal drinking. ... friendly bartenders, this place just felt homey as soon as you sat on a stool. ... this bar has delicious beer and a chill atmosphere that really makes the beer go down quick and easy. ...

4. ... nice happy laid back people and great beer. The jalepeno pale ale was amazen ... The smell and taste wrk great for it, ...

5. Jalapeno pale ale... maybe a little crazy... but so good. ... I have always enjoyed their free will. They have made a couple new brews as of late that I sampled and all are really good. ...

6. ... at least make sure that the beer is cold. I spoke to someone up at bar and she basically said that because the beer is brewed there its not served cold but just slightly colder than room temperature. ...

7. ... the beer is really good. ...

8. Great atmosphere! Was a band on the saturday night we were there that was excellent! The beers were wonderful as well, would be back!

Extractive General Summary

The vibe was ideal for a long night of serious casual drinking. The staff were very friendly and I found the bar a bit like home. Jalapeno pale ale... maybe a little crazy... but so good. The beers were wonderful as well, would be back!

Abstractive General Summary

If you're looking for a comfortable and inviting bar this is a great place to go. They have a lot of unique beers on tap that you will not find anywhere else. The staff here is extremely friendly, and after just a couple of minutes it feels like you are chatting with an old friend. The next time you want to head out for some drinks give them a shot!

Abstractive Aspect-Specific Summary (beer aspect)

The beers here are really good. They have a lot of unique beers on tap that you will not find anywhere else. The jalapeno pale ale is especially very delicious.

Figure 2.3: Examples of different kinds of opinion summaries (extractive general, abstractive general, and abstractive aspect-specific) produced by human annotators for a bar in the Yelp dataset. Colored spans are sentences extracted to form the extractive summary, where each color represents a different aspect (beer, atmosphere, and staff). The underlined sentence refers to a negative opinion about the beer aspect but is not considered salient.

Sentence Representation Before neural networks, sentences were usually represented using hand-picked features that signify their saliency. These include word-level n-grams and part-of-speech tags (Hu and Liu, 2004b,a, 2006; Liu et al., 2005), occurences of aspect-specific words and phrases (Popescu and Etzioni, 2005; Lu et al., 2009), word frequencies (Ku et al., 2006), lexicons and regular expressions (Zhuang et al., 2006), and multinomial distributions from topic models (Mei et al., 2007; Titov and McDonald, 2008).

Hand-picked features can be limited since they do not wholly represent the semantics of sentences. Neural networks and deep learning gave rise to better representation learning (Bengio et al., 2013) which allows us to learn semantic representation of sentences in the form of dense vectors. Angelidis and Lapata (2018b) and Angelidis et al. (2021) represent sentences using a weighted sum of latent aspect embeddings learned using (variational) autoencoders which are essentially neural topic models. Karamanolakis et al. (2019) extend this by using pre-trained language model weights and knowledge distillation techniques. Zhao and Chaturvedi (2020) leverage external domain knowledge to improve aspect representation and consequently create better sentence representation. More recent work (Mukherjee et al., 2020; Wang et al., 2020a) makes use of off-the-shelf sentence-level embeddings pre-trained using large language models and huge datasets (Reimers and Gurevych, 2019).

Sentence Selection The sentence selection stage decides whether sentences should be included in the summary or not. Traditional methods made use of frequency-based metrics such as TF-IDF (Ku et al., 2006), aspect-specific lexicons (Lu et al., 2009), and topical word frequencies from topic models (Mei et al., 2007; Titov and McDonald, 2008). More recent methods use neural models to score sentences based on relevance, polarity, redundancy, and readability (Angelidis and Lapata, 2018b; Zhao and Chaturvedi, 2020; Mukherjee et al., 2020) and rank sentences based on their scores. Angelidis et al. (2021) use a sampling strategy instead of ranking, which allows explicit modeling of opinion popularity.

2.2.2 Abstractive Opinion Summarization

Given input reviews $R = \{r_1, r_2, ..., r_{|R|}\}$, the goal of abstractive opinion summarization is to create a model that generates summary y. This is usually done in a sequence-to-sequence autoregressive manner (Sutskever et al., 2014), that is, we generate tokens

one by one, conditioned on both reviews R and previously predicted summary tokens:

$$p(y|R) = \prod_t p(y_t|R, y_1, ..., y_{t-1}) \tag{2.19}$$

where y_t is the token being predicted at timestep t. The sequential prediction of tokens approximately replicates how humans write summaries *from scratch*; the next content humans write usually depend on both the global context R and the previously written content $y_1, ..., y_{t-1}$.

Prior to neural networks, traditional abstractive methods did not follow an autoregressive generation approach. Instead, they used human-engineered templates, which when filled with relevant content led to abstractive summaries (Carenini et al., 2006; Carenini and Moore, 2006; Di Fabbrizio et al., 2014). With Opinosis (Ganesan et al., 2010), multiple reviews are transformed into a graph with words as nodes, and an abstractive summary is constructed by extracting highly redundant words and phrases from the graph. These methods are not purely abstractive, since the summaries have limited vocabulary and are essentially extracting content from input instead of being generated from scratch. On the other hand, neural models are able to generate abstractive summaries, which can contain novel words that may not be found in the input. The next sections describe neural architectures used for abstractive opinion summarization.

Encoder-Decoder Architecture Neural network models use the encoder-decoder architecture (Sutskever et al., 2014), a modeling framework that first encodes the input text into dense representations, which can be done using either an RNN (Section 2.1.1) or a Transformer (Section 2.1.2) encoder, and then decodes the output text using another neural network module, which can also be an RNN or a Transformer.

More specifically, input token vectors $X = [\mathbf{x}_1, ..., \mathbf{x}_n]$ are passed to the encoder to obtain contextualized hidden vectors $H = [\mathbf{h}_1, ..., \mathbf{h}_n]$:

$$H = \text{Encoder}(X) \tag{2.20}$$

The decoder then uses H as context to predict output token y_t at timestep t, given previous output token y_{t-1}. For each timestep t, given embeddings of previously generated output tokens $Y_{<t} = [\mathbf{y}_1, ..., \mathbf{y}_{t-1}]$, the decoder produces contextualized hidden vectors $G_{<t} = [\mathbf{g}_1, ..., \mathbf{g}_{t-1}]$. The current final hidden vector $\mathbf{g}_{t-1}$ is then passed to a logistic classifier to predict the next token y_t:

$$G_{<t} = \text{Decoder}(Y_{<t}) \tag{2.21}$$

$$p(y_t) = \text{Classifier}(\mathbf{g}_{t-1}) \tag{2.22}$$

where $p(y_t)$ is the probability distribution to predict the output token y_t.

Moreover, there are two training mechanisms widely used to improve model performance. Firstly, at each timestep t, input hidden vectors H are usually reduced into a single context vector $\tilde{\mathbf{h}}_{t-1}$ via an **attention mechanism** (Bahdanau et al., 2014), which is used as additional input to the classifier:

$$p_{att}(y_t) = \text{Classifier}(\mathbf{g}_{t-1}, \tilde{\mathbf{h}}_{t-1}) \tag{2.23}$$

$$\tilde{\mathbf{h}}_{t-1} = \sum_i \alpha_i * f_{value}(\mathbf{h}_i) \tag{2.24}$$

$$\alpha_i = \text{softmax}(\frac{f_{query}(\mathbf{g}_{t-1}) f_{key}(\mathbf{h}_i)^\top}{scale}) \tag{2.25}$$

where α_i is the weight signifying the importance of the ith input token. The functions f_{query}, f_{key}, and f_{value} transform the hidden vectors into query, key, and value vectors, which can be either an identity function (Bahdanau et al., 2014) or a linear transform (Vaswani et al., 2017). The scale is usually the square root of the dimensions of the vectors. Finally, the attention mechanism can also be of multiple heads, as in Vaswani et al. (2017).

Secondly, at each timestep t, a **copy mechnism** (Vinyals et al., 2015; See et al., 2017) is employed over the input tokens to produce the final probability distribution $p_{att+copy}(y_t)$ as a weighted sum over the token generation probability $p_{att}(y_t)$ and the probability $p_{copy}(y_t)$ to copy directly from the set of input tokens X:

$$p_{att+copy}(y_t) = \sigma_{copy} * p_{copy}(y_t) + (1 - \sigma_{copy}) * p_{att}(y_t) \tag{2.26}$$

$$\sigma_{copy} = \text{sigm}(f_{copy}(\mathbf{y}_{t-1}, \mathbf{g}_{t-1}, \tilde{\mathbf{h}}_{t-1})) \tag{2.27}$$

$$p_{copy}(y_t) = \sum_{i:x_i=y_t} \alpha_i \tag{2.28}$$

where $\sigma_{copy} \in [0,1]$ is a scalar weight that controls the tradeoff between copying and generating, and f_{copy} is a function that transforms vectors $\mathbf{y}_{t-1}$, $\mathbf{g}_{t-1}$, and $\tilde{\mathbf{h}}_{t-1}$ into a single scalar value. Figure 2.4 illustrates the encoder-decoder architecture.

Training encoder-decoder models for opinion summarization require datasets of multiple reviews and summary pairs as supervision. These datasets are not freely available and must somehow be constructed synthetically. We will discuss synthetic dataset creation methods in the later chapters of the thesis.

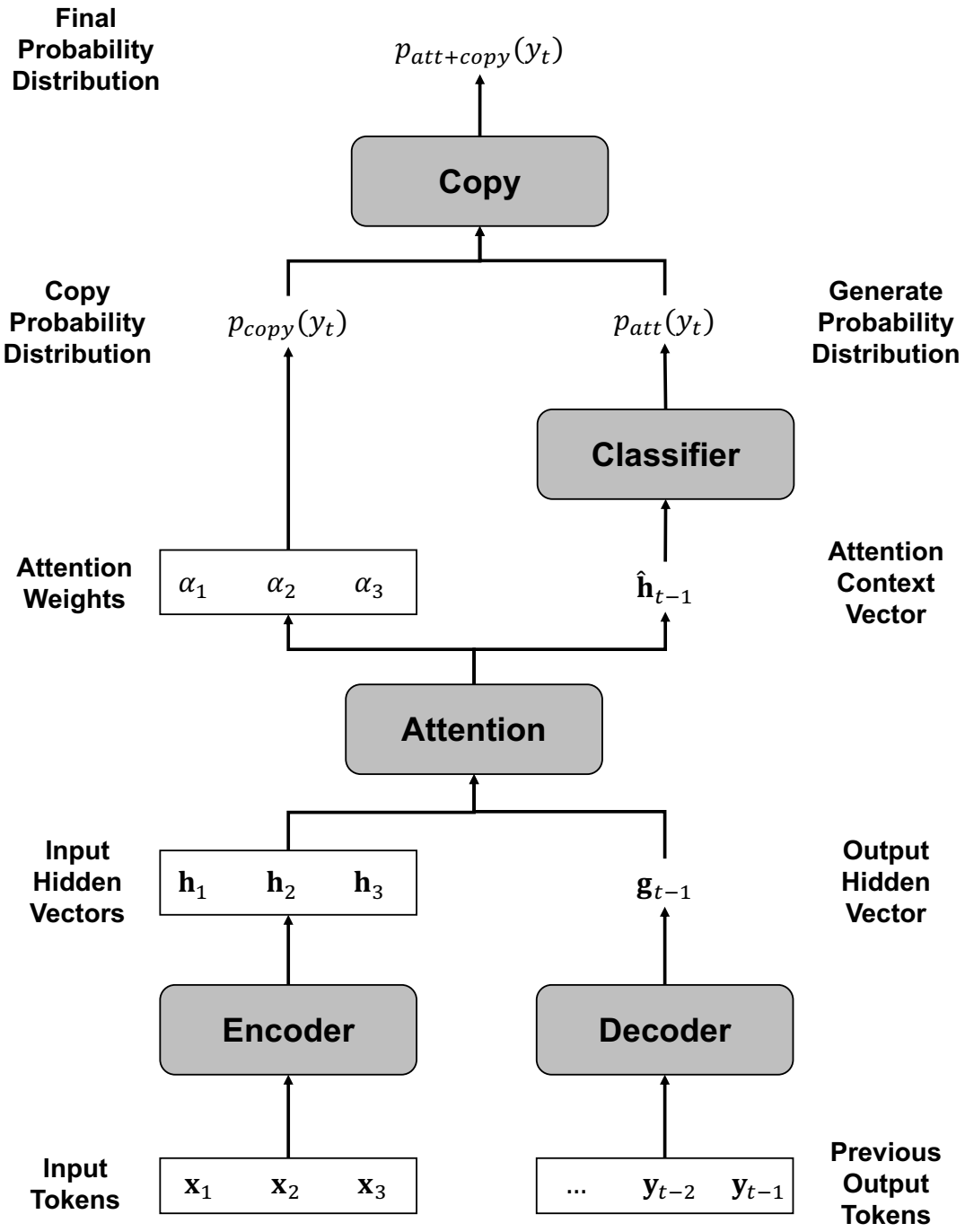


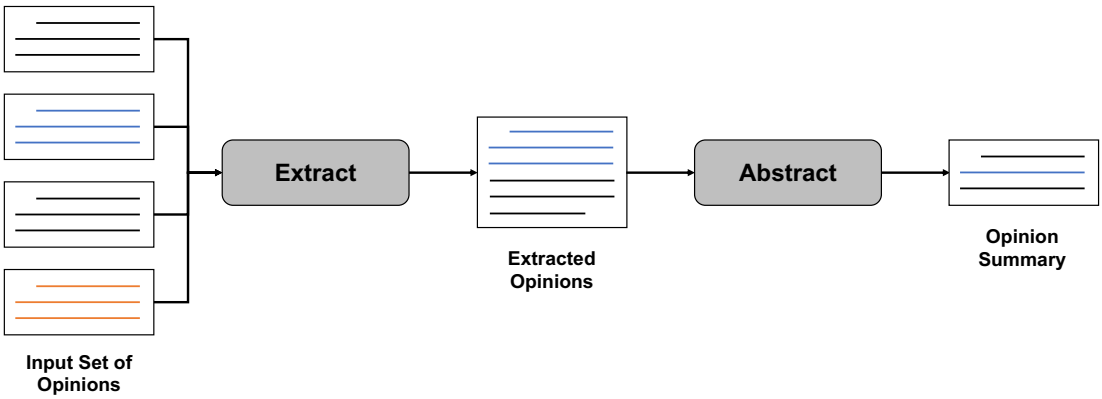
Figure 2.4: Illustration of the encoder-decoder architecture where the flow of information is from bottom to top. The encoder transforms the input tokens $\mathbf{x}_i$ into hidden vectors $\mathbf{h}_i$. For each decoding timestep t, the decoder calculates the final hidden vector $\mathbf{g}_{t-1}$ from previously generated tokens $\mathbf{y}_{<t}$. Input/Output hidden vectors $\mathbf{h}_i$ and $\mathbf{g}_{t-1}$ are then fed into attention and copy mechanisms to obtain the final probability distribution $p_{att+copy}(y_t)$ to predict the next token y_t.

Extract-Abstract Framework Opinion summarization deals with a set of multiple reviews as input, which can be prohibitively many and thus may not fit in memory. There are two strategies to mitigate this problem. Firstly, the Extract-Abstract (EA) framework (Wang and Ling, 2016) is widely used, which first pre-selects a small subset of input reviews that are considered salient, and then uses this subset as input to an encoder-decoder model (see Figure 2.5a).

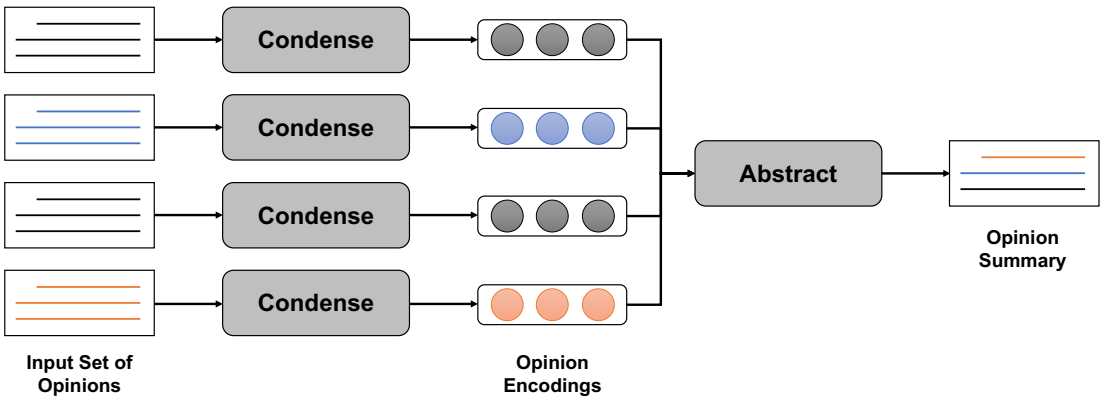
There are multiple ways to pre-select input that are explored in the literature. Wang and Ling (2016) select reviews using a ridge regression model trained with hand-engineered features representing review importance, such as POS tags and TF-IDF scores. Liu et al. (2018a) investigate four different extractive methods: (1) truncating reviews up to allowed input length, (2) ranking reviews using TF-IDF, and (3-4) selecting reviews using extractive summarizers such as TextRank (Mihalcea and Tarau, 2004) and SumBasic (Nenkova and Vanderwende, 2005). Their experiments showed that a TF-IDF ranker performed slightly better than the other alternatives. Amplayo and Lapata (2021) use neural-based extractive summarizers to select salient reviews, and show that an unsupervised centroid-based algorithm (Radev et al., 2000) that uses pre-trained weights (Devlin et al., 2019) performs better than supervised alternatives.

Condense-Abstract Framework Having an extractive stage that pre-selects reviews has two major drawbacks. Firstly, on account of having access to only a small subset of reviews, the summaries can be less informative and inaccurate. And secondly, user preferences cannot be easily taken into account when producing aspect-specific summaries, since more specialized information might have been removed. In Figure 2.5a, for example, the model would not be able to generate opinion summaries focused on the orange aspect, since the orange opinion is filtered out during the extractive stage.

To this end, Amplayo and Lapata (2021) propose an alternative framework called Condense-Abstract (CA), where instead of pre-selecting reviews, it enables the use of all input reviews by first condensing them into multiple dense vectors that serve as input to an abstractive model (see Figure 2.5b). Results in Amplayo and Lapata (2021) show that CA-based models are superior to EA-based ones and customization (e.g., generating summaries focusing on a specific aspect or sentiment) can be done easily in a zero-shot manner.



(a) Extract-Abstract (EA) Framework



(b) Condense-Abstract (CA) Framework

Figure 2.5: Illustration of EA and CA frameworks for opinion summarization. Among the input opinions, the colored ones are considered salient and each color represents an aspect. The EA framework pre-selects opinions from the input, which may cause information loss. The CA framework allows the use of all opinions by condensing them into opinion encodings.

2.2.3 Advantages and Disadvantages of Both Approaches

There are several advantages and disadvantages when using either extractive or abstractive approaches for opinion summarization. We compare both approaches in terms of three important dimensions of text summarization quality: factuality, grammaticality, and informativeness.

In terms of factuality, extractive summaries are better since extractive methods are limited to the selection of information found in the source. This information is always factual, except when there is a need for decontextualization (e.g., "It is good." needs to be decontextualized; Choi et al., 2021). Abstractive summaries, on the other hand,

have more freedom as to what it generates, e.g., it allows generation of novel tokens that are not found in the source. While this is more naturalistic since real-life summaries do sometimes contain novel tokens, it gives abstractive models the opportunity to *hallucinate*, i.e., generate information not supported by the source (Rohrbach et al., 2018; Maynez et al., 2020).

In terms of grammaticality, abstractive summaries are overall better than extractive ones for two reasons. Firstly, advances in neural-based summarization models (Rush et al., 2015; See et al., 2017; Liu and Lapata, 2019b) have substantially improved the fluency of generated summaries. Abstractive summaries are also more coherent since they are generated with sentences such that each sentence is conditioned on the previously generated ones. On the other hand, extractive summaries are produced using a set of extractions (usually in the form of sentences) from different parts of the source. While these extractions by themselves are fluent texts, joining them together may not be able to form a coherent body of text.

Finally, in terms of informativeness, both kinds of summaries can be more informative than the other depending on the user needs. Abstractive summaries contain more salient information in a concise and less redundant manner. While this can be appealing for users who just want a general overview (e.g., "The staff are friendly and helpful."), sometimes they might want more specific information (e.g., "The receptionist helped me carry my bags with a smile on his face."). Extractive summaries can better provide these specific information in detail, sacrificing conciseness and redundancy.

In this thesis, we choose to focus on abstractive opinion summarization models and aim to solve their issues on factuality and informativeness in the following chapters. In Chapter 4, we introduce a content planning module in our summarization model, which has been shown to improve the factuality of summaries (Narayan et al., 2021). In Chapter 5, we introduce a way to personalize the output of the summarization model based on user preferences. This improves the usability and informativeness of the summaries, since they can now be tailored towards different user needs.

2.2.4 Datasets

In this section, we describe five datasets that are most commonly used as benchmarks for opinion summarization. These datasets are gathered from review websites. They represent different domains and have unique characteristics in terms of summary length, the number of reviews to summarize, and the available reference sum-

Dataset	RT	Yelp	Amazon	Oposum	Space
review corpus size	245K	2.32M	1.18M	4.13M	1.14M
#domains	1	1	4	6	1
#aspects	—	—	—	18	6
#test instances	737	100	32	60	50
#reviews/instance	100	8	8	10	100
#references/instance	1	1	3	3	3
#general summaries	737	100	96	180	150
#aspect summaries	—	—	—	540	900

Table 2.1: Opinion summarization datasets and their descriptive statistics. RT refers to the Rotten Tomatoes dataset. Only Oposum and Space have aspect-specific summaries for evaluation. The underlined summaries are extractive.

maries. Table 2.1 presents an overview of the statistics of these datasets and their differences. Figures 2.6 and 2.7 show example reviews and opinion summaries found in the datasets. In subsequent chapters, we will use a subset of these datasets to evaluate our proposed models and compare their performance with baseline methods.

Rotten Tomatoes is a dataset that contains a large set of reviews for various movies written by professional critics (Wang and Ling, 2016). Each set of reviews has a gold-standard consensus summary written by an editor. The dataset is split into training, development, and test sets. In our experiments (see Chapters 3 and 4), we only use the gold-standard summaries during validation and testing, and thus we discard the summaries available during training. The length of reviews and summaries is shorter compared to other datasets, but there are, on average, a hundred reviews for each movie.

Yelp is a dataset of reviews and summaries from the business (mostly restaurants) domain (Chu and Liu, 2019). It consists of a large corpus of reviews for training, and a small subset of review-summary pairs for validation and testing. The latter were generated by annotators from Amazon Mechanical Turk (AMT). While the length of reviews and summaries is longer than those in Rotten Tomatoes, the number of reviews per entity at development and test time is fixed at eight (8) reviews.

Amazon (Bražinskas et al., 2019) contains product reviews for four Amazon categories: (1) Electronics, (2) Clothing, Shoes and Jewelry, (3) Home and Kitchen, and

	Example Review	Example Summary
RT	A suspense thriller with a sense of pleasurable unease, the film also serves up a juicy slice of human nature.	Alias Betty works both as a gripping thriller and as a precisely drawn character study.
Yelp	Great beer to try! Fun flavors like jalapeno pale ale. The staff inside is nice and friendly. I was able to get a t-shirt with no hassle at all. The outdoor seating area is wonderful. Birdsong is next door to noda, so you should definitely check it out!	If you're looking for a comfortable and inviting bar this is a great place to go. They have a lot of unique beers on tap that you will not find anywhere else. The staff here is extremely friendly, and after just a couple of minutes it feels like you are chatting with an old friend. The next time you want to head out for some drinks give them a shot!
Amazon	Produces about 25% coasters, and another 10-15% that won't play on dvd players. The first half was about what I expected, a few coasters, a couple minor errors. The second half I was lucky to get maybe 5 that worked flawlessly. No good!	These are silver discs, not the gold ones as advertised. The packaging is not good, they need to be packed more securely, the dvds shouldn't be able to slide around. The quality of the dvds is hit or miss. You might have a good batch and then run into several that won't work.

Figure 2.6: Example reviews and opinion summaries found in the Rotten Tomatoes (abbreviated as RT), Yelp, and Amazon datasets.

(4) Health and Personal Care. Unlike Rotten Tomatoes and Yelp, the Amazon dataset comes with development and test partitions that have three gold-standard reference summaries produced by Amazon Mechanical Turk (AMT) workers. The length of reviews and summaries is slightly shorter compared to Yelp, and the number of reviews per product is fixed at eight (8) reviews.

Oposum (Angelidis and Lapata, 2018b) is a large corpus of Amazon product reviews from six different domains: (1) laptop bags, (2) bluetooth headsets, (3) boots, (4) keyboards, (5) televisions, and (6) vacuum cleaners. It also includes an evaluation set with human-annotated *extractive* opinion summaries. Amplayo et al. (2021a) extended this

	Example Review	Example Summary
Oposum	Produces about 25% coasters, and another 10-15% that won't play on dvd players. The first half was about what I expected, a few coasters, a couple minor errors. The second half I was lucky to get maybe 5 that worked flawlessly. No good!	**General** The price is great. Lightweight and comfortable fit in the ear. Based on feedback from others I can be heard clearly. Very easy to use and compatible with all of my phones! It holds a charge great, is light enough. The sound quality is great, but cheap. **Aspect-Specific (sound quality)** The sound from the headphones is very good, the audio quality is excellent. Despite this they could have a louder sound for their maximum volume.
Space	Produces about 25% coasters, and another 10-15% that won't play on dvd players. The first half was about what I expected, a few coasters, a couple minor errors. The second half I was lucky to get maybe 5 that worked flawlessly. No good!	**General** Staff was service focused and very welcoming. Common areas of the hotel smelled fresh because of how clean everything was. The rooms were comfortable and came with a fridge and a microwave. Food, both hot and cold, was very well presented and fresh. The hotel was located within walking distance to the French quarter and felt very safe at night. **Aspect-Specific (location)** The location is very good, walking distance to all major sights in French quarter.

Figure 2.7: Example reviews, general and aspect-specific opinion summaries found in the Oposum and Space datasets.

dataset by adding three references for aspect-specific abstractive opinion summaries for three different aspects of a product (a total of nine summaries for each evaluation instance). Moreover, they increased the size of the review corpus to 4 million. See Chapter 5 for more details.

Space is a large corpus of 1 million hotel reviews from TripAdvisor (Angelidis et al., 2021). It also includes human-written abstractive opinion summaries for evaluation purposes, where each of the instances in the evaluation set consists of 100 input reviews and two sets of summaries: (a) three references for general opinion summaries, and (b) three references for aspect-specific summaries for six different aspects (a total of 18 summaries). The six aspects are building, cleanliness, food, location, rooms, and service.

2.2.5 Evaluation

The experiments conducted throughout this thesis are evaluated using automatic metrics and human-based judgment elicitation studies. We describe both types of evaluation in the next paragraphs.

Automatic Evaluation includes ROUGE (Lin, 2004) metrics, which is short for Recall-Oriented Understudy for Gisting Evaluation and is widely used for evaluating automatic summarization systems. ROUGE is defined as the ratio between the number of overlapping *terms x* over the total number of terms in the reference summary.

$$\text{ROUGE} = \frac{\sum_i \sum_j |\{\mathbf{ref}_{i,j} \mid \mathbf{ref}_{i,j} \in \mathbf{sys}\}|}{\sum_i |\mathbf{ref}_i|} \tag{2.29}$$

where **ref** is the list of reference summaries, **sys** is the system summary, and $|\cdot|$ is the set cardinality. While the original metric is recall-oriented, the F_1-score is more commonly used in the recent literature. The definition of *terms* depends on the ROUGE variant, in which the following are the three most popular ones:

- **ROUGE-N**: The terms used are n-grams. ROUGE-N where $N = 1$ and $N = 2$, i.e., unigram and bigram overlap ratio, are the most commonly used as proxy metrics for assessing informativeness.
- **ROUGE-L**: The terms used are tokens in a subsequence. The overlap is calculated as the least common subsequence between system and reference summaries. ROUGE-L is commonly used as a proxy to measuring the fluency of system summaries.
- **ROUGE-SU**: The terms used are *skip-bigrams* and unigrams. Skip-bigrams are bigrams in the text that are not necessarily consecutive. In ROUGE-SU4, we only consider skip-bigrams with a maximum skip distance of four tokens.

ROUGE-SU4 is the most common variant of ROUGE-SU and is used as an alternative to ROUGE-L.

Another widely used automatic evaluation metric is METEOR (Banerjee and Lavie, 2005), a recall oriented metric that also rewards matching stems, synonyms, paraphrases. The metric is based on the harmonic mean **f** of unigram precision and recall with recall weighted higher, that is:

$$\text{METEOR} = \frac{10 * \mathbf{prec} * \mathbf{rec}}{9 * \mathbf{prec} + \mathbf{rec}} (1 - \mathbf{penalty}) \tag{2.30}$$

$$\mathbf{prec} = \frac{m}{|\mathbf{sys}|} \tag{2.31}$$

$$\mathbf{rec} = \frac{m}{|\mathbf{ref}|} \tag{2.32}$$

$$\mathbf{penalty} = 0.5 \left(\frac{c}{m}\right)^3 \tag{2.33}$$

where m is the number of *good mappings*, which can be an exact match, or an exact match after stemming (e.g., *car* and *cars*), or a synonymic match (e.g., *car* and *automobile*). A **penalty** is also used to take into account chunk-level mappings, i.e., a match of two or more consecutive tokens, where c is the fewest possible chunks.

Human Evaluation While ROUGE-based metrics allow for easy comparison amongst different systems, they do not correlate very well with human assessments (Tay et al., 2019; Fabbri et al., 2021). To this end, we also conduct three kinds of evaluation methods that elicit human judgments.

Firstly, we assess the quality of system summaries based on several criteria using **Best-Worst Scaling** (BWS; Louviere et al., 2015). BWS is a less labor-intensive alternative to paired comparisons that has been shown to produce more reliable results than rating scales (Kiritchenko and Mohammad, 2017). Specifically, participants are shown a combination of the set of system summaries (e.g., one possible triple given five system summaries) and the gold-standard summary. Participants are then asked to select the *best* and *worst* among system summaries taking into account how much they deviate from the gold-standard summary given multiple possible criteria:

- **Informativeness**: How consistent are the opinions with the reference?
- **Coherence**: Is the summary easy to read and well-organized?
- **Conciseness**: Does the summary provide useful information in a concise manner?

- **Fluency**: Is the summary grammatical?

A BWS rating can be computed as the percentage of times the system in question is chosen as best minus the percentage of times it is selected as worst. The values range from -100 (unanimously worst) to 100 (unanimously best). We provide the detailed instructions shown to our annotators in Appendix A.1.

Secondly, we also examine the **veridicality of the system summaries**, namely whether the facts mentioned in them are indeed discussed in the input reviews. Specifically, participants are shown reviews and the corresponding system summary and are asked to verify for each sentence of the summary whether it is (a) fully supported by the reviews, (b) partially supported, or (c) not at all supported. For this study, we only compare abstractive summaries since extractive summaries are by default contain facts mentioned in the reviews. We then report the percentage of fully, partially, and un-supported sentences for each system summary. Systems with the highest percentage of supported sentences are considered more faithful to the input. We provide the instructions shown to annotators in Appendix A.2.

Finally, in this thesis, we also conduct studies to assess the **quality of aspect-specific summaries**. When assessing single-aspect summaries, we show participants the aspect in question (e.g., location of a hotel) as well as aspect summaries of competing systems. Participants are then asked to decide whether the summaries discuss the given aspect (a) exclusively, (b) partially, or (c) not at all. The system with the highest percentage of summaries that exclusively discuss the target aspect is considered best performing.

We also verify whether systems can produce summaries covering two aspects (i.e., multi-aspect summaries). Participants are shown two aspects in question (e.g., location and food of a hotel) and the corresponding multi-aspect summaries generated by competing systems. They are then asked to decide whether the summaries discuss (a) both target aspects exclusively, (b) one of the aspects, (c) other aspects in addition to the target ones, and (d) none of the two aspects. The system with the highest percentage in (a) is considered better at producing multi-aspect summaries. For more detailed instructions, we refer the interested reader to Appendix A.3.

2.3 Summary

In this chapter, we introduced two neural network models commonly used as building blocks in Natural Language Processing, namely recurrent neural networks with Long Short-Term Memory units and Transformers. We also defined the task of opinion summarization and its previously proposed variants, i.e., extractive and abstractive opinion summarization. Finally, we described extractive and abstractive methods for opinion summarization. In the next chapter, we will explore how to train neural opinion summarizers without review-summary pairs, by creating synthetic datasets.

Chapter 3

Synthetic Dataset Creation for Opinion Summarization

The majority of previous document summarization literature has focused on news articles, where training datasets consisting of document-summary pairs can be sourced relatively easily to facilitate supervised learning. Examples of these datasets include CNN/DailyMail (Hermann et al., 2015), Newsroom (Grusky et al., 2018), and Multi-News (Fabbri et al., 2019). The existence of large-scale training datasets is critical to the recent success of deep learning techniques (Hermann et al., 2015; Grusky et al., 2018; Liu et al., 2018a; Fabbri et al., 2019), especially for abstractive summarization. Unfortunately, in most review domains, such training data is not available and cannot be easily sourced. Manually writing opinion summaries is practically impossible since an annotator must read all available reviews for a given product or service which can be prohibitively many. Previous methods have relied on transfer learning techniques, where datasets from a source domain are used to train models for a target domain (Pan and Yang, 2010). However, different types of entities (e.g., products, movies, or businesses) impose different restrictions on the summaries which might vary in terms of length, or the types of aspect being mentioned, thus transfer learning can be problematic.

Motivated by these issues, Chu and Liu (2019) consider an *unsupervised* learning setting where there are only documents (product or business reviews) available without corresponding summaries. They propose an end-to-end neural model to perform abstractive summarization based on (a) an autoencoder that learns representations for each review and (b) a summarization module which takes the aggregate encoding of reviews as input and learns to generate a summary which is semantically similar to

the source documents (i.e., similar in terms of the cosine similarity between vectors of the summary and the documents). Due to the absence of ground truth summaries, the model is not trained to reconstruct the aggregate encoding of reviews, but rather it only learns to reconstruct the encoding of *individual* reviews. As a result it may not be able to generate meaningful text when the number of reviews is large. Furthermore, autoencoders are constrained to use simple decoders lacking attention (Bahdanau et al., 2014) and copy (Vinyals et al., 2015) mechanisms which have proven useful in the supervised setting leading to the generation of informative and detailed summaries. Problematically, a powerful decoder might be detrimental to the reconstruction objective, learning to express arbitrary distributions of the output sequence while ignoring the encoded input (Kingma and Welling, 2014; Bowman et al., 2016).

In this chapter, we enable the use of supervised techniques for opinion summarization. Specifically, we automatically generate a synthetic training dataset from a corpus of product reviews, and use this dataset to train a more powerful neural model with supervised learning. The synthetic data is created by selecting a review from the corpus, pretending it is a summary, generating multiple noisy versions thereof and treating these as *pseudo-reviews*. The latter are obtained with two noise generation functions targeting textual units of different granularity: *segment* noising introduces noise at the word- and phrase-level, while *document* noising replaces a review with a semantically similar one. We use the synthetic data to train a neural model that learns to denoise the pseudo-reviews and generate the summary. This is motivated by how humans write opinion summaries, where denoising can be seen as removing diverging information. Our proposed model consists of a multi-source encoder and a decoder equiped with an attention mechanism. Additionally, we introduce three modules: (a) *explicit denoising* guides how the model removes noise from the input encodings, (b) *partial copy* allows us to copy information from the source reviews only when necessary, and (c) a *discriminator* helps the decoder generate topically consistent text.

We perform experiments on two review datasets representing different domains (movies and businesses) and summarization requirements (short and longer summaries). Results based on automatic and human evaluation show that our method outperforms previous opinion summarization models, including the state-of-the-art abstractive system of Chu and Liu (2019) and is on the same par with a state-of-the-art supervised model (Wang and Ling, 2016) trained on a small sample of (genuine) review-summary pairs.

3.1 Related Work

As mentioned in Chapter 2, most previous work on unsupervised opinion summarization has focused on extractive approaches (Carenini et al., 2006; Ku et al., 2006; Paul et al., 2010; Angelidis and Lapata, 2018b) where a clustering model groups opinions of the same aspect, and a sentence extraction model identifies text representative of each cluster. Ganesan et al. (2010) propose a graph-based abstractive framework for generating concise opinion summaries, while Di Fabbrizio et al. (2014) use an extractive system to first select salient sentences and then generate an abstractive summary based on hand-written templates (Carenini and Moore, 2006).

In this chapter, we follow the setting of Chu and Liu (2019) in assuming that we have access to reviews but no gold-standard summaries. Their model is an autoencoder which learns to reconstruct individual reviews. At test time, the model generates an opinion summary by reconstructing a canonical review of the average encoding of input reviews. The canoncial review is then treated as the output summary. Our proposed method is also abstractive and neural-based, but eschews the use of an autoencoder in favor of supervised sequence-to-sequence learning through the creation of a synthetic training dataset. Bražinskas et al. (2019) also use synthetic datasets to train a sequence-to-sequence model with attention and copy mechanisms (See et al., 2017), along with a hierarchical variational autoencoder to learn the latent code of the summary. They use random sampling to synthesize the data for supervised training, while our dataset construction method is more principled and makes use of linguistically motivated noise functions.

Our work relates to denoising autoencoders (DAEs; Vincent et al., 2008), which have been effectively used as unsupervised methods for various NLP tasks. Earlier approaches have shown that DAEs can be used to learn high-level text representations for domain adaptation (Glorot et al., 2011b) and multimodal representations of textual and visual input (Silberer and Lapata, 2014). Recent work has applied DAEs to text generation tasks, specifically to data-to-text generation (Freitag and Roy, 2018) and extractive sentence compression (Fevry and Phang, 2018). Our model differs from these approaches in two respects. Firstly, while previous work has adopted trivial noising methods such as randomly adding or removing words (Fevry and Phang, 2018) and randomly corrupting encodings (Silberer and Lapata, 2014), our noise generators are more linguistically informed and suitable for the opinion summarization task. Secondly, while in Freitag and Roy (2018) the decoder is limited to vanilla RNNs, our

noising method enables the use of more complex architectures, enhanced with attention and copy mechanisms, which are known to improve the performance of summarization systems (Rush et al., 2015; See et al., 2017).

3.2 Modeling Approach

Let $R_e = \{r_1, ..., r_{|R_e|}\}$ denote a set of reviews about entity e (e.g., a movie or business). Our aim is to generate a summary y_e of the opinions expressed in R_e. We further assume access to a corpus $C = \{R_1, ..., R_{|E|}\}$ containing multiple sets of reviews about $|E|$ products without corresponding opinion summaries.

Our method consists of two parts. We first create a synthetic dataset $\hat{D} = \{(\hat{R}, \hat{y})\}$ consisting of summary-review pairs. Specifically, we sample a single review from corpus C, pretend it is a summary, and generate multiple noisy versions thereof (i.e., pseudo-reviews). At training time, a denoising model learns to remove the noise from the reviews and generate the summary. At test time, the same denoising model is used to summarize *actual* reviews. We use denoising as an auxiliary task for opinion summarization to simulate the fact that summaries tend to omit opinions that do not represent consensus (i.e., noise in the pseudo-review), but include salient opinions found in most reviews (i.e., non-noisy parts of the pseudo-review).

3.2.1 Synthetic Dataset Creation via Noising

We sample a review as a candidate summary and generate noisy versions thereof, using two functions: (a) segment noising adds noise at the token and chunk level, and (b) document noising adds noise at the text level. The noise functions are illustrated in Figure 3.1.

Summary Sampling Summaries and reviews follow different writing conventions. For example, reviews are subjective, and often include first-person singular pronouns such as *I* and *my* and several unnecessary characters or symbols. They may also vary in length and detail. We discard reviews from corpus C which display an excess of these characteristics based on a list of domain-specific constraints (detailed in Section 3.3). We sample a review $\hat{y}$ from the filtered corpus, which we use as the candidate summary.

Segment Noising Given candidate summary $\hat{y} = \{w_1, ..., w_L\}$, we create a set of segment-level noisy versions $\hat{R}^{(c)} = \{r_1^{(c)}, ..., r_N^{(c)}\}$. Previous work has adopted nois-

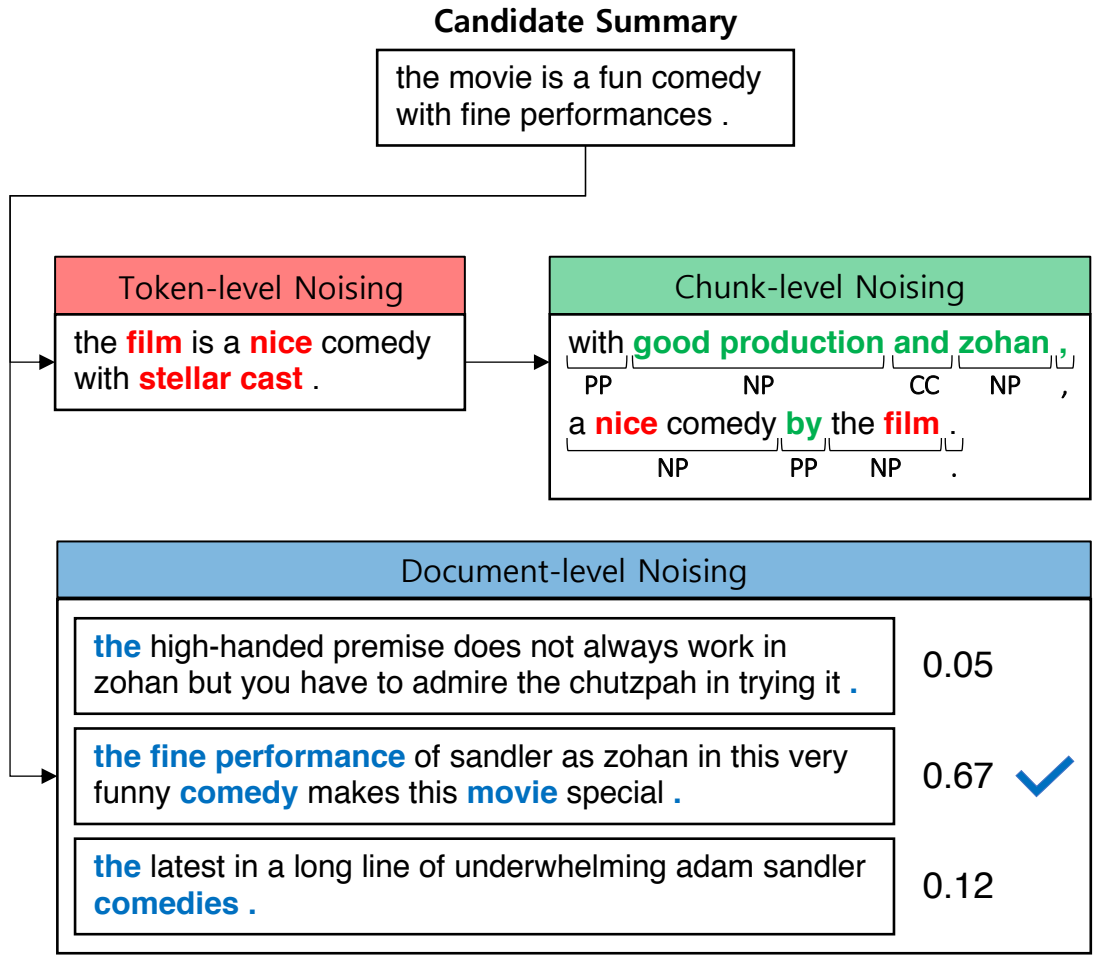


Figure 3.1: Synthetic dataset creation. Given a sampled candidate summary, we add noise using two methods: (a) segment noising performs token- and chunk-level alterations, and (b) document noising replaces the text with a semantically similar review.

ing techniques based on random n-gram alterations (Fevry and Phang, 2018), however, we instead rely on two simple, linguistically informed noise functions. Firstly, we train a bidirectional language model (BiLM; Peters et al., 2018) on the review corpus C. For each token in $\hat{y}$, the BiLM predicts a softmax token distribution which can be used to replace tokens. Secondly, we utilize FLAIR[1] (Akbik et al., 2019), an off-the-shelf state-of-the-art syntactic chunker that leverages contextual embeddings, to shallowly parse each review r in corpus C. This results in a list of chunks $\mathcal{C}_r = \{c_1, ..., c_K\}$ with corresponding syntactic labels $\mathcal{G}_r = \{g_1, ..., g_K\}$ for each review r, which we use for replacing and rearranging chunks.

Segment-level noising involves token- and chunk-level alterations. Token-level

[1] https://github.com/zalandoresearch/flair

alterations are performed by replacing tokens in $\hat{y}$ with probability $p^{\mathcal{R}}$. Specifically, we replace token w_j in $\hat{y}$, by sampling token w'_j from the BiLM predicted word distribution (see in Figure 3.1). We use nucleus sampling (Holtzman et al., 2019), which samples from a rescaled distribution of tokens with probability higher than a threshold $p^{\mathcal{N}}$, instead of the original distribution. This has been shown to yield better samples in comparison to top-k sampling, mitigating the problem of text degeneration (Holtzman et al., 2019).

Chunk-level alterations are performed by removing and inserting chunks in $\hat{y}$, and rearranging them based on a sampled syntactic template. Specifically, we first shallowly parse y using FLAIR, obtaining a list of chunks $\mathcal{C}_y$, each of which is removed with probability $p^{\mathcal{R}}$. We then randomly sample a review r from our corpus and use its sequence of chunk labels $\mathcal{G}_r$ as a syntactic template, which we fill in with chunks in $\mathcal{C}_y$ (sampled without replacement), if available, or with chunks in corpus $\mathcal{C}$, otherwise. This results in a noisy version $r^{(c)}$ (see Figure 3.1 for an example). Repeating the process N times produces the segment-level noisy set of reviews $\hat{R}^{(c)}$.

Document Noising Given candidate summary $\hat{y}$, we also create another set of document-level noisy versions $\hat{R}^{(d)} = \{r_1^{(d)}, ..., r_N^{(d)}\}$. Instead of manipulating parts of the summary, we altogether replace it with a similar review from the corpus and treat it as a noisy version. Specifically, we select N reviews that are most similar to $\hat{y}$ and discuss the same product. To measure similarity, we use IDF-weighted ROUGE-1 F1 (Lin, 2004), where we calculate the lexical overlap between the review and the candidate summary, weighted by token importance:

$$overlap = \sum_{w_j \in r} \big(\text{IDF}(w_j) * 1(w_j \in \hat{y})\big) \tag{3.1}$$

$$\text{P} = overlap/|x| \qquad \text{R} = overlap/|y| \tag{3.2}$$

$$\text{F}_1 = (2 * \text{P} * \text{R})/(\text{P} + \text{R}) \tag{3.3}$$

where r is a review in the corpus, $1(\cdot)$ is an indicator function, and P, R, and F_1 are the ROUGE-1 precision, recall, and F_1 score, respectively. The reviews with the highest F_1 scores are selected as noisy versions of $\hat{y}$, resulting in the noisy set $\hat{R}^{(d)}$ (see Figure 3.1).

We create a total of $2 * N$ noisy versions of $\hat{y}$, i.e., $\hat{R} = \hat{R}^{(c)} \cup \hat{R}^{(d)}$ and obtain our synthetic training data $\hat{D} = \{(\hat{R}, \hat{y})\}$ by generating $|\hat{D}|$ pseudo-review-summary pairs. Both noising methods are necessary to achieve aspect diversity amongst input reviews. Segment noising creates reviews which may mention aspects not found in the summary, while document noising creates reviews with content similar to the summary.

Candidate Summary:	
Quite possibly the greatest romantic comedy since some like it hot.	
Segment-Level Noisy Versions	
S.1.	Some can't laugh hard of the best romantic comedy.
S.2.	The best romantic comedy set funny and unexpectedly moving, organically revealed but the sets something since the sexes.
S.3.	Movies created since its main pleasures' mystique recklessly assembled and all hers... love showcases of his stars.
Document-Level Noisy Versions	
D.1.	Meg Ryan-Billy Crystal romantic comedy is hard not to like.
D.2.	`[MOVIE]` ... is an adult romantic comedy in a time when we don't get very many, and it has one thing going for it that gives it an enormous boost – it's very funny.
D.3.	... A better-than-average romantic comedy that remains just as relevant now as it did in 1989

Figure 3.2: Noisy versions generated using segment and document noising on the Rotten Tomatoes dataset, where `[MOVIE]` is the title of the movie.

Relying on either noise function alone decreases performance (see the ablation studies in Section 3.4).

Figure 3.2 shows example noisy versions of a candidate summary using both segment and document noising methods. Although segment noising yields texts which may not be entirely comprehensible to humans, a few segments contain understandable content that could be perceived as diverging information and as such should not be included in the summary (e.g., "some can't laugh hard" in S.1). Similarly, noisy versions generated by document noising include content that is somewhat related but not critical for generating the summary (e.g., "remains just as relevant now as it did in 1989" in D.3). These examples show that our noise functions are not entirely random, in contrast to previous trivial and un-informed approaches (Fevry and Phang, 2018; Silberer and Lapata, 2014).

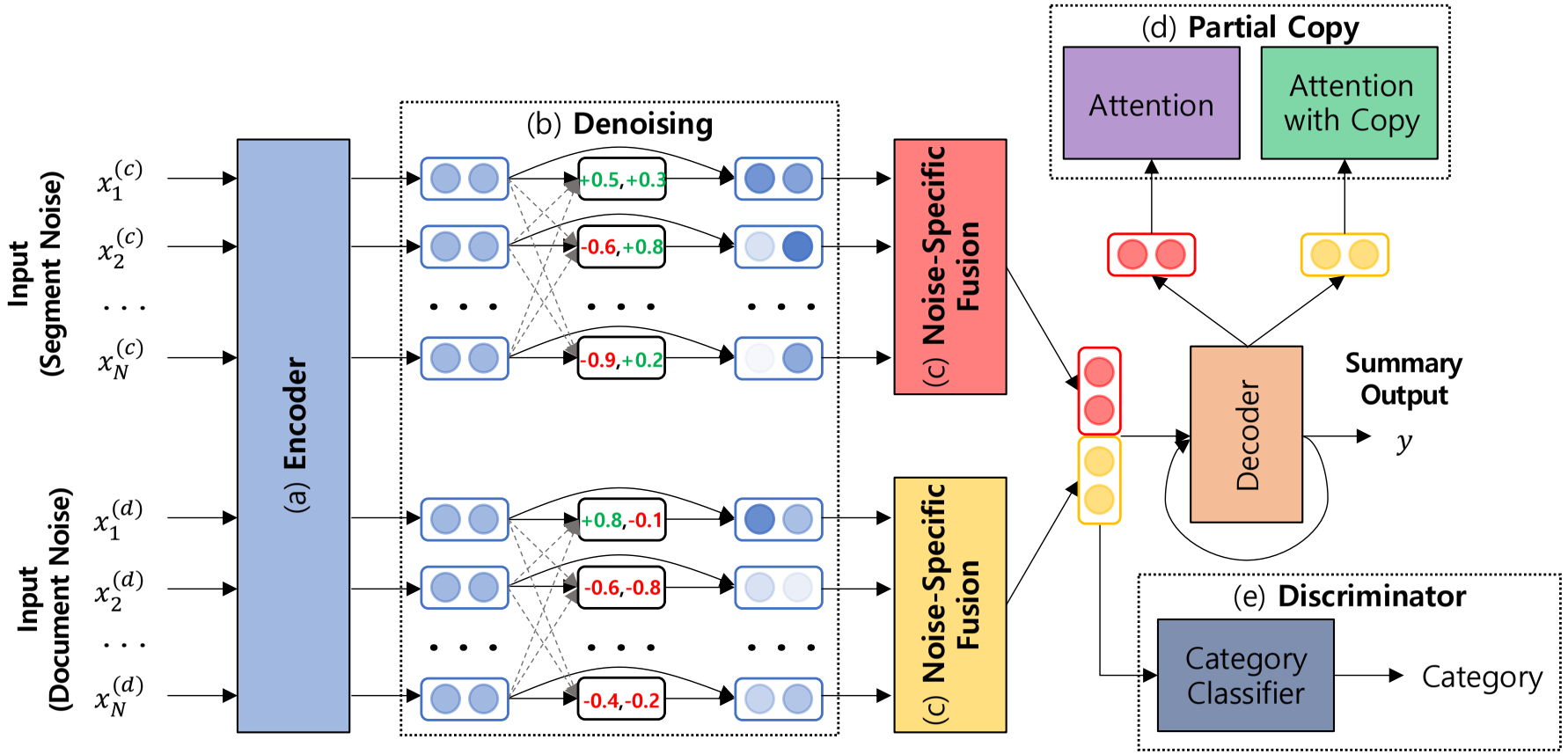


Figure 3.3: Architecture of DENOISESUM: it consists of a multi-source encoder with explicit denoising, noise-specific fusion, a decoder with partial copy, and a review category classifier. The green and red numbers indicate the amount of positive and negative adjustments needed to denoise the encodings.

3.2.2 Summarization via Denoising

We summarize (aka denoise) the input set of reviews R with our model which we call DENOISESUM, illustrated in Figure 3.3. A multi-source encoder produces an encoding for each pseudo-review. The encodings are further corrected via an explicit denoising module, and then fused into an aggregate encoding for each type of noise. Finally, the fused encodings are passed to a decoder with a partial copy mechanism to generate the summary y.

Multi-Source Encoder For each review $r_j \in R$ where $r_j = \{\mathbf{x}_1, ..., \mathbf{x}_L\}$ and $\mathbf{x}_k$ is the vector representation of the kth token in r_j, we obtain contextualized token encodings $\{\mathbf{h}_k\}$ and an overall review encoding $\mathbf{d}_j$ with a BiLSTM encoder (Hochreiter and Schmidhuber, 1997):

$$\overrightarrow{\mathbf{h}}_k = \text{LSTM}_f(\mathbf{x}_k, \overrightarrow{\mathbf{h}}_{k-1}) \tag{3.4}$$

$$\overleftarrow{\mathbf{h}}_k = \text{LSTM}_b(\mathbf{x}_k, \overleftarrow{\mathbf{h}}_{k+1}) \tag{3.5}$$

$$\mathbf{h}_k = [\overrightarrow{\mathbf{h}}_k; \overleftarrow{\mathbf{h}}_k] \tag{3.6}$$

$$\mathbf{d}_j = [\overrightarrow{\mathbf{h}}_L; \overleftarrow{\mathbf{h}}_1] \tag{3.7}$$

where $\overrightarrow{\mathbf{h}}_k$ and $\overleftarrow{\mathbf{h}}_k$ are forward and backward hidden states of the BiLSTM at timestep k, and ; denotes concatenation (see module (a) in Figure 3.3).

Explicit Denoising The model should be able to remove noise from the encodings before decoding the text. While previous methods (Vincent et al., 2008; Freitag and Roy, 2018) *implicitly* assign the denoising task to the encoder, we propose an *explicit* denoising component (see module (b) in Figure 3.3). Specifically, we create a correction vector $\mathbf{c}_j$ for each review $\mathbf{d}_j$. $\mathbf{c}_j$ represents the adjustment needed to denoise each dimension of $\mathbf{d}_j$ and is used to create $\mathbf{d}'_j$, a denoised encoding of $\mathbf{d}_j$:

$$\mathbf{q} = \sum_{j=1}^{N} \mathbf{d}_j / N \tag{3.8}$$

$$\mathbf{c}_j = \tanh(\mathbf{W}_d[\mathbf{d}_j; \mathbf{q}] + \mathbf{b}_d) \tag{3.9}$$

$$\mathbf{d}'_j = \mathbf{d}_j + \mathbf{c}_j \tag{3.10}$$

where $\mathbf{q}$ represents a mean review encoding and functions as a query vector, $\mathbf{W}$ and $\mathbf{b}$ are learned parameters. We can interpret the correction vector as removing or adding information to each dimension when its value is negative or positive, respectively. We

use two explicit denoising modules, one for segment noisy inputs and another for document noisy inputs.

Noise-Specific Fusion For each type of noise (segment and document), we create a noise-specific aggregate encoding by fusing the denoised encodings into one (see module (c) in Figure 3.3). Given $\{\mathbf{d}'_j\}$, the set of denoised encodings, we create aggregate encoding $\mathbf{s}$:

$$\alpha_j = \text{softmax}(\mathbf{W}_f \mathbf{d}'_j + \mathbf{b}_f) \tag{3.11}$$

$$\mathbf{s} = \sum_j \mathbf{d}'_j * \alpha_j \tag{3.12}$$

where α_j is a weight vector with the same dimensionality as the denoised encodings, which signifies the importance of denoised embeddings $\{\mathbf{d}'_j\}$.

Decoder with Partial Copy Our decoder generates a summary $y = \{w_t\}$ given aggregated encodings for segment and document noise, i.e., $\mathbf{s}^{(c)}$ and $\mathbf{s}^{(d)}$, as input. An advantage of our method is its ability to incorporate techniques used in supervised models, such as attention (Bahdanau et al., 2014) and copy (Vinyals et al., 2015). Pseudo-reviews created using segment noising include various chunk permutations, which could result to ungrammatical and incoherent text. Using a copy mechanism on these texts may hurt the fluency of the output. We therefore allow copy on document noisy inputs only (see module (d) in Figure 3.3).

To implement partial copy, we use two LSTM decoders for the aggregate encodings, one equipped with attention and copy mechanisms, and one without copy mechanism. We then combine the results of these decoders using a learned gate. Specifically, token w_t at timestep t is predicted as:

$$\mathbf{s}_t^{(c)}, p^{(c)}(w_t) = \text{LSTM}_{\text{att}}(\mathbf{x}_{t-1}, \mathbf{s}_{t-1}^{(c)}) \tag{3.13}$$

$$\mathbf{s}_t^{(d)}, p^{(d)}(w_t) = \text{LSTM}_{\text{att+copy}}(\mathbf{x}_{t-1}, \mathbf{s}_{t-1}^{(d)}) \tag{3.14}$$

$$\lambda_t = \sigma(\mathbf{W}_p[\mathbf{x}_{t-1}; \mathbf{s}_t^{(c)}; \mathbf{s}_t^{(d)}] + \mathbf{b}_p) \tag{3.15}$$

$$p(w_t) = \lambda_t * p^{(c)}(w_t) + (1 - \lambda_t) * p^{(d)}(w_t) \tag{3.16}$$

where $\mathbf{s}_t$ and $p(w_t)$ are the hidden state and predicted token distribution at timestep t, $\mathbf{x}_t$ is the embedding representation of w_t, and $\sigma(\cdot)$ is the sigmoid function.

3.2.3 Training and Inference

We use a maximum likelihood loss to optimize the generation probability distribution based on summary $y = \{w_t\}$ from our synthetic dataset:

$$\mathcal{L}_{gen} = -\sum_{w_t \in y} \log p(w_t) \tag{3.17}$$

The decoder depends on $\mathcal{L}_{gen}$ to generate meaningful, denoised outputs. As this is a rather *indirect* way to optimize our denoising module, we additionally use a discriminative loss providing *direct* supervision. The discriminator operates at the output of the fusion module and predicts the category distribution $p(z)$ of the output summary y (see module (e) in Figure 3.3). The type of categories varies across domains. For movies, categories can be information about their genre (e.g., drama, comedy), while for businesses their specific type (e.g., restaurant, beauty parlor). This information is often included in reviews but we assume otherwise and use an LDA topic model (Blei et al., 2003) to infer $p(z)$ (we present experiments with human labeled and automatically induced categories in Section 3.4). An MLP classifier takes as input aggregate encodings $\mathbf{s}^{(c)}$ and $\mathbf{s}^{(d)}$ and infers $q(z)$. The discriminator is trained by calculating the KL divergence between predicted and actual category distributions $q(z)$ and $p(z)$:

$$q(z) = \text{MLP}_d(\mathbf{s}^{(c)}, \mathbf{s}^{(d)}) \tag{3.18}$$

$$\mathcal{L}_{disc} = D_{KL}(p(z) \parallel q(z)) \tag{3.19}$$

The final objective is the sum of both loss functions:

$$\mathcal{L} = \mathcal{L}_{gen} + \mathcal{L}_{disc} \tag{3.20}$$

At test time, we are given genuine reviews R as input instead of synthetic ones. We generate a summary by treating R as $\hat{R}^{(c)}$ and $\hat{R}^{(d)}$, i.e., the outcome of segment and document noising.

3.3 Experimental Setup

3.3.1 Dataset

We performed experiments on two datasets which represent different domains and summary types. The Rotten Tomatoes dataset[2] (Wang and Ling, 2016) contains a large

[2] `http://www.ccs.neu.edu/home/luwang/data.html`

Rotten Tomatoes	Train*	Dev	Test
#movies	25k	536	737
#reviews/movie	40.0	98.0	100.3
#tokens/review	28.4	23.5	23.6
#tokens/summary	22.7	23.6	23.8
corpus size			245,848

Yelp	Train*	Dev	Test
#businesses	100k	100	100
#reviews/business	8.0	8.0	8.0
#tokens/review	72.3	70.3	67.8
#tokens/summary	64.8	70.9	67.3
corpus size			2,320,800

Table 3.1: Dataset statistics; Train* column refers to the synthetic data we created through noising (detailed in Section 3.2.1).

set of reviews for various movies written by critics. Each set of reviews has a gold-standard consensus summary written by an editor. We follow the partition of Wang and Ling (2016) but do not use ground truth summaries during training to simulate our unsupervised setting. The Yelp dataset[3] from Chu and Liu (2019) includes a large training corpus of reviews without gold-standard summaries. The latter are provided for development and testing and were generated by an Amazon Mechanical Turker. We follow the splits introduced in their work. A comparison between the two datasets is provided in Table 3.1. As can be seen, Rotten Tomatoes summaries are generally short while Yelp reviews are approximately three times longer. Interestingly, there are a lot more reviews to summarize in Rotten Tomatoes (approximately 100 reviews) while input reviews in Yelp are considerably less (i.e., 8 reviews). We refer the readers to Chapter 2 for details and example reviews and summaries of both datasets.

3.3.2 Implementation

To create the synthetic dataset, we sample candidate summaries using the following constraints: (1) the number of non-alphanumeric symbols must be less than 3, (2) there

[3] https://github.com/sosuperic/MeanSum

must be no first-person singular pronouns (not used for Yelp), and (3) the number of tokens must be between 20 to 30 (50 to 90 for Yelp). We decided on these constraints based on summary statistics in the development set. We set $p^{\mathcal{R}}$ to 0.8 and 0.4 for token and chunk noise, and $p^{\mathcal{N}}$ to 0.9. For each review-summary pair, the number of reviews N is sampled from the Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$ where μ and σ are the mean and standard deviation of the number of reviews in the development set. We created 25k (Rotten Tomatoes) and 100k (Yelp) pseudo-reviews for our synthetic datasets (see Table 3.1).

We set the dimensions of the word embeddings to 300, the vocabulary size to 50K, the hidden dimensions to 256, the batch size to 8, and dropout (Srivastava et al., 2014) to 0.1. For our discriminator, we employed an LDA topic model (Blei et al., 2003) trained on the review corpus, with 50 (Rotten Tomatoes) and 100 (Yelp) topics (tuned on the development set). See Appendix B for examples of these topics. The LSTM weights were pretrained with a language modeling objective, using the corpus as training data (245K reviews for Rotten Tomatoes and 2.3M reviews for Yelp; see Table 3.1). For Yelp, we additionally trained a coverage mechanism (See et al., 2017) in a separate training phase to avoid repetition. That is, at each timestep we maintain a coverage vector, which is calculated as the sum of attention distributions over all previous timesteps, and is used as extra input to the attention mechanism (in Equations 3.13 and 3.14). We used the Adam optimizer (Kingma and Ba, 2015) with a learning rate of 0.001 and l_2 constraint of 3. At test time, summaries were generated using length normalized beam search with a beam size of 5. We performed early stopping based on the performance of the model on the development set. Our model was trained on a single GeForce GTX 1080 Ti GPU and is implemented using PyTorch.

3.3.3 Comparison Systems

We compared DENOISESUM to several unsupervised extractive and abstractive methods. Extractive approaches include

- LEXRANK (Erkan and Radev, 2004), where a graph with sentences as nodes and sentence similarity as edges is created. Then, PageRank is used to calculate the centrality scores of sentences. The highest scoring sentences are selected as part of the summary.

- WORD2VEC (Rossiello et al., 2017), a centroid-based method which represents reviews as token embeddings aggregated using weights based on the inverse doc-

ument frequency (IDF) of the tokens. The review closest to the *centroid* (Radev et al., 2000), i.e. the average review representation, is then used as the summary.

- SENTINEURON, which is similar to WORD2VEC but uses Sentiment Neuron (Radford et al., 2017), an LSTM-based language model trained using a large-scale review corpus, as the input representation to calculate the centroid.

Abstractive methods include

- OPINOSIS (Ganesan et al., 2010), a method that first transforms opinions into a graph. The graph consists of words as nodes, that are weighted by their occurence in the input and connected by edges if they are neighboring words. The highly occuring and connected words are then treated as the abstractive summary.

- MEANSUM (Chu and Liu, 2019), a neural model based on autoencoders that is trained to reconstruct individual reviews. To generate the summaries, it first aggregate encodings of reviews through averaging, and used the aggregated encoding to reconstruct a textual output. This is then treated as the summary.

As an upper bound, we use an ORACLE strategy, which selects as summary the review which maximizes the ROUGE-1/2/L F1 score against the gold summary. Finally, for Rotten Tomatoes, we also compared with the state-of-the-art supervised model CONDASUM proposed in Amplayo and Lapata (2021), which use the Condense-Abstract framework (see Chapter 2) to enable the use of all reviews as input and is trained using the original training split (i.e., with gold-standard summaries).

3.4 Results

3.4.1 Automatic Evaluation

Our results on Rotten Tomatoes are shown in Table 3.2. Following previous work (Wang and Ling, 2016; Amplayo and Lapata, 2021) we report five metrics: METEOR Denkowski and Lavie (2014), a recall-oriented metric that rewards matching stems, synonyms, and paraphrases; ROUGE-SU4 (Lin, 2004), the recall of unigrams and skip-bigrams of up to four words; and the F1-score of ROUGE-1/2/L, which respectively measures word-overlap, bigram-overlap, and the longest common subsequence

Model	METEOR	ROUGE-SU4	ROUGE-1	ROUGE-2	ROUGE-L
LEXRANK	5.59	3.98	—	—	—
WORD2VEC	6.14	4.04	13.93	2.10	10.81
SENTINEURON	7.02	4.77	15.90	2.01	11.74
OPINOSIS	6.07	4.90	—	—	—
MEANSUM	6.07	4.41	15.79	1.94	12.26
DENOISESUM	**8.30***	**6.84***	**21.26***	**4.61***	**16.27***
CONDASUM	8.90	7.79	22.49	7.65	18.47
ORACLE	12.10	12.01	30.94	10.75	24.95

Table 3.2: Automatic evaluation on Rotten Tomatoes. Extractive/abstractive models shown in the first/second block. Best performing results for unsupervised models are **boldfaced**. An asterisk (*) means there is a significant difference between DENOISESUM and MEANSUM (based on paired bootstrap resampling; $p < 0.05$).

between system and reference summaries. Results on Yelp are given in Table 3.3 where we compare systems using ROUGE-1/2/L F1, following Chu and Liu (2019).

As can be seen, DENOISESUM outperforms all competing models on both datasets. When compared to MEANSUM, the difference in performance is especially large on Rotten Tomatoes, where we see a 4.01 improvement in ROUGE-L. We believe this is because MEANSUM does not learn to reconstruct encodings of aggregated inputs, and as a result it is unable to produce meaningful summaries when the number of input reviews is large, as is the case for Rotten Tomatoes. In fact, the best extractive model, SENTINEURON, slightly outperforms MEANSUM on this dataset across metrics with the exception of ROUGE-L. When compared to the best supervised system, DENOISESUM performs comparably on several metrics, specifically METEOR and ROUGE-1, however, there is still a gap on ROUGE-2, showing the limitations of systems trained without gold-standard summaries.

We show example summaries produced by four systems: SENTINEURON, MEANSUM, our model DENOISESUM, as well as the GOLD-standard summary in Figure 3.4 (for Rotten Tomatoes) and Figure 3.5 (for Yelp). The extractive model SENTINEURON tends to select reviews that are longer and more verbose. Summaries generated by MEANSUM on Rotten Tomatoes are mostly gibberish, which we argue is due to the model being unable to handle the large number of input reviews in this dataset. Overall, DENOISESUM produces the best summaries among the three systems.

Model	ROUGE-1	ROUGE-2	ROUGE-L
ORACLE	31.07	6.11	18.11
LEXRANK	24.62	3.66	14.51
WORD2VEC	24.61	2.85	13.81
SENTINEURON	25.05	3.09	14.56
OPINOSIS	20.85	1.52	11.46
MEANSUM	28.86	3.66	15.91
DENOISESUM	**30.14***	**4.99***	**17.65***

Table 3.3: Automatic evaluation on Yelp. Extractive/abstractive models shown in the first/second block. Best performing unsupervised models are **boldfaced**. An asterisk (*) means there is a significant difference between DENOISESUM and MEANSUM (based on paired bootstrap resampling; $p < 0.05$).

Table 3.4 presents various ablation studies on Rotten Tomatoes (RT) and Yelp which assess the contribution of different model components. Specifically, we evaluate multiple versions of DENOISESUM using ROUGE-L as the metric, divided into three categories. Firstly, we compare versions of our model trained with less synthetic datasets. Our results show that increasing the size of synthetic dataset improves performance, where our final model with 100% data performs the best. Secondly, we compare with versions that only use a single noising method. Interestingly, the performance of the noising methods depends on the dataset, where segment noising is more useful in Rotten Tomatoes while document noising is more useful in Yelp. The final model which uses both noising methods outperforms the other models. Finally, we compare DENOISESUM with versions where a component is removed. As can be seen in the table, all these versions perform worse than the final model, showing that all components help improve the performance of DENOISESUM. We also tried using human-labeled categories, genres (e.g., horror, romance, etc.) in Rotten Tomatoes and categories (e.g., Asian, burgers, etc.) in Yelp, instead of LDA topics to train the discriminator. This interestingly results to a decrease in model performance, which suggests that more useful labels for discriminator can be approximated better by automatic means.

Model	Rotten Tomatoes	Yelp
DENOISESUM	16.27	17.65
10% synthetic dataset	15.39	16.22
50% synthetic dataset	15.76	17.54
no segment noising	16.03	16.88
no document noising	16.22	16.67
no explicit denoising	16.06	17.06
no partial copy	15.89	16.31
no discriminator	15.84	16.64
using human categories	15.87	15.86

Table 3.4: ROUGE-L of our model and versions thereof with less synthetic data (second block), using only one noising method (third block), and without some modules (fourth block).

3.4.2 Human Evaluation

We also conducted two judgment elicitation studies using the Amazon Mechanical Turk (AMT) crowdsourcing platform. The first study assessed the quality of the summaries using Best-Worst Scaling (BWS; Louviere et al., 2015; see Chapter 2), a less labor-intensive alternative to paired comparisons that has been shown to produce more reliable results than rating scales (Kiritchenko and Mohammad, 2017). Specifically, participants were shown the movie/business name, some basic background information, and a gold-standard summary. They were also presented with three system summaries, produced by SENTINEURON (best extractive model), MEANSUM (most related unsupervised model), and DENOISESUM. We refer the readers to Appendix A.1 for the full instructions of the experiment.

Participants were asked to select the *best* and *worst* among system summaries taking into account how much they deviated from the ground truth summary in terms of: *Informativeness* (i.e., does the summary present opinions about specific aspects of the movie/business in a concise manner?), *Coherence* (i.e., is the summary easy to read and does it follow a natural ordering of facts?), and *Fluency* (i.e., is the summary fluent and grammatical?). We randomly selected 50 instances from the test set for each dataset (i.e. Rotten Tomatoes and Yelp). We collected five judgments for each comparison. The order of summaries was randomized per participant. A rating per system was computed as the percentage of times it was chosen as best minus the percentage

	Rotten Tomatoes			Yelp		
Model	Informative	Coherent	Fluent	Informative	Coherent	Fluent
SENTINEURON	11.8	8.3	**25.4**	-24.8	-0.8	**9.3**
MEANSUM	-32.1	-34.4	-46.8	6.3	-7.5	-10.8
DENOISESUM	**20.3**	**26.1**	21.4	**18.5**	**8.2**	1.6

Table 3.5: Best-worst scaling evaluation. Between systems differences are all significant, using a one-way ANOVA with posthoc Tukey HSD tests ($p < 0.01$).

of times it was selected as worst. Results are reported in Table 3.5. DENOISESUM was ranked best in terms of informativeness and coherence, while the extractive system SENTINEURON was ranked best on fluency. This is not entirely surprising since extractive summaries written by humans are by definition grammatical.

Our second study examined the veridicality of the generated summaries, namely whether the facts mentioned in them are indeed discussed in the input reviews. Participants were shown reviews and the corresponding summary and were asked to verify for each summary sentence whether it was fully supported by the reviews, partially supported, or not at all supported. We performed this experiment on Yelp only since the number of reviews is small and participants could read them all in a timely fashion. We used the same 50 instances as in our first study and collected five judgments per instance. Participants assessed the summaries produced by MEANSUM and DENOISESUM. We also included GOLD-standard summaries as an upper bound but no output from an extractive system as it by default contains facts mentioned in the reviews. We refer the readers to Appendix A.2 for the full instructions of the experiment.

Table 3.6 reports the percentage of fully, partially, and un-supported sentences. Gold summaries display the highest percentage of fully supported sentences (63.3%), followed by DENOISESUM (55.1%), and MEANSUM (41.7%). These results are encouraging, indicating that our model hallucinates to a lesser extent compared to MEANSUM.

3.5 Summary

We consider a learning setting for opinion summarization where there are only reviews available without corresponding summaries. Our key insight is to enable the use of supervised techniques by creating synthetic review-summary pairs using noise

	Yelp		
Model	Full Support	Partial Support	No Support
MEANSUM	41.7%	20.4%	38.0%
DENOISESUM	55.1%	24.3%	20.5%
GOLD	63.6%	23.6%	12.8%

Table 3.6: Summary veridicality evaluation. In the "Full Support" column, pairwise differences are statistically significant (using chi-squared test; $p < 0.05$).

generation methods. Our summarization model, DENOISESUM, introduces explicit denoising, partial copy, and discrimination modules which improve overall summary quality, outperforming competitive systems by a wide margin.

Creating synthetic datasets through noising is based on the idea that not all opinions in the input set of reviews are necessarily mentioned in the opinion summary, therefore these opinions are considered noise. However, the methods proposed in this chapter produce datasets that also include *grammatical noise* from segment noising. Moreover, in real world datasets, reviews may discuss a variety of opinions, where some of them are not salient and should not be included in the summary. However, while document noising ensures that the input reviews are similar with the summary, there is no mechanism to control variance such that the input reviews contain non-salient opinions. In the next chapter, we will describe a method that creates synthetic datasets that are more natural and resembles real world review-summary pairs.

Movie: “Iron Man 2”	
Synopsis	In ”Iron Man 2,” the world is aware that billionaire inventor Tony Stark is the armored Super Hero Iron Man. Under pressure from the government, the press and the public to share his technology with the ...
Released Date	May 7, 2010
Genre	Action & Adventure, Science Fiction & Fantasy
Director	Jon Favreau
Actors	Robert Downey Jr. as Tony Stark, Gwyneth Paltrow as Virginia ’Pepper’ Potts, Don Cheadle as Colonel James ’Rhodey’ Rhodes, ...
GOLD	It isn’t quite the breath of fresh air that Iron Man was, but this sequel comes close with solid performances and an action-packed plot.
SENTINEURON	Flabby, disjointed, and eschewing conflict for extended scenes of improv clowning, it’s the superheroic equivalent of a rat pack film.
MEANSUM	... the movie has too many twists in its own way, but it’s a bit too busy.
DENOISESUM	Iron Man 2 isn’t as good as the first movie, but it is a fun and fascinating film.

Figure 3.4: Examples of opinion summaries generated by three systems on the Rotten Tomatoes dataset. We also show the human-generated consensus summary (GOLD), as well as basic background information about the movie.

Business: "Noodle Pot"	
Location	Las Vegas
Categories	Noodles, Specialty Food, Restaurants, Food, Ethnic Food, Chinese, Taiwanese
GOLD	This is a decent place to go for Chinese food. I really like the beef noodle soup. It has the perfect amount of salt. I wish the place wasn't so small and stuffy. I just didn't feel comfortable at all. Next time I'll just place a to go order.
SENTINEURON	The restaurant is really tiny and more of a cafe. The beef stew noodle is so perfect. Not too salty. Just enough beef, bok choy, and handmade noodles to satisfy any appetite. The pork chop noodle was also very good. It is a milder soup but the saltiness of the pork chop helps to balance it. The house-made chili gives everything the perfect kick and the burn pleasantly creeps up on you. The wontons in red sauce are also spectacular. I believe the red sauce has a vinegar base with hints of sweet and spicy. The restaurant is the perfect place to go during the winter.
MEANSUM	This place has the best beef noodle soup. I had the chicken with broccoli and it was good. Its not spicy at all but not bad either. I have been coming here for over 10 years and have been back many times. They have a wide variety of dishes, not too oily, and good. The staff is friendly, clean, never disappointed.
DENOISESUM	I love this place. The restaurant is really clean and very friendly. The beef noodle soup is very good and the beef is so perfect. The food is very filling. I receive really good service here and recommend the place when you are in Vegas. The only complaint is that there's always a line every time I go here. I am definitely going back here again.

Figure 3.5: Examples of opinion summaries generated by two systems on the Yelp dataset. We also show the human-generated summary (GOLD), as well as basic background information of the restaurant.

Chapter 4

Content Planning in Opinion Summarization

In the previous chapter, we presented a method to create synthetic opinion summarization datasets through noising. Although our approach enables the use of supervised training and produces higher quality summaries than autoencoder-based methods (Chu and Liu, 2019; see previous chapter), it does not, by definition, generate datasets which resemble real-world data. Segment noising generates syntactic noise which can result to ungrammatical text, while document noising extracts lexically similar reviews under the unrealistic assumption that all reviews with overlapping vocabulary will be semantically similar to the summary. As shown in Figure 4.1, real-world reviews discuss a variety of opinions covering different aspects of the entity under consideration (e.g., for a bar it might be the price of the drinks, the staff, the atmosphere of the place). Some of these opinions reached consensus and thus are considered salient, and we expect to see them mentioned in the summary, while others will be less salient and absent from the summary. There is also variety among reviews: some will focus on several aspects, others on a single one, and there will be some which will discuss idiosyncratic details.

In this chapter, we propose to incorporate *content planning* in unsupervised opinion summarization. The generation literature provides multiple examples of content planning components (Kukich, 1983; McKeown, 1985) for various domains and tasks including data-to-text generation (Gehrmann et al., 2018; Puduppully et al., 2019), argument generation (Hua and Wang, 2019), and summarization (Kan and McKeown, 2002). Aside from guiding generation towards more informative text, we argue that content plans can be usefully employed to reflect a natural variation of sampled reviews in creating a synthetic dataset. Our content plans take the form of aspect and

Input Reviews
1. *Local dive bar experience! Authentic phoenix experience squished behind the starbucks.* Pros: Decent prices, $2 mystery shots, *clean bathroom ...*
2. Cheap drinks, awesome bar staff, *stiff pours ...*
3. Cheap drinks, great happy hour (that's ridiculously long and cheap) ... I've only found great bartenders and patrons at this little bar ...
4. It's a local bar with *no frills except pool table, bar,* and friendly people ... *The sliding glass door with the little beach is what makes this place awesome!!! ...*
5. Bartender was friendly and made great shots, *but the place was full of regulars who made it impossible to have fun ...*
6. *Their Christmas decorations rival that of coach house but without the Scottsdale crowd.* You can find every type of person hanging out here. The staff is friendly ...
7. ... reminds me of back home in the Mid West. Good times and great spot to mingle and meet new people!
8. Lynn is the reason I continue to come back!! She is personable, fun, and dedicated.
Opinion Summary
The drinks here are well priced, especially during happy hour. There is a large variety of regulars from various backgrounds and ages. Great place to meet new people. The staff are great they provide a nice judgement free environment and they aren't stingy on the pours.

Figure 4.1: Yelp reviews about a local bar and corresponding summary. Aspect-specific opinions are in color (e.g., drinks, guests, staff), while less salient opinions are shown in *italics*.

sentiment probability distributions which are induced from data without access to expensive annotations. Using these as parameters to Dirichlet distributions, we create a synthetic dataset of review-summary pairs, where the variation of aspect mentions among reviews can be controlled. We also propose an opinion summarization model that uses these distributions as a content plan to guide the generation of abstractive summaries.

Experiments on three datasets (Wang and Ling, 2016; Chu and Liu, 2019; Bražinskas et al., 2019) representing different domains (movies, business, and product reviews)

and summarization requirements (short vs longer summaries) show that our approach outperforms competitive systems in terms of ROUGE, achieving state of the art across the board. Human evaluation further confirms that the summaries produced by our model capture salient opinions as well as being coherent and fluent.

4.1 Related Work

Without gold-standard summaries during training, text generation methods (Freitag and Roy, 2018; Fevry and Phang, 2018; Chu and Liu, 2019) conventionally make use of variational autoencoders (Kingma and Welling, 2014), while employing relatively simple decoders in order to mitigate posterior collapse (Kingma and Welling, 2014; Bowman et al., 2016). A more recent line of work (Bražinskas et al., 2019; Amplayo and Lapata, 2020; see previous chapter) creates synthetic datasets in cases where gold standard summaries are not available which in turn allow to train models in a supervised setting and make use of of effective decoding techniques such as attention and copy. Our method is in line with this work, but ultimately different in its use of content planning to guide both text summarization and synthetic data creation.

Content plans have been successfully used to improve natural language generation performance in both traditional and neural-based systems. Traditionally (Kukich, 1983; McKeown, 1985), content planning is the first part of a pipeline of modules preceding sentence planning (i.e., determining the structure and lexical content of sentences) and surface realization (i.e., converting the sentence plan to natural language). In recent neural-based systems (Gehrmann et al., 2018; Puduppully et al., 2019), it is incorporated as a module into neural network models to select which information from the input is salient and should be reflected in the output. Content plans are often discrete and designed with a specific task and domain in mind. Examples include a sequence of facts for data-to-text generation (Gehrmann et al., 2018; Moryossef et al., 2019; Puduppully et al., 2019), a list of Wikipedia key-phrases for argument generation (Hua and Wang, 2019), and entity mentions and their clusters in news summarization (Amplayo et al., 2018b; Sharma et al., 2019). Our content plans are neither discrete nor domain-specific. They take the form of aspect and sentiment distributions, and serve the dual purpose of creating more naturalistic datasets for model training and guiding the decoder towards more informative summaries. Finally, to the best of our knowledge, our work is the first to explore the use of content planning for opinion summarization.

4.2 Modeling Approach

We assume access to a collection of reviews $C = \{R_{e_1}, ..., R_{e_{|E|}}\}$, where $R_e = \{r_1, ..., r_{|R_e|}\}$ is a list of reviews about a specific entity $e \in E$ (e.g., a movie, product, business). Each review r has a sentiment rating s which can be either binary (e.g., positive or negative) or on a scale (e.g., from 1 to 5). We further assume that reviews typically focus on one or more aspects among a set of aspects $A_e = \{a_1, ..., a_{|A_e|}\}$ for entity e (e.g., the price and image quality of a television, the acting and plot of a movie). Finally, we do not assume access to gold-standard summaries, since in most domains these do not exist.

Our method consists of three parts. For each review r, we first induce aspect and sentiment probability distributions $p(a)$ and $p(s)$. We do this with a content plan induction model which learns to reconstruct the review from aspect and sentiment embeddings (Section 4.2.1). Distributions $p(a)$ and $p(s)$ are then used to create a synthetic dataset $\hat{D} = \{(\hat{R}, \hat{y})\}$ of review-summary pairs. We make use of the Dirichlet distribution parameterized with $p(a)$ and $p(s)$ for sampling, which ensures that the reviews are naturally varied and the summary is representative of the opinions found in the reviews (Section 4.2.2). Finally, we generate opinion summary y using a summarization model, which is conditioned on the input reviews R, but also guided by distributions $p(a)$ and $p(s)$, which we view as a content plan (Section 4.2.3).

4.2.1 Content Plan Induction

Our content plan induction model is illustrated in Figure 4.2. It induces probability distributions $p(a)$ and $p(s)$ from review r by learning aspect and sentiment embeddings, and reconstructing the encoding of r through these embeddings. It is similar to neural topic models that are used for aspect extraction (He et al., 2017; Angelidis and Lapata, 2018b), but also extracts sentiment from reviews. Unlike in generic text where documents are assumed to be grouped solely by latent topics (Blei et al., 2003), in review domains, there are two variables we want to classify our text, the aspect and the sentiment.

We encode review $r = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$ using a BiLSTM (Hochreiter and Schmidhuber, 1997) followed by a mean pooling operation, where $\mathbf{x}_i$ is the embedding of the ith token in r. The output encoding is divided into two chunks to represent aspect- and sentiment-specific document encodings, $\mathbf{h}_a$ and $\mathbf{h}_s$, respectively, which are used in

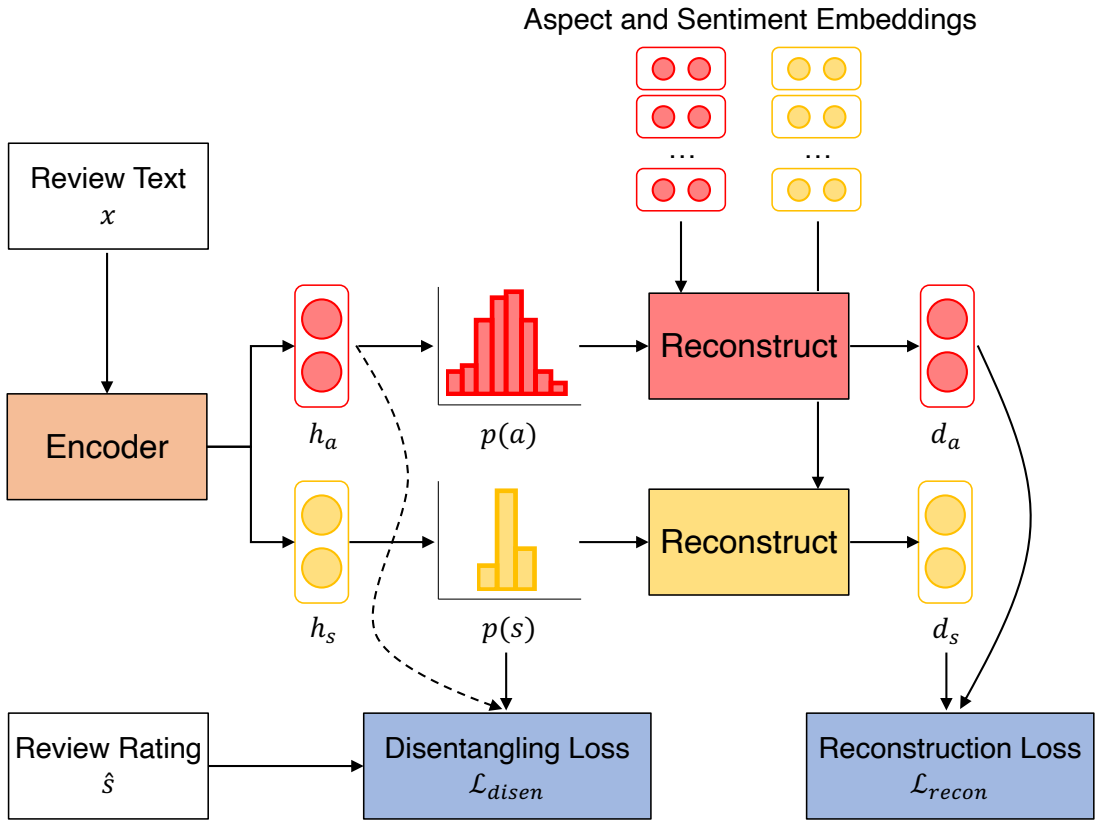


Figure 4.2: Model architecture of our content plan induction model. Aspect- and sentiment-specific modules are colored red and yellow, respectively. The dotted line indicates that a reverse gradient function is applied.

softmax classifiers to obtain distributions $p(a)$ and $p(s)$ (see Figure 4.2):

$$\{\mathbf{h}_i\} = \text{BiLSTM}(\{\mathbf{x}_i\}) \tag{4.1}$$

$$\mathbf{h}_a, \mathbf{h}_s = \text{chunk}(\textstyle\sum_i \mathbf{h}_i / N, 2) \tag{4.2}$$

$$p(a) = \text{softmax}(\mathbf{W}_a \mathbf{h}_a + \mathbf{b}_a) \tag{4.3}$$

$$p(s) = \text{softmax}(\mathbf{W}_s \mathbf{h}_s + \mathbf{b}_s) \tag{4.4}$$

where N is the number of review tokens, $\text{chunk}(\cdot, 2)$ is the operation to split an encoding into two equal chunks, $\mathbf{W}$ and $\mathbf{b}$ are model parameters.

We learn aspect and sentiment embedding matrices $\mathbf{A} \in \mathbb{R}^{|A| \times d}$ and $\mathbf{S} \in \mathbb{R}^{|S| \times d}$, where d is the embedding size, $|A|$ and $|S|$ is the number of *latent* aspects and sentiments, via reconstructing the review. Formally, we obtain reconstructions $\mathbf{d}_a$ and $\mathbf{d}_s$ by

weight-summing embeddings using $p(a)$ and $p(s)$:

$$\mathbf{d}_a = \sum_i \mathbf{A}[i] * p(a_i) \tag{4.5}$$

$$\mathbf{d}_s = \sum_i \mathbf{S}[i] * p(s_i) \tag{4.6}$$

The model is trained using two different objectives. Firstly, a contrastive max-margin objective function is used to reconstruct the original encodings $\mathbf{h}_a$ and $\mathbf{h}_s$ with $\mathbf{d}_a$ and $\mathbf{d}_s$, respectively. For each review, we randomly sample m reviews as negative samples and obtain their corresponding aspect and sentiment encodings $\{\mathbf{n}_a^{(i)}, \mathbf{n}_s^{(i)}\}$ for $1 \leq i \leq m$. We formulate the objective function as a hinge loss $\mathcal{L}_{recon}$ that maximizes the inner product between $\mathbf{d}_a$ and $\mathbf{d}_s$ and the original encodings and minimizes the inner product between $\mathbf{d}_a$ and $\mathbf{d}_s$ and the negative samples. Furthermore, the embeddings should refer to different aspects/sentiments and thus should be unique from each other. To encourage uniqueness of each embeddings, we additionally ensure diversity among aspect and sentiment embeddings in memory (He et al., 2017) by adding a regularization term $\mathcal{R}_{recon}$:

$$\begin{aligned}\mathcal{L}_{recon} = &\sum_i \max(0, 1 - \mathbf{d}_a\mathbf{h}_a + \mathbf{d}_a\mathbf{n}_a^{(i)}) \\ &+ \sum_i \max(0, 1 - \mathbf{d}_s\mathbf{h}_s + \mathbf{d}_s\mathbf{n}_s^{(i)})\end{aligned} \tag{4.7}$$

$$\mathcal{R}_{recon} = \|\mathbf{A}\mathbf{A}^\top - \mathbf{I}\| + \|\mathbf{S}\mathbf{S}^\top - \mathbf{I}\| \tag{4.8}$$

where $\mathbf{I}$ is the identity matrix. $\mathcal{R}_{recon}$ minimizes the dot product between two different embeddings in each of the aspect and sentiment embedding matrices $\mathbf{A}$ and $\mathbf{S}$, encouraging orthogonality.

We also ensure that the aspect embedding matrix $\mathbf{A}$ does not include information regarding sentiment, and vice versa, by adding a disentanglement loss $\mathcal{L}_{disen}$. This is important since we want to use aspect information to plan the summary content without bias towards a certain sentiment. To distinguish sentiment information, we leverage review ratings $\hat{s}$ as sentiment labels and learn an *adversarial* classifier. The adversarial classifier is trained to remove sentiment information from aspect encoding $\mathbf{h}_a$.

Specifically, we use softmax as the adversarial classifier, which accepts as input aspect encoding $\mathbf{h}_a$ and returns an adversarial sentiment distribution $p(s)_{adv}$. The classifier uses a reverse gradient function (Ganin et al., 2016) which is an identity function during forward propagation, but reverses the sign of the gradient during backward propagation:

$$p(s)_{adv} = \text{softmax}(\text{GradRev}(\mathbf{W}_{adv}\mathbf{h}_a + \mathbf{b}_{adv})) \tag{4.9}$$

where GradRev is the reverse gradient function. This would help in separating sentiment information from the aspect embedding matrix. Finally, the cross-entropy functions for both the original and adversarial sentiment distributions $p(s)$ (Equation 4.4) and $p(s)_{adv}$ are used as the disentanglement loss:

$$\mathcal{L}_{disen} = -\log p(\hat{s}) - \log p(\hat{s})_{adv} \tag{4.10}$$

The overall training loss is the linear addition of the reconstruction and disentanglement losses, and the regularization term mentioned above (λ is a hyperparameter controlling the regularization):

$$\mathcal{L}_{induce} = \mathcal{L}_{recon} + \mathcal{L}_{disen} + \lambda \mathcal{R}_{recon} \tag{4.11}$$

After training, we obtain probability distributions $p(a)$ and $p(s)$ for each review, and use them to create a synthetic dataset (Section 4.2.2) and train a summarization model (Section 4.2.3).

4.2.2 Synthetic Dataset Creation

To create synthetic dataset $\hat{D} = \{\hat{R}, \hat{y}\}$, we first sample a review from the corpus and pretend it is summary $\hat{y}$. Next, we sample a set of reviews $\hat{R}$ conditioned on $\hat{y}$ and pretend they serve as the input which led to summary $\hat{y}$. We impose a few (stylistic) constraints on the selection of candidate summaries to ensure that they resemble actual summaries. We discuss these constraints in Section 4.3.

Review samples are created such that they follow the variation of aspect and sentiment mentions in the sampled summary. Specifically, we use a Dirichlet distribution, the conjugate prior of the multinomial distribution, to sample N pairs of aspect and sentiment distributions. Given pseudo-summary $\hat{y}$ and its induced distributions $p(a)$ and $p(s)$, the ith pair of aspect and sentiment distributions $\{(p_i(a)p_i(s))\}$, $1 \leq i \leq N$ is sampled as:

$$p_i(a) \sim \text{Dirichlet}(\alpha_a * p(a)) \tag{4.12}$$

$$p_i(s) \sim \text{Dirichlet}(\alpha_s * p(s)) \tag{4.13}$$

where α_a and α_s are constants which control the variance of the distributions sampled from the Dirichlet. Sampling a multinomial distribution $p = \{x_1, ..., x_K\}$ from the Dirichlet with parameter α can be done using the following probability density function:

$$\text{Dirichlet}(\alpha) = \frac{1}{\text{Beta}(\alpha)} \prod_{i=1}^{K} x_i^{\alpha_i - 1} \tag{4.14}$$

where Beta$(\cdot)$ is the beta function.

When α values are small, $p(a)$ and $p(s)$ will look more different from the distribution of the summary, and when α values are larger, the sampled distributions will look more similar to the summary. Sampling from the Dirichlet ensures that the average of the sampled distribution equals that of the summary allowing us to control how the synthetic dataset is created modulating how aspect and sentiment are represented.

Figure 4.3 shows three examples of sampled reviews given a candidate summary and with different α values. We also report the average ROUGE scores between the reviews and the candidate summary. As can be seen, ROUGE increases as the α value increases, which means that the sampled reviews get more similar to the summary the larger the value is. Another way to interpret this is that the review sampling becomes random when the constant approaches zero, while review sampling uses the similarity function when it approaches infinity.

Finally, for each sampled pair $(p_i(a), p_i(s))$, we run a nearest neighbor search over the corpus to find the review r_i with the most similar pair of distributions. We use Hellinger (1909) distance to quantify the similarity between two distributions, i.e.. $sim(p,q) = \|\sqrt{p} - \sqrt{q}\|_2/\sqrt{2}$ (we take the average of the similarity scores between aspect and sentiment distributions). This results to an instance within dataset $\hat{D}$, where $\hat{R} = \{r_1, ..., r_N\}$ is the set of reviews for pseudo-summary $\hat{y}$. We repeat this process multiple times to obtain a large-scale training dataset.

4.2.3 Opinion Summarization

We use the synthetic dataset $\hat{D}$ to train our summarization model which we call PLANSUM and illustrate in Figure 4.4. A fusion module aggregates token-level encodings in input reviews R to reduce the number of tokens. The fused encodings are then passed to a decoder that uses the mean aspect and sentiment distributions as a content plan to generate output summary y. We do not employ an encoder in our model, but rather reuse the encodings from the content plan induction model, which improves memory-efficiency in comparison to related architectures (Chu and Liu, 2019; Bražinskas et al., 2019; Amplayo and Lapata, 2020). At test time, the same model is used to summarize actual reviews.

Mean and Injective Fusion For the ith review $r_i \in R$ with tokens $\{w_j^{(i)}\}$, we obtain contextualized token encodings $\{\mathbf{h}_j^{(i)}\}$ and probability distributions $p^{(i)}(a)$ and $p^{(i)}(s)$,

Candidate Summary
I bought one for my mother some years ago due to her arthritis. It worked for her. I bought one for myself and then for all my family members. I don't wish to spend much of my life peeling veggies, but when I do it has to be this product. The soft grip is very helpful in avoiding discomfort, fatigue, and pain.
Reviews when $\alpha = 1.0$ Review-Summary ROUGE: **24.75/3.06/15.54**
1. This is without a doubt the best peeler I've ever used. My hands never got tired ...
2. I'm a sucker for shiny, expensive things, so of course I bought the $27 stainless steel ...
3. I am a sharp and easy to use peeler. I boast a sleek design and wide handle. I glide ...
4. This is our second one we have purchased. The first one lasted for many years ...
5. I should have gotten one of these years ago. Fits nicely in my hand and peels great ...
6. Had to find replacement for my mother's peeler that I'd used for 20 years. This one ...
7. I really like this peeler. It is really smooth, easy to clean, and hold fairly well. I do ...
8. As a vegan, I work with a lot of vegetables. As a result, this peeler is practically an ...
Reviews when $\alpha = 10.0$ Review-Summary ROUGE: **26.34/3.46/16.81**
1. I am a sharp and easy to use peeler. I boast a sleek design and wide handle. I glide ...
2. This is our second one we have purchased. The first one lasted for many years ...
3. moved into a new apartment and these are obviously a must have for any cook ...
4. I got mine years ago and just bought 2 for family members. Super easy to hold ...
5. Bought this from my grandma in the caribbean so she doesn't have to use a kitchen ...
6. This swivel peeler works so well!! I even peel mango using this peeler, and it ...
7. As the man in the house who cooks, I always appreciate good tools. This peeler ...
8. I have been using this peeler for quite some time now. it does it job perfectly well ...
Reviews when $\alpha = 100.0$ Review-Summary ROUGE: **30.56/4.15/18.93**
1. I am a sharp and easy to use peeler. I boast a sleek design and wide handle. I glide ...
2. I should have gotten one of these years ago. fits nicely in my hand and peels great ...
3. Bought this from my grandma in the caribbean so she doesn't have to use a kitchen ...
4. Had to find replacement for my mother's peeler that I'd used for 20 years. This one ...
5. I have been using this peeler for quite some time now. It does it job perfectly well ...
6. I did order two, but now I only have one. My daughter was at my place helping me ...
7. My wife's nascent arthritis can make it hard for her to grip small handles, but she ...
8. It is my first peeler, so I can hardly compare, but it works very well, very smooth ...

Figure 4.3: Examples of sampled reviews given a candidate summary, when the Dirichlet constant α is varied (Amazon dataset). For simplicity, we use the same value for both α_a and α_s. The ROUGE scores are used to measure the similarity between the reviews and the candidate summary.

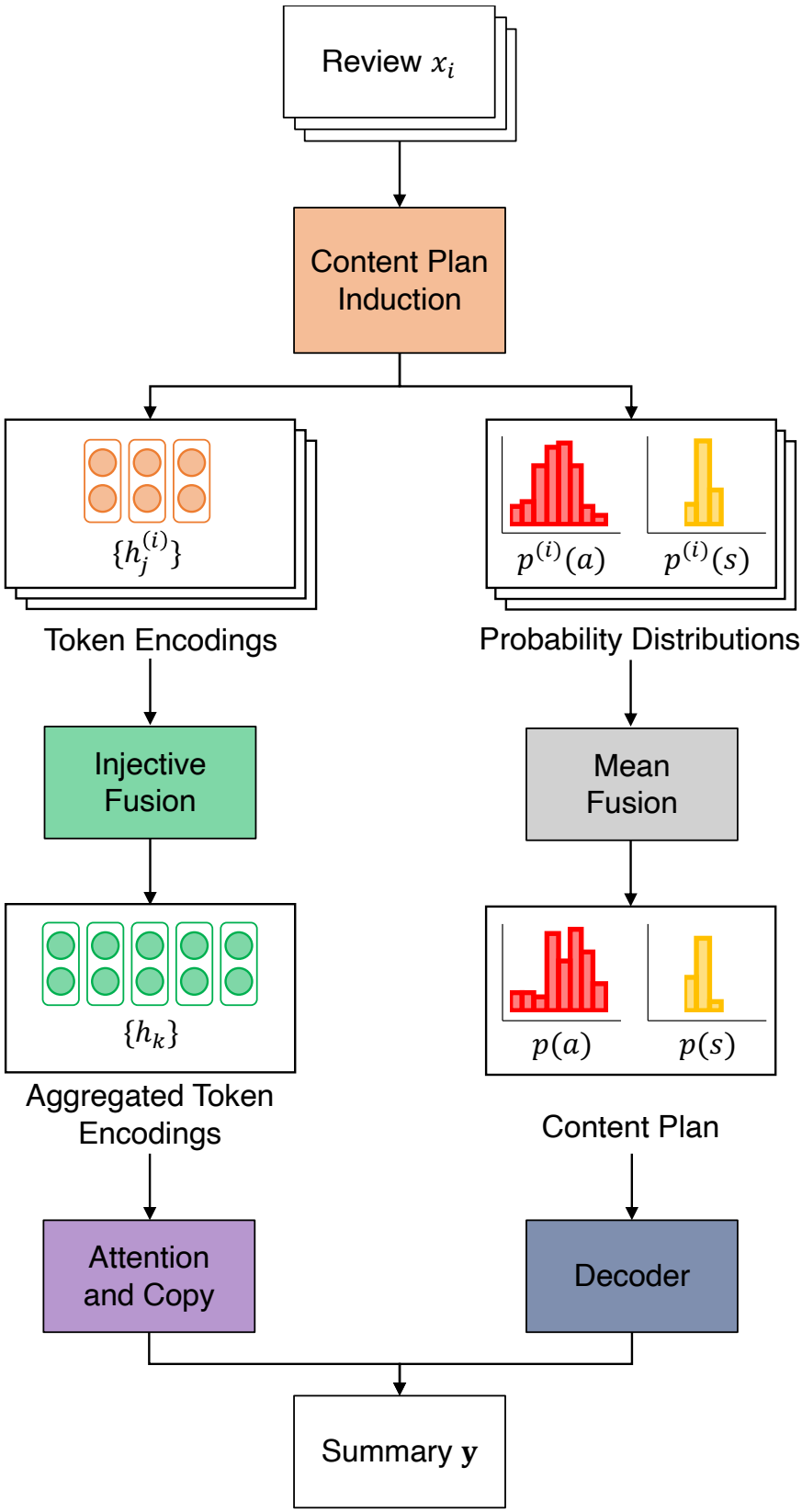


Figure 4.4: Model architecture of PLANSUM (information flows from top to bottom). The content plan is constructed as the average of the aspect and sentiment probability distributions induced by the content plan induction model. It is then passed to the decoder, along with the aggregated token encodings to generate the summary.

using the content plan induction in Equation (4.1). We then aggregate these encodings and distributions to collectively represent the set of input reviews.

It is trivial to aggregate aspect and sentiment distributions since the synthetic dataset is by construction such that their average equals to the summary. We thus take their mean as follows:

$$p(a) = \sum_i p^{(i)}(a)/N \tag{4.15}$$

$$p(s) = \sum_i p^{(i)}(s)/N \tag{4.16}$$

On the other hand, it is nontrivial to fuse token embeddings as the number of input tokens can be prohibitively large causing out-of-memory issues. We could fuse token embeddings by aggregating over the same word, especially since multiple reviews are highly redundant. However, simple aggregation methods such as mean and max pooling may be all too effective at eliminating redundancy since they cannot retain information regarding token frequency. This would be problematic for our task, redundancy is an important feature of opinion summarization, and repetition can indicate which aspects are considered important. To mitigate this, we borrow a fusion method from graph neural networks (Xu et al., 2019) that uses an injective function, to effectively discriminate representations of the same token but with different levels of redundancy:

$$\mathbf{h}_k = \text{MLP}(\mathbf{e}_k + \sum_{(i,j):w_j^{(i)}=w_k} \mathbf{h}_j^{(i)}) \tag{4.17}$$

where $\mathbf{e}_k$ is a learned embedding for the token w_k in the vocabulary, and $w_j^{(i)}$ (and $\mathbf{h}_j^{(i)}$) is the jth token (and its encoding) of the ith review.

Decoder with Content Planning Our decoder is an LSTM equipped with attention (Bahdanau et al., 2014) and copy (Vinyals et al., 2015) mechanisms, where the aggregated token encodings $\{\mathbf{h}_k\}$ are used as keys. Additionally, at each timestep, the decoder makes use of the aggregated probability distributions $p(a)$ and $p(s)$ as a content plan. This guides the model towards generating correct aspect and sentiment information. Specifically, we use embedding matrices $\mathbf{A}$ and $\mathbf{S}$ from the content plan induction model to obtain aspect and sentiment encodings $\mathbf{d}_a$ and $\mathbf{d}_s$, using Equations (4.5) and (4.6). We then combine these encodings with the previous output token y_{t-1} to predict

the next token y_t at timestep t:

$$y'_t = f(d_a, d_s, y_{t-1}) \tag{4.18}$$

$$s_t = \text{LSTM}(y'_t, s_t) \tag{4.19}$$

$$p(y_t) = \text{ATTENDCOPY}(y'_t, s_t, \{h_k\}) \tag{4.20}$$

where $f(\cdot)$ is a linear function.

Training and Inference We use a maximum likelihood loss to optimize the probability distribution based on summary y. Additionally, we apply a label smoothing method (Szegedy et al., 2016) as a form of regularization of the output vocabulary distribution. We extend this method to use BERT (Devlin et al., 2019) predictions as the prior distribution, which is a more informed prior compared to the commonly used uniform distribution, i.e.:

$$y'_t = (1-\delta) * y_t + \delta * \text{BERT}(y_{-t}) \tag{4.21}$$

$$\mathcal{L}_{gen} = -\sum_t y'_t \log p(y_t) \tag{4.22}$$

where $\text{BERT}(\cdot)$ outputs the tth token prediction given all the previous tokens.

4.3 Experimental Setup

4.3.1 Datasets

In this chapter, we performed experiments on three opinion summarization benchmarks. These include the Rotten Tomatoes dataset[1] (Wang and Ling, 2016) which contains a large set of reviews for various movies written by critics. Each set of reviews has a gold-standard opinion summary written by an editor. However, we do not use ground truth summaries for training, to simulate our unsupervised setting. Our second dataset is Yelp[2] (Chu and Liu, 2019) which includes a large training corpus of reviews for businesses without gold-standard summaries, as well as development and test sets with summaries generated by Amazon Mechanical Turk (AMT) crowdworkers. Finally, the Amazon dataset[3] (Bražinskas et al., 2019) contains product reviews for four Amazon categories: *Electronics*, *Clothing, Shoes and Jewelry*, *Home and Kitchen*,

[1] http://www.ccs.neu.edu/home/luwang/data.html
[2] https://github.com/sosuperic/MeanSum
[3] https://github.com/ixlan/Copycat-abstractive-Amazon-product-summaries

and *Health and Personal Care*. The development and test partitions come with three gold-standard reference summaries produced by AMT annotators. All datasets include review ratings which we used as sentiment labels: Rotten Tomatoes has binary labels, while Yelp and Amazon have a 1–5 scale. We refer the readers to Chapter 2 for details and example reviews and summaries of the datasets.

To create synthetic training data, we sampled candidate summaries using the following constraints we also used in the previous chapter (we refer the readers to Chapter 3 for our justification on using these constraints): (1) there must be no non-alphanumeric symbols aside from punctuation, (2) there must be no first-person singular pronouns (not used in Yelp/Amazon), and (3) the number of tokens must be between 50–90 (20–50 for Rotten Tomatoes). We also made sure that sampled reviews and candidate summary discuss the same entity. After applying these constraints we obtained 100k (Yelp), 25k (Rotten Tomatoes), and 90k (Amazon) review-summary pairs. Statistics of these datasets are reported in Table 4.1. As can be seen, RT contains the largest number of input reviews but the shortest summaries (22–35 tokens). While Amazon and Yelp have a smaller number of input reviews but longer summaries (66–70.9 and 62.5–59.8 tokens, respectively).

4.3.2 Training Configuration

Across models, we set all hidden dimensions to 256, the dropout rate to 0.1, and batch size to 16. We used the subword tokenizer of BERT (Devlin et al., 2019), which has a 30K token vocabulary trained using WordPiece (Wu et al., 2016b). For Rotten Tomatoes, we follow Wang and Ling (2016) and add a generic label for movie titles during training which we replace with the original title during inference. We used the Adam optimizer (Kingma and Ba, 2015) with a learning rate of $3e-4$, l_2 constraint of 3, and warmup of 8,000 steps. We also used dropout (Srivastava et al., 2014) after every non-linear function. For each dataset, we additionally tuned the number of aspects $|A|$, regularization parameter λ, Dirichlet parameters α_a and α_s, label smoothing parameter δ, and beam search size on the development set. We performed early stopping based on the token-level accuracy of the model, again on the development set. Our model was trained on a single GeForce GTX 1080Ti GPU and is implemented using PyTorch.

Yelp	Train*	Dev	Test
#summary	100k	100	100
#reviews	8.0	8.0	8.0
#tokens/summary	66.0	70.9	67.3
#tokens/review	65.7	70.3	67.8
corpus size			2,320,800

Rotten Tomatoes	Train*	Dev	Test
#summary	25k	536	737
#reviews	72.3	98.0	100.3
#tokens/summary	25.8	23.6	23.8
#tokens/review	22.9	23.5	23.6
corpus size			245,848

Amazon	Train*	Dev	Test
#summary	90k	28×3	32×3
#reviews	8.0	8.0	8.0
#tokens/summary	59.8	60.5	62.5
#tokens/review	55.8	56.0	56.0
corpus size			1,175,191

Table 4.1: Dataset statistics; Train* column refers to the synthetic data we created (Section 4.2.2). Amazon contains three reference summaries (× 3) per instance.

4.3.3 Comparison Systems

We compared PLANSUM to several previously proposed approaches. Extractive systems include:

1. LEXRANK (Erkan and Radev, 2004), a PageRank-like algorithm that selects the most salient sentences from the input (see Section 3.3.3 for details).

2. CENTroid-based methods (Radev et al., 2004) which select as summary the review closest to the centroid, i.e., the mean review representation. We present three centroid methods that use different input representations, such as (2.1) W2VCENT that uses weight-summed word2vec embeddings (Mikolov et al., 2013) learned using the review corpus (Rossiello et al., 2017; see Section 3.3.3

for details), (2.2) SNCENT that uses encodings from Sentiment Neuron, an LSTM-based language model trained on a large-scale Amazon review corpus (Amplayo and Lapata, 2020; see Section 3.3.3), and (2.3) BERTCENT, which uses pre-trained encodings from BERT (Devlin et al., 2019), a large transformer-based language model trained using huge amounts of data. Specifically, given a review, we obtain its encoding representation as the average of its token encodings obtained from BERT. We then use the review closest to the centroid as the extractive summary.

Abstractive comparison systems include (see Section 3.3.3 for detailed model descriptions):

3. OPINOSIS (Ganesan et al., 2010), a graph-based method that uses token-level redundancy to generate summaries.

4. MEANSUM (Chu and Liu, 2019), an autoencoder that generates summaries by reconstructing the mean of review encodings.

5. DENOISESUM (Amplayo and Lapata, 2020; Chapter 3), a denoising model that treats non-salient information as noise and removes them to generate a summary.

6. COPYCAT (Bražinskas et al., 2019), a hierarchical variational autoencoder which learns a latent code of the summary.

4.4 Results

4.4.1 Automatic Evaluation

We evaluated the quality of opinion summaries using F_1 ROUGE (Lin and Hovy, 2003). Unigram and bigram overlap (ROUGE-1 and ROUGE-2) are a proxy for assessing informativeness while the longest common subsequence (ROUGE-L) measures fluency.

Our results are summarized in Table 4.2. Among extractive models, BERTCENT performs best, indicating that representations from large transformer-based language models can be used as a simple method to produce good extractive summaries. Extractive models, however, are consistently worse than neural-based abstractive models. Amongst the latter, PLANSUM performs best across datasets and metrics save in terms of ROUGE-L on Amazon. The slight better performance of COPYCAT suggests that

the use of a VAE objective may also be beneficial for our model, however we leave this to future work. Especially on Yelp, we observe a large improvement, with an increase of 5.32, 1.75, and 1.65 points in ROUGE-1/2/L over the best comparison systems.

We present in Table 4.3 various ablation studies on the three datasets, which assess the contribution of different model components. Firstly, we compare with variants of PLANSUM that use less expressive content plan induction modules, without either disentangling loss or regularization loss. Results show that both variants perform worse than our final model, confirming that both aspect and sentiment disentanglement and embedding regularization in the content plan induction module improve performance. Secondly, we also compare with PLANSUM that is trained using a different review sampling strategy, i.e., random and similarity sampling. Our experiments show that our dataset creation method that incorporates content planning when sampling reviews outperforms alternative strategies. This is especially the case on Rotten Tomatoes, where there is an 1.92 decrease in ROUGE-L. Rotten Tomatoes differs from Amazon and Yelp in that the input reviews are multiple (in the excess of 50) and thus contains more variety which our content planning approach manages to capture and reproduce in generating the synthetic data. Finally, we show comparisons between our summarization model and three variants: (a) one without using the probability distributions as content plan, (b) one using a mean fusion to aggregate token encodings (instead of injective fusion), and (c) one without incorporating BERT LM for label smoothing. Our final summarization model outperforms all these variants, which confirms that all the modules contribute to the performace of PLANSUM.

We show how content planning modulates summary output in Table 4.5 (for Yelp), Table 4.6 (for Rotten Tomatoes), and Table 4.7 (for Amazon). We present summaries produced by PLANSUM and variants without a content plan during synthetic data creation (see Random and Similarity Sampling) and in the summarization model (No Plan). Summaries from models without any planning whatsoever either miss out on salient aspects, or focus on aspects that do not reach consensus (i.e., aspect mentions absent from the summary). For example in Table 4.5, PLANSUM with random sampling is unable to generate all salient aspects, while the same model with similarity sampling outputs opinions about non-salient aspects. Using our content planning strategy to create a synthetic dataset, PLANSUM is able to generate opinion summaries containing only aspects that reach consensus.

	Model	ROUGE-1	ROUGE-2	ROUGE-L
Yelp	LexRank	25.50	2.64	13.37
	W2vCent	24.61	2.85	13.81
	SnCent	25.05	3.09	14.56
	BertCent	26.67	3.19	14.67
	Opinosis	25.15	2.61	13.54
	MeanSum	28.86	3.66	15.91
	DenoiseSum	<u>30.14</u>	4.99	17.65
	Copycat	29.47	<u>5.26</u>	<u>18.09</u>
	PlanSum	**34.79***	**7.01***	**19.74***

	Model	ROUGE-1	ROUGE-2	ROUGE-L
Rotten Tomatoes	LexRank	14.88	1.94	10.50
	W2vCent	13.93	2.10	10.81
	SnCent	15.90	2.01	11.74
	BertCent	17.65	2.78	12.78
	Opinosis	14.98	3.07	12.19
	MeanSum	15.79	1.94	12.26
	DenoiseSum	<u>21.26</u>	<u>4.61</u>	<u>16.27</u>
	Copycat	—	—	—
	PlanSum	**21.77***	**6.18**	**16.98***

	Model	ROUGE-1	ROUGE-2	ROUGE-L
Amazon	LexRank	28.74	5.47	16.75
	W2vCent	28.73	4.97	17.45
	SnCent	30.45	5.40	17.73
	BertCent	30.67	5.21	17.76
	Opinosis	28.42	4.57	15.50
	MeanSum	29.20	4.70	18.15
	DenoiseSum	—	—	—
	Copycat	31.97	5.81	**20.16**
	PlanSum	**32.87***	**6.12***	<u>19.05</u>

Table 4.2: Automatic evaluation on Yelp, Rotten Tomatoes, and Amazon datasets. Extractive/Abstractive models shown in first/second block. Best systems shown in bold and 2nd best systems are underlined; asterisk (*) means there is a significant difference between best and 2nd best systems (based on paired bootstrap resampling; $p < 0.05$).

Model	Yelp	Rotten Tomatoes	Amazon
PLANSUM	19.74	16.98	19.05
No disentangling	18.83	16.09	18.52
No regularization	19.00	16.85	18.92
Random sampling	19.22	16.61	18.70
Similarity sampling	19.38	15.06	18.31
No content plan	19.03	16.56	18.28
Mean token fusion	18.72	16.76	18.57
Uniform label prior	18.80	16.77	18.94

Table 4.3: Performance (based on ROUGE-L) of PLANSUM and versions thereof with less expressive plan induction (second block), using alternative review sampling methods (third block), and without some modules (fourth block).

4.4.2 Human Evaluation

We also conducted a judgment elicitation study using the Amazon Mechanical Turk crowdsourcing platform. We assessed the quality of system summaries using Best-Worst Scaling (Louviere et al., 2015). Specifically, we asked participants to select the *best* and *worst* among system summaries taking into account how much they deviated from given input reviews in terms of four criteria. The first two criteria assess informativeness and ask crowdworkers to select a summary based on whether it mentions the majority of *aspects* discussed in the original reviews and agrees with their overall *sentiment*. We also evaluate summaries in terms of *coherence* (i.e., is the summary easy to read and does it follow a natural ordering of facts?), and *grammaticality* (i.e., is the summary fluent and grammatical?). We randomly selected 30 instances from the test set. For Rotten Tomatoes, we filtered out instances where the number of input reviews exceeded 30 so that participants could read the reviews in a timely fashion. We collected three judgments for each comparison. The order of summaries was randomized per participant. A rating per system was computed as the percentage of times it was chosen as best minus the percentage of times it was selected as worst. We refer the readers to Appendix A.1 for the full instructions of the experiment.

We compared summaries produced by the BERTCENT extractive baseline, our model PLANSUM, and two competitive abstractive systems, DENOISESUM (Amplayo and Lapata, 2020) and COPYCAT (Bražinskas et al., 2019). We also included human-authored summaries as an upper bound. The ratings are reported in Table 4.4. Overall,

	Model	Asp	Sen	Coh	Gam
Yelp	BERTCENT	−9.0	−1.5	−2.9	−7.4
	DENOISESUM	−11.3	−11.1	−6.5	−10.6
	COPYCAT	−5.8	−15.0	−15.8	−10.0
	PLANSUM	**3.9**	**6.9**	**5.7**	**7.0**
	GOLD	22.2	20.7	19.4	20.9

	Model	Asp	Sen	Coh	Gam
Rotten Tomatoes	BERTCENT	−8.4	−12.2	−6.9	−4.0*
	DENOISESUM	−31.1	−6.9*	−25.1	−17.3
	COPYCAT	—	—	—	−10.0
	PLANSUM	**10.7**	**1.3**	**2.2**	**−2.2**
	GOLD	28.9	20.4	29.8	23.6

	Model	Asp	Sen	Coh	Gam
Amazon	BERTCENT	−10.7	**−3.1***	−7.1	−9.1*
	DENOISESUM	—	—	—	—
	COPYCAT	−9.8	−18.9	−10.2	−12.22
	PLANSUM	**0.0**	−6.4	**7.1**	**−1.8**
	GOLD	20.4	28.4	10.2	23.1

Table 4.4: Best-worst scaling: aspect- and sentiment-based informativeness (Asp and Sen), coherence (Coh), grammaticality (Gram). All pairwise differences between PLANSUM and other systems are significant, except when there is an asterisk (*), using a one-way ANOVA with posthoc Tukey HSD tests ($p < 0.05$).

the gold summaries were consistently rated the highest on all criteria. Among the system summaries, PLANSUM was rated the best in terms of all criteria, except on sentiment-based informativeness for Amazon, where BERTCENT was given the highest rating. Overall and across three datasets, BERTCENT surprisingly was rated higher than the other abstractive systems. We inspected the summaries produced by these systems and found that COPYCAT summaries are more positive-oriented and DENOISESUM summaries contain more grammatical errors, as also reflected in the ratings. We posit that these errors are possibly due to the use of random sampling and noising functions, respectively, when creating the synthetic dataset.

4.5 Summary

In this chapter, we considered the use of aspect and sentiment distributions as a content plan for unsupervised opinion summarization. With our content plans, we introduced a method to control the variance of opinions in reviews which results into synthetic datasets that resemble real-world data. We also designed a summarization model that is trained using the synthetic dataset, as well as incorporating content planning leading to higher quality summaries. Extensive automatic and human-based evaluation showed that our model outperforms competitive systems on three benchmarks with varying characteristics. Finally, we confirmed that our model produces summaries with opinions that reach consensus.

Consensus summaries may help in giving consumers a general overview about the product or service. However, they may have different needs and preferences, and thus might view products differently. For example, a person might only care about the location of the hotel, while someone else might care more about the breakfast served in the hotel. Theoretically, content plans can be modified to introduce aspect control in the summary generation. However, they are in the form of a probability distribution and are therefore not discrete and cannot be used to control the summary. In the next chapter, we will look at a new task called aspect-controllable opinion summarization and propose a method that generates opinion summaries that can be personalized according to user needs.

Gold Summary
If you're looking for a comfortable and inviting bar this is a great place to go. They have a lot of unique beers on tap that you will not find anywhere else. The staff here is extremely friendly, and after just a couple of minutes it feels like you are chatting with an old friend. The next time you want to head out for some drinks give them a shot!
PLANSUM
This is a great place to hang out with friends. The staff is very friendly and helpful. They have a lot of different beers to choose from and the beer selection is great. I'm not a big fan of beers but this place has some good selections. If you're in the mood for a beer and a fun atmosphere, this will be the place for you.
PLANSUM with Random Sampling
This is a great place to hang out with friends and family. The beer selection is great, and the atmosphere is very nice. I've been here a few times and have never had a bad experience. It's a fun place for a group of friends or groups.
PLANSUM with Similarity Sampling
This is *a great place to go if you're in the area.* It's a cool place for a night out, but it is well worth it. The atmosphere is great and the staff is always friendly. I'm not sure if I will go back.
PLANSUM with no planning
This is a great place to hang out with friends. The staff is very friendly and the beer selection is great. I've had a couple of beers and they have a good selection of beer and beer. *It's a little pricey but it is worth the wait.*

Table 4.5: Yelp summaries generated by PLANSUM and variants thereof. Aspects also mentioned in the gold summary are in color (atmosphere, staff, and beer), all other aspects are *italicized*.

Gold Summary
Alias Betty works both as a gripping thriller and as a precisely drawn character study.
PLANSUM
The film is a powerfully constructed thriller that is hypnotic, disturbing, unsettling, and darkly funny.
PLANSUM with Random Sampling
The film's episodic ters is a film that hurtss the viewer with the simplicity of the bourgeois and the city.
PLANSUM with Similarity Sampling
It's not a perfect film, but it is a film that raises a lot of ground and redemption.
PLANSUM with no planning
The film's lasting impression is expressed, but it is a movie that'll stay with you afterward. it has a lot of thoughts.

Table 4.6: Rotten Tomatoes summaries generated by PLANSUM and variants thereof.

Gold Summary
This fm/am radio, iPhone docking station and alarm clock is a perfect combination! The sound is amazing, the alarm clock is not annoying, and the design looks great! It would be nice to have a place to use an aux cord and certain apple products require a $5 adapter to use the docking station but other than that, this product is fantastic!
PLANSUM
This is a great little radio *for the price*. *It is easy to use* and the sound quality is great. The only thing I don't like is that it's not really a dock, since it does not have a cord. I would recommend this to anyone who wants to listen to music.
PLANSUM with Random Sampling
This is a great product. *It is very easy to use*, and the sound quality is great. the only complaint i have is that the alarm clock isn't very loud. i would not recommend this product to anyone.
PLANSUM with Similarity Sampling
I bought this for my wife's iPod nano. *it is very easy to set up*, and the sound quality is great. The only drawback is that it doesn't have a lot of features to charge it. I would recommend this product to anyone.
PLANSUM with no planning
This is a great product. *It is easy to use and works great with my iphone 4s*. the only problem i have is that it's a little bulky, but i'm not sure if it would have been a problem. i would recommend this player to anyone who is looking for a docking station.

Table 4.7: Amazon summaries generated by PLANSUM and variants thereof. Aspects also mentioned in the gold summary are in color (sound quality, accessories, and design), all other aspects are *italicized*.

Chapter 5

Aspect-Controllable Opinion Summarization

The previous chapters focused on the task of producing opinion summaries under the assumption that opinions are considered salient if they are *popular* or *redundant* across reviews, following previous formulations of the opinion summarization task (Angelidis and Lapata, 2018b; Suhara et al., 2020). However, in this chapter we argue that the notion of salience in reviews largely depends on *user interest*. For example, one might only care about the connectivity of a television product, an aspect which might be unpopular amongst reviews. As a result models that create *general* opinion summaries may not satisfy the needs of all users, limiting their ability to make decisions. Angelidis et al. (2021) mitigate this problem with an extractive approach that produces both general and *aspect-specific* opinion summaries. They achieve this essentially by clustering opinions through a discrete latent variable model (van den Oord et al., 2017) and extracting sentences based on popular aspects or a particular aspect. By virtue of being extractive, their summaries can be incoherent, and verbose containing unnecessary redundancy. And although their model creates summaries for individual aspects, it is not clear how to control the number of aspects in the output (e.g., to obtain summaries that mention multiple rather than a single aspect of an entity) without modifications.

In this chapter, we propose an abstractive opinion summarization model that generates aspect-controllable summaries. Using a corpus of reviews on entities (e.g., hotels, television sets), we construct a synthetic training dataset consisting of reviews, a pseudo-summary, and three types of *aspect controllers* which reflect different levels of granularity: aspect-related keywords, review sentences, and document-level aspect codes. We induce aspect controllers automatically based on a multiple instance learn-

General

The room was clean and comfortable. The staff was very friendly and helpful. It was a great location, just a short walk to the beach. There wasn't much to do in the area, but the food was good.

Location

The location was great, right on the Boardwalk, and close to the Venice beach.

Rooms

The room was very clean and the bathroom was very nice. The bathroom had a large separate shower. There was a TV in the room.

Location and **Rooms**

The location is great, right on Boardwalk, and the beach is very nice. The room was very clean and the bathroom was very nice and the shower was great.

Cleanliness, **Location**, **Room**, and **Service**

The staff was very friendly and helpful. The room was very clean, and the bathroom was very nice. It was a great location, right on the beach.

Figure 5.1: General and aspect-specific summaries generated by our model for a hotel from the SPACE dataset. Aspects and aspect-specific sentences are color-coded.

ing model (Keeler and Rumelhart, 1991) and very little human involvement. Using the aspect-enriched dataset, we then fine-tune a pretrained model (Raffel et al., 2020) on summary generation. By modifying the controllers, we can flexibly generate general and aspect-specific summaries, discussing one or more aspects. Figure 5.1 shows summaries generated by our model.

We perform experiments on SPACE (Angelidis et al., 2021), a single domain dataset consisting of hotel reviews, and OPOSUM (Angelidis and Lapata, 2018b), a dataset with product reviews from multiple domains (e.g., "laptop bags", "boots"). Automatic and human evaluation shows that our model outperforms previous approaches on both tasks of general and aspect-specific summarization. We also demonstrate that it can effectively generate multi-aspect summaries based on user preferences.

5.1 Related Work

Our work in this chapter is closest to Angelidis et al. (2021) who propose an extractive summarization model that uses a vector-quantized variational autoencoder (van den

Oord et al., 2017) to learn aspect-specific review representations. It is trained to reconstruct sentences in reviews, which consequently learns sentence encodings and a *latent codebook* that represents aspects. Effectively, their model groups opinion sentences into clusters and extracts those capturing aspect-relevant information. On the other hand, our work employs multi-instance learning to identify aspect-bearing elements in reviews with varying degrees of granularity (e.g., words, sentences, documents) which we argue affords greater flexibility and better control of the output summaries. In doing so, we also introduce an effective method to create synthetic datasets for aspect-guided opinion summarization.

Our work also relates to approaches which attempt to control summarization output based on different characteristics of text (Liu et al., 2018b; Cao and Wang, 2021). Kikuchi et al. (2016) control the length of the summary using four different decoding strategies: beam search with end-of-summary tags, discarding out-of-range sequences, introducing length embeddings as additional input, and length-based LSTM memory cell initialization. Fan et al. (2018) improves the summary quality by automatically controlling attributes such as length and style, and customizing summaries based on user preferences such as entities and portions of the input they might be interested in. These customizations are all done by introducing control tokens and prepending them into the input. Finally, query-focused summarization (Dang, 2006; Xu and Lapata, 2020, 2021) is another related task which uses queries to control the output and focus on generating summaries that answer the given query. Although we focus solely on aspect, our method is general and could be used to adjust additional properties of a summary such as sentiment (e.g., positive vs. negative) or style (e.g., formal vs. colloquial).

A key machine learning framework that we use in our approach is multiple instance learning (MIL; Keeler and Rumelhart, 1991). MIL is a machine learning framework where labels are associated with groups of instances (i.e., *bags*), while instance labels are unobserved. The goal is then to infer labels for multiple instances, given only the observable bag labels (Dietterich et al., 1997; Maron and Ratan, 1998), or to jointly infer labels for both instances and bags (Zhou et al., 2009; Wei et al., 2014; Kotzias et al., 2015). In NLP, there are three ways to represent bags and instances. Firstly, a bag can be a document, where its sentences are considered as instances (Andrews et al., 2002; Ray and Craven, 2005; Angelidis and Lapata, 2018a). Secondly, a bag can also be a sentence with instances in the form of tokens (Surdeanu et al., 2012). Finally, the first two representations can be combined in a hierarchical manner, where documents

are bags of sentences and sentences are bags of tokens (Kotzias et al., 2014; Xu and Lapata, 2019). Our approach follows the hierarchical representation to infer aspect controllers of multiple levels of granularities (i.e., word-, sentence-, and documen-level).

5.2 Problem Formulation

Let C denote a corpus of reviews about entities (e.g., products, hotels). Let $R_e = \{r_1, r_2, ..., r_N\}$ denote a set of reviews for entity e and $A_e = \{a_1, a_2, ..., a_M\}$ a set of aspects that are relevant for the entity (e.g., cleanliness and location of a hotel). Each review r_i is a sequence of tokens $\{w_1, w_2, ...\}$, while each aspect a_j is represented by a small set of *seed words* $\{v_1, v_2, ...\}$ (e.g., *spotless*, *dirty*, *stain*). These seed words can be acquired automatically (Angelidis and Lapata, 2018b) or provided by users.

Our approach creates two types of summaries: (a) a general summary that contains salient opinions about *all* aspects of an entity, and (b) an aspect-specific summary that focuses on opinions about *particular* aspects of interest specified by a query $Q = \{q_1, q_2, ..., q_M\}$; here, q_j is an indicator function which designates whether the aspect should be mentioned in the summary. We emphasize that the query can represent more than one aspect to reflect real-world usage. To facilitate supervised training, we create a synthetic training dataset $D = (R, z, y)$, which is a set of triples composed of input reviews R, a pseudo-summary y, and aspect controllers z (Section 5.2.2). Our aspect controllers are induced with a unified model based on multi-instance learning (Section 5.2.1) and correspond to different levels of granularity: (1) document-level aspect codes, (2) aspect-related review sentences, and (3) aspect keywords.

At training time, we fine-tune a pretrained sequence-to-sequence Transformer model (Raffel et al., 2020) using controllers z as input and a pseudo-summary as output. During inference, we modulate summary generation by modifying the controllers, e.g., we produce a general summary using all aspect codes, or an aspect-specific one based on a subset thereof (Section 5.2.3).

5.2.1 Controller Induction Model

A key feature of our approach is the set of aspect controllers which allow our summarization model to be controllable. We induce these controllers using a multiple instance learning (MIL) model, illustrated in Figure 5.2. In our setting, documents

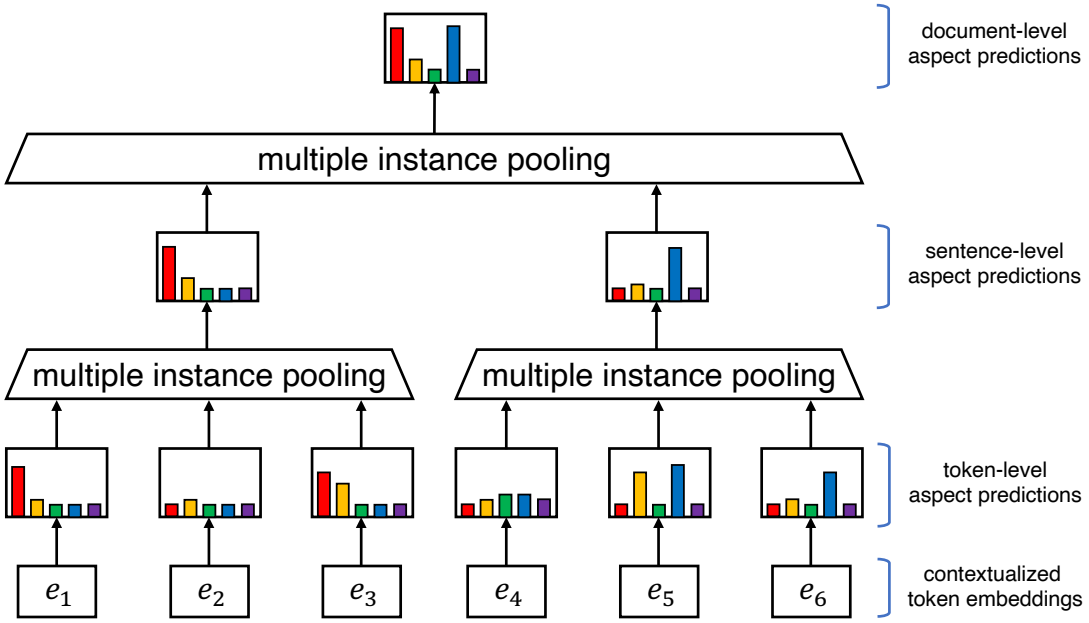


Figure 5.2: Overview of the controller induction model. Token-level aspect predictions are aggregated into sentence-level predictions using a multiple instance pooling mechanism (illustrated in Figure 5.3). The process is repeated from sentence- to document-level predictions. The colors represent different aspects.

are bags of sentences and sentences are bags of tokens. We further assume that only documents have aspect labels, which are normally not available but can be inferred using the procedure described later. Given review r with tokens $\{w_k\}$, we obtain token encodings $\mathbf{E} = \{\mathbf{e}_k\}$ from a pretrained language model (PLM; Liu et al., 2019) which uses the popular Transformer architecture (Vaswani et al., 2017). We use a non-linear transformation to obtain token-level aspect predictions $\mathbf{z}_{\mathcal{T}}$:

$$\mathbf{E} = \text{PLM}(\{w_k\}) \tag{5.1}$$

$$\mathbf{z}_{\mathcal{T}} = \tanh(\mathbf{W}\mathbf{E} + \mathbf{b}) \tag{5.2}$$

where $\mathbf{z}_{\mathcal{T}} \in [-1,1]^{N \times M}$, and N and M are the number of tokens and aspects, respectively. A positive value denotes that the token is related to the aspect of interest (and otherwise unrelated).

Multiple Instance Pooling To obtain sentence-level aspect predictions $\mathbf{z}_{\mathcal{S}}$, we aggregate token-level predictions $\mathbf{z}_{\mathcal{T}}$ using a new pooling method particularly effective for our multi-instance learning setting, illustrated in Figure 5.3. We first obtain multiple

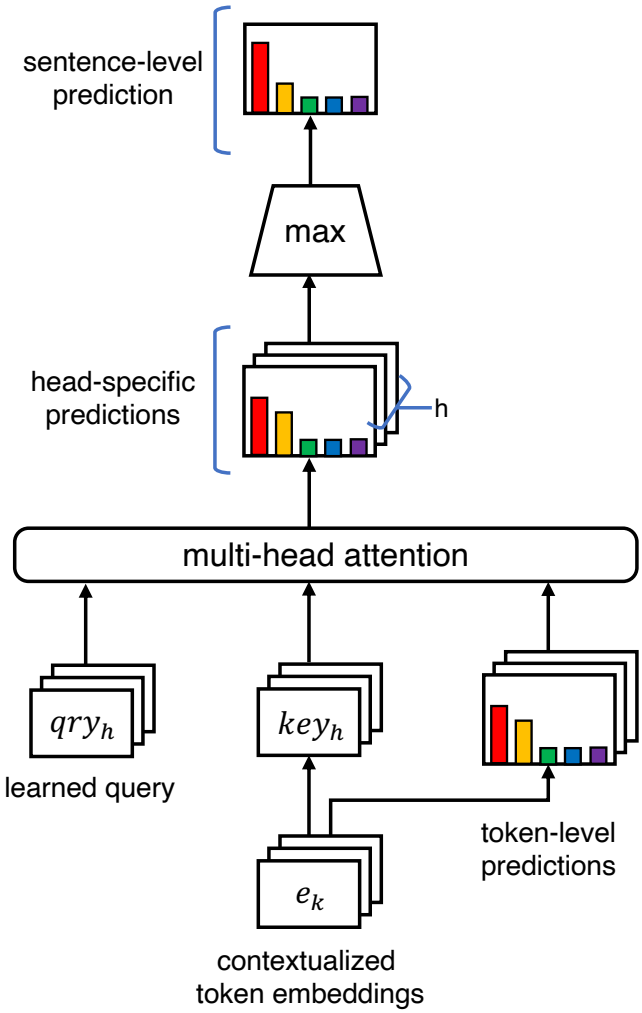


Figure 5.3: Multiple Instance Pooling of token-level predictions. Token embeddings $\{\mathbf{e}_k\}$ are transformed into head-specific keys $\{\mathbf{key}_h\}$, which are then used to weight-sum token-level predictions to obtain head-specific predicitons. Head-specific predictions are finally max-pooled into sentence-level prediction. The same process is done for pooling sentence-level predictions. The colors represent different aspects.

predictions $\mathbf{z}_h$ for each attention head h:

$$\mathbf{z}_h = \sum_k (\mathbf{z}_{\mathcal{T}} * a_h[k]) \tag{5.3}$$

$$\alpha_h = \text{softmax}(\mathbf{key}_h \cdot \mathbf{qry}_h) \tag{5.4}$$

where $*$ is element-wise multiplication, $\cdot$ is dot product, k is the token index, $\mathbf{qry}_h$ is a head-specific query vector, and $\mathbf{key}_h$ is defined below:

$$\mathbf{key}_h = \tanh(\mathbf{W}_h\mathbf{E} + \mathbf{b}_h) \tag{5.5}$$

We hypothesize that different attention heads represent different aspects of the semantic space, and are thus helpful at predicting multiple aspects. We obtain a sentence-level prediction by max pooling the predictions of individual heads:

$$\mathbf{z}_S = \text{max-pool}(\{\mathbf{z}_h\}) \tag{5.6}$$

We use max pooling since we want to isolate the most pertinent aspects for a given sentence; standard pooling methods such as mean and attention pooling (Angelidis and Lapata, 2018a; Xu and Lapata, 2019) assume that *all* instances of a bag contribute to its label. In Figure 5.2 (right) we illustrate our pooling mechanism and empirically show in experiments (see Section 5.4.1) it is superior to alternatives.

We so far discussed how multiple instance pooling is applied at the token-level to obtain sentence-level predictions $\mathbf{z}_S$. Analogously, multiple instance pooling is applied to sentences to obtain document-level predictions $\mathbf{z}_\mathcal{D}$ (see Figure 5.2).

Training and Inference Training the multiple instance model just described requires a dataset consisting of (review, aspect label) pairs. Unfortunately, we do not have access to annotations denoting which aspects are discussed in each review. Recall, however, that aspects are represented by seed words $\{v_1, v_2, ...\}$, which we exploit to induce silver-standard labels. Specifically, for each review in the dataset, we obtain binary labels $\hat{\mathbf{z}}_\mathcal{D}$ where $\hat{\mathbf{z}}_\mathcal{D}[a] = 1$ if at least one seed word for aspect a is found in the review (and -1 otherwise).

We train the model using a soft margin loss, summing over all aspects $a \in A$:

$$\mathcal{L}_{ctrl} = \sum_a \log(1 + \exp(-\mathbf{z}_\mathcal{D}[a] * \hat{\mathbf{z}}_\mathcal{D}[a])) \tag{5.7}$$

The parameters of the pretrained language model (see Equation (5.2)) are frozen, i.e., they are not fine-tuned during training which makes our controller induction model lightweight and efficient. It is important to have as little parameters as possible since in cases where there are multiple different domains with different aspects, it is necessary to train separate controller induction models, one for each domain. Having fine-tuned parameters for each domain can be very expensive.

5.2.2 Synthetic Dataset Creation

The MIL model allows us to learn three kinds of aspect controllers which are subsequently used to create a synthetic dataset for training our summarizer. These are *aspect*

Summary y
At first they took us to an unready room which was disappointing but after a short wait they took us to a really big room with a great harbor scene as an apology to the mess. The rooms are pretty new or renovated recently. The breakfast buffet wasn't included in the price. but we enjoyed continental breakfast.
Review r_1
Check in was quick and our bags were brought to the room in a timely manner. The rooms and hallways left a little more to be desired. The rooms didnt look nearly as good as they did in other less known cities. No safe or frig in the rooms. The staff was great.
Review r_2
Only option for a hot meal for breakfast was scrambled eggs and bacon; the toaster was broken as well, with burned out elements. Other food in the lounge was good (fruit, coffee). Recommendation: eat elsewhere; even room service would probably have been better.

Figure 5.4: Pseudo-summary y and input reviews R; the aspect codes for summary y are room and food. Review sentences with the same aspect are underlined and same aspect-keywords are magnified.

codes, essentially document-level aspect predictions $\mathbf{z}_D$, which control the overall aspect of the summary, *aspect keywords* ensure content support by explicitly highlighting which tokens from the input should appear in the summary, and *aspect-relevant sentences* which provide textual context for summary generation (while non-aspect-related sentences are ignored).

We first sample review r_i as a pseudo-summary from review set R_e of entity e. We treat r_i as a pseudo-summary provided it resembles a real summary. We assume that opinion summaries discuss specific aspects regarding entity e. We use our controller induction model to verify this, i.e., document-level aspect predictions $\mathbf{z}_D$ for r_i should be positive for at least one aspect. Provided r_i fulfills this constraint, we use it as summary y and $R_e - \{r_i\}$ as review set R. A simplified example is shown in Figure 5.4, the pseudo summary is highlighted in gray and the input reviews in cyan. The summary focuses on the room and food aspects of a hotel and these are its aspect codes (shown in blue and red).

Let (R, y) denote review set R for summary y (we only show two reviews in Fig-

ure 5.4 but there are usually hundreds). We obtain (positive) document-level aspect predictions $\mathbf{z}_D^{(y)}$ for summary y and sentence-level aspect predictions $\mathbf{z}_S^{(x)}$ for all reviews $x \in X$. We then rank review sentences in X based on their similarity to the summary's overall aspect. Specifically, we compare predictions $\mathbf{z}_S^{(x)}$ with $\mathbf{z}_D^{(y)}$ using the soft margin loss function from Equation (5.7). We also compare token-level predictions $\mathbf{z}_T^{(x)}$ with $\mathbf{z}_D^{(y)}$ using the same function to induce aspect keywords. In Figure 5.4 sentences which discuss the same aspect as the summary are underlined, and same-aspect keywords are magnified. For illustration purposes we only show two aspect codes in Figure 5.4, but these can be one or several, and different review sentences and keywords would be selected for different aspects.

5.2.3 Opinion Summarization Model

We use a pretrained sequence-to-sequence Transformer model (Raffel et al., 2020) to generate opinion summaries. We transform the aspect controllers z into the following format:

> `[CODE]` `[ASPECT`$_2$`]` `[ASPECT`$_3$`]` `[KEY]` keyword$_1$ keyword$_2$ ... `[SNT]` this is the first sentence `[SNT]` this is the second sentence ...

where `[CODE]`, `[KEY]`, and `[SNT]` are indicators denoting that the next tokens correspond to aspect codes, keywords, and review sentences.

Instead of the full set of input reviews R, the encoder takes z as input and produces multi-layer encodings $\mathbf{Z}$. The decoder then outputs a token distribution $p(y_t)$ for each time step t, conditioned on both $\mathbf{Z}$ and $y_{1:t-1}$ through attention:

$$\mathbf{Z} = \text{Encoder}(z) \tag{5.8}$$

$$p(y_t) = \text{Decoder}(y_{1:t-1}, \mathbf{Z}) \tag{5.9}$$

We fine-tune the model using a maximum likelihood loss to optimize the probability distribution $p(y)$ based on gold summary $\hat{y}$:

$$\mathcal{L}_{gen} = -\sum_t \hat{y}_t \log p(y_t) \tag{5.10}$$

During inference, we can generate different kinds of opinion summaries by modifying the aspect controllers. When creating a general summary, we use all aspect codes as input. Analogously, when generating a single aspect summary, we use one aspect code. The aspect codes guide the selection of keywords and sentences from the input

Aspect controller for a **general summary**:
[CODE] [BUILDING] [CLEANLINESS] [FOOD] [LOCATION] [ROOMS] [SERVICE] [KEY] breakfast clean room location staff walk helpful friendly service walking [SNT] for a large city like rome, to be in such a good location with a clean and serviceable room, hotel navona is the ticket. [SNT] location is very good, rooms are clean and spacious, breakfast can at least get you going and the staff, well, just remember you are not there to socialize. ...

Aspect controller for a **single aspect summary**:
[CODE] [ROOMS] [KEY] room bathroom bed shower bath roomy bedroom cupboard mould comfortable [SNT] once you recover from the front desk being on the first floor and having to drag your bags up yourself our room was lovely. [SNT] my teenage son and i were very pleased with the comfort of our room (and easy access to wifi), but mostly we were impressed with the kindness, courtesy and efficiency of the hotel staff who assisted in many details and small matters that made our 7-day stay very positive. ...

Aspect controller for a **multi-aspect summary**:
[CODE] [CLEANLINESS] [FOOD] [KEY] breakfast clean food bread croissant dinner meal dirty breads pastries [SNT] continental breakfast included, is served quickly and has a nice variety. [SNT] the rooms were very clean and quite large for a european hotel. [SNT] breakfast is simple but nice & filling and keeps you going - euro breakfast. ...

Figure 5.5: Example aspect controllers to generate general, single aspect, and multi-aspect opinion summaries.

reviews (see Figure 5.4) which are given as input to our Transformer model to generate the summary (see Equation (5.8)). Figure 5.5 shows examples of aspect controllers for generating general, single aspect, and multi-aspect summaries for hotel reviews.

5.3 Experimental Setup

5.3.1 Datasets

We performed experiments on two opinion summarization datasets covering different review domains. SPACE (Angelidis et al., 2021) is a large corpus of "hotel" reviews

Dataset	SPACE	OPOSUM+
review corpus size	1.14M	4.13M
#domains	1	6
#aspects	6	18
#test examples	50	60
#reviews/example	100	10
#summaries/example	3	3
#general summaries	150	180
#aspect summaries	900	540

Table 5.1: Statistics for SPACE and OPOSUM+ (underlined summaries are extractive). Domains in OPOSUM+ include "Laptop Bags", "Bluetooth Headsets", "Boots", "Keyboards", "Televisions", and "Vacuums", each of which has three aspects (see Table 5.3). Aspects in SPACE include building, cleanliness, food, location, rooms, and service.

from TripAdvisor; it contains human-written abstractive opinion summaries for evaluation only. Each instance in the evaluation set consists of 100 reviews and seven summaries: one general summary and six aspect-specific ones representing the aspects building, cleanliness, food, location, rooms, and service. OPOSUM (Angelidis and Lapata, 2018b) is a large corpus of product reviews from six different domains: "laptop bags", "bluetooth headsets", "boots", "keyboards", "televisions", and "vacuums". It also includes an evaluation set with extractive general summaries.

We extended OPOSUM in two ways. Firstly, we increased the size of the review corpus. The original dataset includes only 359K reviews, which is the result of downsampling the *Amazon Product Dataset* introduced in (McAuley et al., 2015). We instead gathered all reviews tagged with at least one of the OPOSUM domains ("Laptop Bags", "Bluetooth Headsets", "Boots", "Keyboards", "Televisions", and "Vacuums") from the newest version of the *Amazon Product Dataset* compiled in (Ni et al., 2019). Since "Laptop Bags" and "Blueetooth Headsets" were significantly smaller than the other domains, we additionally included all reviews tagged with "Bags" and "Headsets". We were able to increase the dataset to 4.13M reviews, i.e., by a factor of 12.

Secondly, we created a large collection of human-written abstractive summaries for aspect-specific summarization evaluation. For each test product (e.g., television set) and for each aspect (e.g., `image quality`), we asked three annotators to write an opinion summary about the given aspect. The annotators were shown 10 input

Aspect	"Hotels"
building	lobby pool decor gym area
cleanliness	clean spotless garbage dirty stain
food	breakfast food buffet restaurant meal
location	location walk station distance bus
rooms	room bed bathroom shower spacious
service	staff service friendly helpful desk

Table 5.2: SPACE seed words for its six aspects.

reviews, in which opinions about the target aspect were highlighted to aid them in their task. We only used the three most common aspects for each domain (see 5.3 for the list of aspects), since opinions about less common aspects do not appear frequently in reviews. We gathered 540 aspect-specific summaries in total. We call this extended dataset OPOSUM+.

Both datasets include five human-annotated seed words[1] for each aspect (see Tables 5.2 and 5.3). Data statistics are shown in Table 5.1. Using our synthetic dataset creation method, we were able to generate 512K and 341K training instances for SPACE and OPOSUM+, respectively. We refer the readers to Chapter 2 for details and example reviews and summaries of the datasets.

5.3.2 Implementation

For our pretrained Transformer models, we used weights and settings available in the HuggingFace library (Wolf et al., 2020). Specifically, we used `distilroberta-base` (Liu et al., 2019; Sanh et al., 2019) as our language model and `t5-small` (Raffel et al., 2020) as our sequence-to-sequence model. We trained the controller induction model with a learning rate of $1e-4$ for 100K steps, using $h = 12$ heads. For OPOSUM+, we trained separate controller induction models for different domains. For the aspect controllers, we selected 10-best keywords, and review sentences were truncated up to 500 tokens to fit in the pretrained model. For summarization, we used a learning rate of $1e-6$ and 500K training steps. We used Adam with weight decay (Loshchilov and Hutter, 2019) to optimize both models. We added a linear learning rate warm-up for

[1] To simplify annotation, seed words are selected by human annotators from a larger list of seed words automatically produced using the normalized TF-IDF approach introduced in Angelidis and Lapata (2018b). We report comparisons when using automated and human-generated seed words in the later sections (see Table 5.8).

the first 10K steps. We generate summaries with beam search of size 2 and refrain from repeating ngrams of size 3 (Paulus et al., 2018).

Aspect	"Laptop Bags"
looks	looks color stylish looked pretty
quality	quality material poor broke durable
size	fit fits size big space

Aspect	"Bluetooth Headsets"
comfort	ear fit comfortable fits buds
ease of use	easy button simple setup control
sound quality	sound quality hear noise volume

Aspect	"Boots"
comfort	comfortable foot hurt ankle comfy
looks	cute look looked fringe style
size	size half big little bigger

Aspect	"Keyboards"
build quality	working months build stopped quality
feel/comfort	feel comfortable feels mushy shallow
layout	key keys delete backspace size

Aspect	"Televisions"
connectivity	hdmi computer port usb internet
image quality	picture color colors bright clear
sound quality	sound speakers loud tinny bass

Aspect	"Vacuums"
accessories	filter brush attachments attachment turbo
ease of use	easy push corners awkward impossible
suction power	suction powerful power hair quiet

Table 5.3: OPOSUM+ seed words for various "domains" and their aspects.

5.4 Results

We compared our **A**spect **C**ontroll**ed** **Sum**marization (ACESUM) model with several extractive and abstractive approaches. Extractive systems include

1. CENTROID (Radev et al., 2004), an extractive method that selects as a summary the review closest to the centroid. Following results in the previous chapter, we used average token encodings from BERT (Devlin et al., 2019) to represent reviews (see Section 3.3.3 for details).

2. LEXRANK (Erkan and Radev, 2004), a PageRank-like algorithm that selects the most salient sentences from the input. We also used BERT encodings to represent sentences for this method (see Section 3.3.3 for details).

3. QT (Angelidis et al., 2021), a neural clustering method that uses Vector-Quantized Variational Autoencoders (van den Oord et al., 2017) to represent opinions in quantized space. QT is trained to reconstruct sentences in reviews, which consequently learns sentence encodings and a *latent codebook* that represents aspects. At test time, the sentences that are nearest to the latent codes are included in the extractive summary (see Section 5.1 for details).

4. ACESUMEXT, an extractive version of our model. It uses only aspect sentences ranked by our controller induction model (and discards aspect keywords and codes). We use the highly ranked sentences (truncated up to 500 tokens) as input to LexRank to produce the extractive summary.

Abstractive systems include:

5. MEANSUM (Chu and Liu, 2019), an autoencoder that generates summaries by reconstructing the mean of review encodings (see Section 3.3.3 for details).

6. COPYCAT (Bražinskas et al., 2019), a hierarchical variational autoencoder which learns a latent code of the summary (see Section 3.3.3 for details).

7. Two T5 baselines that use different methods to create synthetic datasets, namely (7.1) T5-RANDOM, which naively uses random sampling to select the pseudo-summary, as in Bražinskas et al. (2019), and (7.2) T5-SIMILAR, which treats the review that is the most similar with all the other reviews in terms of IDF-weighted ROUGE-1 as the pseudo-summary, as in document-level noising function in Chapter 3.

Finally, we compared against two upper bounds: an extractive ORACLE which selects as a summary the review with the best ROUGE score against the input, and a HUMAN upper bound, calculated as inter-annotator ROUGE.

5.4.1 Automatic Evaluation

We evaluated the quality of general and aspect-specific opinion summaries using F_1 ROUGE (Lin and Hovy, 2003). Unigram and bigram overlap (ROUGE-1/2) are proxies for assessing informativeness while the longest common subsequence (ROUGE-L) measures fluency.

General Opinion Summarization Table 5.4 reports results on general opinion summarization. As can be seen, ACESUM outperforms all competing models on SPACE and performs best among abstractive systems on OPOSUM+. Our extractive model, ACESUMEXT, is overall best on OPOSUM+. This is expected since general OPOSUM+ summaries are extractive. Amongst abstractive models, Transformer-based models outperform MEANSUM and COPYCAT, demonstrating that pretraining is helpful for opinion summarization.

Aspect-Specific Opinion Summarization Most comparison systems (all except QT) cannot naturally generate aspect-specific summaries. We use a simple sentence-filtering method to remove non-aspect-related sentences from the input during inference. Specifically, we use BERT encodings (Devlin et al., 2019) to represent tokens in review sentences $\{r_i^{(bert)}\}$ and aspect seeds $\{a_j^{(bert)}\}$. We then rank the review sentences based on the maximum similarity between seed and sentence tokens, calculated as $\max_{i,j}(\text{sim}(r_i^{(bert)}, a_j^{(bert)}))$, where $\text{sim}(a,b)$ is the cosine similarity function. This method cannot be ported to the CENTROID and ORACLE baselines, and thus we do not compare with them.

Our results are summarized in Table 5.5. Note that SPACE and OPOSUM+ focus exclusively on *single* aspect summaries. We assess our model's ability to generate summaries covering multiple aspects in the following section. Overall, ACESUM performs best across datasets and metrics, which shows that our controllers can effectively customize summaries based on aspect queries. Interestingly, amongst extractive models, ACESUMEXT performs best. This suggests that, a simple centrality-based extractive approach such as LexRank (Erkan and Radev, 2004) can produce good enough summaries as long as an effective sentence filtering method is applied beforehand (in our

	Model	ROUGE-1	ROUGE-2	ROUGE-L
SPACE	CENTROID	31.29	4.91	16.43
	LEXRANK	31.41	5.05	18.12
	QT	38.66	10.22	21.90
	ACESUMEXT	35.50	7.82	20.09
	MEANSUM	34.95	7.49	19.92
	COPYCAT	36.66	8.87	20.90
	T5-RANDOM	37.65	10.62	22.82
	T5-SIMILAR	<u>38.84</u>	<u>10.82</u>	<u>22.74</u>
	ACESUM	**40.37***	**11.51***	**23.23**
	ORACLE	40.23	13.96	23.46
	HUMAN	49.80	18.80	29.19

	Model	ROUGE-1	ROUGE-2	ROUGE-L
OPOSUM+	CENTROID	33.44	11.00	20.54
	LEXRANK	35.42	10.22	20.92
	QT	<u>37.72</u>	<u>14.65</u>	<u>21.69</u>
	ACESUMEXT	**38.48***	**15.17***	**22.82***
	MEANSUM	26.25	4.62	16.49
	COPYCAT	27.98	5.79	17.07
	T5-RANDOM	29.88	5.64	17.19
	T5-SIMILAR	30.42	6.07	17.17
	ACESUM	32.98	10.72	20.27
	ORACLE	41.88	21.52	29.30
	HUMAN	55.42	37.26	44.85

Table 5.4: Automatic evaluation for *general summarization*. Extractive/Abstractive/Upper-bound models are shown in first/second/third block. Best systems are boldfaced while the second best systems are underlined. An asterisk (*) means there is a significant difference between best and second best systems (based on paired bootstrap resampling; $p < 0.05$).

case this is based on the controller induction model). T5 models perform substantially worse on this task, indicating that synthetic datasets based on either random or similarity-based sampling techniques are not suited to aspect-specific opinion summa-

	Model	ROUGE-1	ROUGE-2	ROUGE-L
SPACE	LEXRANK	24.61	3.41	18.03
	QT	29.43	8.45	22.37
	ACESUMEXT	30.91	8.77	23.61
	MEANSUM	25.68	4.61	18.44
	COPYCAT	27.19	5.63	19.18
	T5-RANDOM	21.40	4.83	15.45
	T5-SIMILAR	22.69	5.12	16.44
	ACESUM	**32.41***	**9.47***	**25.46***
	HUMAN	44.86	18.45	34.58

	Model	ROUGE-1	ROUGE-2	ROUGE-L
OPOSUM+	LEXRANK	22.51	3.35	17.27
	QT	23.99	4.36	16.61
	ACESUMEXT	26.16	5.75	18.55
	MEANSUM	24.63	3.47	17.53
	COPYCAT	26.17	4.30	18.20
	T5-RANDOM	24.47	4.20	16.18
	T5-SIMILAR	23.86	4.30	16.36
	ACESUM	**29.53***	**6.79***	**21.06***
	HUMAN	43.03	16.16	31.53

Table 5.5: Automatic evaluation for *aspect-specific summarization*. Extractive/Abstractive/Upper-bound models are shown in first/second/third block. Best systems are boldfaced while the second best systems are underlined. An asterisk (*) means there is a significant difference between best and second best systems (based on paired bootstrap resampling; $p < 0.05$).

rization.

Ablation Studies We present various ablation studies on the controller induction model and the summarization model itself. In Table 5.6, we compare our multiple instance pooling (MIP) mechanism with three standard pooling methods: mean, max, and attention-based pooling. We evaluate models using document and sentence F_1 which measures the quality of document- and sentence-level aspect predictions. We

	SPACE		OPOSUM+	
Model	Document F_1	Sentence F_1	Document F_1	Sentence F_1
MIP (ours)	**77.35**	**40.85**	**83.28**	**50.48**
Max	63.35	35.12	66.52	44.00
Attention	31.77	29.30	34.00	35.80
Mean	27.38	27.87	30.38	34.35

Table 5.6: Performance of controller induction models (document- and sentence-level); comparison of multiple instance pooling (MIP) against max, mean, and attention pooling. Bold-faced values are the best for each column, and are significantly different when compared to the second best values which are underlined (based on paired bootstrap resampling; $p < 0.05$).

extrapolate aspect labels for documents and sentences from the development sets of SPACE and OPOSUM+ which contain aspect-specific summaries. That is, we assume the target aspect of the summary is the document label and that all sentences within the summary are also representative of the same aspect. Results show that attention and mean pooling are not suitable for multi-instance learning, underperforming especially on document-level F_1. This suggests that token-level predictions are not used effectively to predict higher level aspects. Our results confirm that using multiple experts (i.e., attention heads) yields better aspect predictions.

In Table 5.7, we evaluate the contribution of different aspect controllers to summarization output. Removing a certain aspect controller (aspect-related keywords, sentences, and codes) decreases the performance of the model. Among the three kinds of controllers, the aspect-related keywords contribute the least to the performance increase of the model in generating aspect-specific summaries. However, removing the keywords decreases the model performance on general summarization. Moreover, selecting sentences randomly rather than based on aspect hurts performance. This is especially true when generating aspect-specific summaries, where we see at least 5.87 decrease in ROUGE-L on both datasets. We also find that aspect codes substantially increase model performance in OPOSUM+. Without them, both general and aspect-specific summarization performance decrease by 2.42 and 2.62 ROUGE-L, respectively. We conjecture that this is due to OPOSUM+ having multiple domains and, consequently, more aspects compared to SPACE.

Finally, we evaluate the performance of ACESUM when using a different set of

	SPACE		OPOSUM+	
Model	General	Aspect	General	Aspect
ACESUM	23.23	25.03	19.64	20.16
No keywords	21.88	24.82	18.97	19.97
Random sentences	22.42	19.16	18.96	13.44
No aspect code	22.29	24.99	17.22	17.54

Table 5.7: Variants of ACESUM with different aspect controllers. Results are shown using ROUGE-L for general and aspect-specific opinion summaries.

seed words to generate aspect-specific summaries. In their work, Angelidis and Lapata (2018b) proposed to use seed words which are *automatically* obtained using a small number of aspect-annotated reviews. They ranked words using a scoring function based on their normalized TF-IDF (i.e., clarity scoring function in information retrieval; Cronen-Townsend et al., 2002), where high-scoring words are considered important. The same method was used to generate seed words for QT in Angelidis et al. (2021). The automatic seed words for SPACE and OPOSUM+ are shown in Appendix C. In Table 5.8, we compared ACESUM and QT using automatically generated seed words instead of human-annotated ones as used in Section 5.4.1. As can be seen, regardless of the seed words used by the models, ACESUM consistently outperforms QT on both datasets. Moreover, the use of automatic seed words reduces the performance of both models, indicating that tiny human intervention can lead to better written summaries.

5.4.2 Human Evaluation

We conducted several human elicitation studies to further analyze the summaries produced by competing systems using the Amazon Mechanical Turk crowdsourcing platform.

Best-Worst Scaling The first study assessed the quality of general opinion summaries using Best-Worst Scaling (BWS; Louviere et al., 2015). Participants were shown a human-written summary, in relation to which they were asked to select the best and worst among system summaries, taking into account the following criteria: *Informativeness* (how consistent are the opinions with the reference?), *Coherence* (is the summary easy to read and well-organized?), *Conciseness* (does the summary provide

	Model	ROUGE-1	ROUGE-2	ROUGE-L
SPACE	*using automatic seed words*			
	QT	28.95	8.34	21.77
	ACESUM	30.78	8.39	23.82
	using human seed words			
	QT	29.43	8.45	22.37
	ACESUM	32.41	9.47	25.46

	Model	ROUGE-1	ROUGE-2	ROUGE-L
OPOSUM+	*using automatic seed words*			
	QT	23.16	4.13	16.81
	ACESUM	27.11	6.05	19.67
	using human seed words			
	QT	23.99	4.36	16.61
	ACESUM	29.53	6.79	21.06

Table 5.8: ROUGE scores of QT and ACESUM for aspect-specific opinion summarization using different sets of seed words.

useful information in a concise manner?), and *Fluency* (is the summary grammatical?). We refer the readers to Appendix A.1 for the full instructions of the experiment.

We compared general summaries produced by the two best performing extractive (LEXRANK, QT) and abstractive (T5-SIMILAR, ACESUM) systems according to ROUGE. We elicited three judgements for all entities in the SPACE and OPOSUM+ test sets. Table 5.9 summarizes our results. BWS values range from -100 (unanimously worst) to 100 (unanimously best). ACESUM is deemed best for all criteria on both datasets. Crowdworkers also rated QT high on informativeness, which indicates that aspect modeling is helpful, but low on other criteria (e.g., coherence and conciseness) due to its extractive nature. We found that crowdworkers rank LEXRANK and QT lower on both coherence and fluency. This is due to their *extractive* nature; since two sentences from different reviews are concatenated together, they can become incoherent and ungrammatical. Abstractive systems such as ACESUM do not have this issue, since new tokens are generated based on previous tokens.

SPACE				
Model	Informative	Coherent	Concise	Fluent
LEXRANK	−48.3	−38.4	−36.9	−43.3
T5-SIMILAR	5.8	<u>11.2</u>	<u>17.2</u>	0.6
QT	<u>20.4</u>	1.3	1.2	<u>2.6</u>
ACESUM	**22.1**	**26.0***	**18.5**	**38.8***

OPOSUM+				
Model	Informative	Coherent	Concise	Fluent
LEXRANK	−27.3	−21.1	−18.2	−23.8
T5-SIMILAR	−31.1	<u>10.0</u>	<u>4.7</u>	<u>−1.9</u>
QT	<u>20.3</u>	−25.3	−21.6	−9.6
ACESUM	**38.1***	**36.3***	**35.2***	**35.3***

Table 5.9: *Best-Worst Scaling* evaluation. Best values are bold-faced while the second best ones are underlined. An asterisk (*) means that the system is significantly better than the second best system (one-way ANOVA with posthoc Tukey HSD tests, $p < 0.05$).

Aspect Controllability We also conducted a user study to assess the quality of aspect-specific summaries. We showed participants the aspect in question as well as aspect summaries from T5-SIMILAR, QT, ACESUM, and HUMAN. Crowdworkers were asked to decide whether the summaries discussed the given aspect *exclusively*, *partially*, or *not at all*. We elicited three judgments for all test entities. We refer the readers to Appendix A.3.1 for the full instructions of the experiment. As can be seen in Table 5.10, SPACE summaries produced by ACESUM exclusively discuss a single aspect 50.9% of the time. T5-SIMILAR mostly produces general summaries (74.8% of them partially discuss the given aspect) which is not surprising, given that it has no special-purpose mechanism for modeling aspect. QT summaries are more topical for the opposite reason. In general, automatic systems perform worse on OPOSUM+ whose larger number of domains renders this dataset more challenging. Finally, we observe a big gap between model and HUMAN performance.

We further verified whether ACESUM can produce summaries covering two aspects. Although it can generate summaries with more aspects (see Table 5.1), we hypothesize that user queries pertaining to two aspects would be most frequent. Be-

	SPACE		
Model	Exclusive	Partial	None
T5-SIMILAR	10.6	74.8	14.6
QT	43.8	39.0	17.1
ACESUM	50.9	42.6	6.5
HUMAN	64.9	31.6	3.5

	OPOSUM+		
Model	Exclusive	Partial	None
T5-SIMILAR	9.4	48.2	42.5
QT	22.2	41.9	35.9
ACESUM	42.2	45.4	12.4
HUMAN	63.0	31.5	5.6

Table 5.10: Proportion of summaries that discuss the target aspect exclusively, partially, or not at all. In the Exclusive column, all pairwise differences are significant ($p < 0.05$; χ^2 test).

sides, if performance with two aspects is inferior, there is little chance it will improve with more aspects. For each test example we elicited three judgments and randomly selected two aspect pairs from the set of all possible aspect combinations. We compared ACESUM against QT (for which we used seed words representing both target aspects). Participants were shown the two aspects and the summaries generated by QT and ACESUM. They were asked to decide whether the summaries discussed (a) both target aspects exclusively (b) one of the aspects (c) other aspects in addition to the target ones, and (d) none of the two aspects. We refer the readers to Appendix A.3.1 for the full instructions of the experiment. The results in Table 5.11 show that ACESUM is able to produce two-aspect summaries effectively 61.3% of the time on SPACE and 47.0% of the time on OPOSUM+. QT on the other hand mostly creates single-aspect summaries.

Summary Veridicality Our third study examined the veridicality of the generated summaries, i.e., whether the opinions mentioned in them are indeed discussed in the input reviews. Participants were shown reviews and corresponding system summaries and were asked to verify, for each sentence of the summary, whether it was fully sup-

	SPACE			
Model	All	One	Other	None
QT	10.0	35.3	34.7	20.0
ACESUM	61.3	19.3	18.0	1.3

	OPOSUM+			
Model	All	One	Other	None
QT	18.8	27.5	33.6	20.1
ACESUM	47.0	16.8	26.8	9.4

Table 5.11: Proportion of target aspects discussed in system summaries (All: both aspects are mentioned; One: only one is mentioned; Other: other aspects are also mentioned; None: no aspects are mentioned). In the All column, the difference between QT and ACESUM is significant ($p < 0.05$; χ^2 test).

ported by the reviews, partially supported, or not at all supported. We performed this experiment on OPOSUM+ only since the number of reviews is small and participants could read them all in a timely fashion. We collected three judgments for all system summaries, both general and aspect-specific ones. Participants assessed the summaries produced by T5-SIMILAR and ACESUM. We also included GOLD-standard summaries as an upper bound but no output from an extractive system as it by default produces veridical summaries which contain facts mentioned in the reviews. We refer the readers to Appendix A.2 for the full instructions of the experiment.

Table 3.6 reports the percentage of fully, partially, and un-supported sentences. Perhaps unsurprisingly, GOLD summaries display the highest percentage of fully supported sentences for both general and aspect-specific summaries. ACESUM and T5-SIMILAR present similar proportions of supported sentences when it comes to general summaries, with ACESUM having a slight advantage. The proportion of supported sentences is higher in aspect summaries for T5-SIMILAR. Note that this model struggles to actually generate aspect-specific summaries (see Table 5.10); instead, it generates any-aspect summaries which maybe veridical but off-topic.

OPOSUM+ General			
Model	Full Support	Partial Support	No Support
T5-SIMILAR	53.3	36.9	9.8
ACESUM	59.9	32.2	8.0
HUMAN	88.4	7.0	4.6

OPOSUM+ Aspect			
Model	Full Support	Partial Support	No Support
T5-SIMILAR	57.3	29.4	13.3
ACESUM	54.2	32.3	13.5
HUMAN	67.8	20.7	11.6

Table 5.12: *Summary veridicality* evaluation. Proportion of summaries that are fully supported, partially supported, or not supported at all. In the Full Support column, only pairwise differences with HUMAN are significant ($p < 0.05$; χ^2 test).

5.4.3 Example Summaries

We provide examples of general and aspect-specific opinion summaries produced by QT, T5-SIMILAR, ACESUM, and HUMAN on SPACE (Figure 5.6) and OPOSUM+ (Figure 5.7). In general, T5-SIMILAR is not able to generate opinion summaries that are specific to a target aspect; it generates summaries that discusses all , despite receiving an aspect query. QT is able to generate aspect-specific summaries, however since the model is extractive, the sentences are inevitably incoherent. For example, the Comfort-specific summary in Figure 5.7 starts with the sentence "Thank you!" which is out of place. ACESUM summaries are more coherent and better follows the aspect query compared to other systems.

We also show two-aspect summaries produced by QT and ACESUM on both datasets in Figure 5.8. When there are two aspects in the query, QT oftentimes cannot produce summaries that focus on the target aspects. On the other hand, ACESUM summaries are able to generate opinions regarding two target aspects. Similar to other models that use synthetic datasets (Amplayo and Lapata, 2020; Bražinskas et al., 2019; Elsahar et al., 2021; Amplayo et al., 2021b), one limitation of ACESUM is that it generates sentences that follow common templates that are usually found in reviews. An example would be the Comfort- and Looks-specific opinion summary for a pair of boots in Figure 5.8, where it ends with a sentence following the template "If you are looking

for ..., do not order this.", a common template found in reviews.

5.5 Summary

In this work, we presented an abstractive approach to aspect-controlled opinion summarization. Key to our model is the induction of aspect controllers which facilitate the creation of a synthetic training dataset and guide summary generation towards the designated aspects. Extensive experiments on two benchmarks show that our model achieves state of the art across the board, for both general and aspect-specific opinion summarization. We also show that our model can effectively generate multi-aspect summaries, simply by modifying the aspect controllers.

HUMAN	**General** Staff was service focused and very welcoming. Common areas of the hotel smelled fresh because of how clean everything was. The rooms were comfortable and came with a fridge and a microwave. Food, both hot and cold, was very well presented and fresh. The hotel was located within walking distance to the French quarter and felt very safe at night. **Cleanliness** The hotel's lounge, bathrooms, hallways, and even the bedding were all clean and even smelled fresh. **Food** The breakfast is very good and plentiful and was more than just continental, offering eggs, sausage and grits in addition to the usual waffles, cereal, and fruit.
ACESUM	**General** The hotel is in a great location, close to the French quarter and the market. The room was clean and comfortable. Breakfast was good, and the staff was very helpful. There is a small restaurant in the lobby. **Cleanliness** The room was clean and comfortable. The bathroom was very clean with a nice shower. **Food** The breakfast was very good, with a variety of choices. The breakfast buffet was good.
T5-SIMILAR	**General** I stayed at the hotel for 3 nights. The room was very clean, the staff was friendly and the breakfast was excellent! The location was great - a short walk to the Eiffel Tower and St Marks Square. Breakfast was good, and the staff was very helpful. There is a small restaurant in the lobby. **Cleanliness** I stayed at the hotel for 3 nights. The staff was very friendly and helpful, the rooms were clean and the breakfast was excellent! The location is great - just a few minutes walk from the Eiffel Tower and it's close to the metro and Bourbon Street. **Food** I stayed at the San Diego for a week. The staff was very friendly and helpful, the rooms were clean and the location was great! The breakfast was good and there wasn't much to do in the room.
QT	**General** Great location. The breakfast was very good. We would definitely stay here again. Room was clean. This hotel is great. The room was large with two queen beds. Nice hotel in a nice location. This is a multi-year award winning hotel. Staff were very helpful. The hotel is very clean. Front desk was friendly and helpful. The room was clean and comfy. The breakfast was average. It is very good. We enjoyed our stay here. **Cleanliness** Pick this one. Toom was clean. The hotel is very clean. Great 5 star service. Room was nice and clean. This one was by far the best. **Food** The breakfast was very good. When you factor in the delicious complimentary breakfast consisting of scrambled eggs, grits, freshly-made waffles, bagels, bacon, sausage, cereal, toast, juice, and coffee.

Figure 5.6: *General* and *aspect-specific* summaries for a hotel generated by four systems (SPACE dataset).

HUMAN	**General** The price is great. Lightweight and comfortable fit in the ear. Based on feedback from others I can be heard clearly. Very easy to use and compatible with all of my phones! It holds a charge great, is light enough. The sound quality is great, but cheap. **Comfort** This unit is very lightweight and comfortable to fit in the ear. Makes it really easy to keep in there. Fits perfectly. **Ease of Use** The headphones are very easy to use and fit almost any cell phone. The only problem is that sometimes it is difficult to know if the equipment is on or off.
ACESUM	**General** I've had this for a few months now and it's still working great. I haven't had any issues with the sound quality, and I am very happy with the quality of the product. The earpiece is very comfortable and is very lightweight. **Comfort** It's a comfortable fit. The earpiece isn't too big, but it fits your ear well. It is a little snug, but my ear is very comfortable. It is comfortable to wear for long periods of time. **Ease of Use** It is easy to use, easy to set up, and easy to connect to my phone. It's not that easy to get on and off.
T5-SIMILAR	**General** I bought this headset for my son. He loves it, but a lot of people have complained about the sound quality of the headset! It is very comfortable and the mic isn't loud enough to hear. It has sounded great. **Comfort** II bought this headset for my husband. He loves it, and it works great! It has a great sound and the sound quality is excellent - the only thing is that the microphone isn't very loud. **Ease of Use** I bought this headset for my husband. He loves it, and it is very comfortable! If you are looking for a good headset, this is the best headset you can buy for the price.
QT	**General** Thank you! The battery life is ... bizarre. Light to the ear. I highly recommend this bluetooth headset. Lightweight and comfortable fit in the ear. I returned it and received a refund. I used it mostly in my car on my commute to work. Great product. **Comfort** I would really like it if it would stay in my ear or if the loop that went around my ear would hold it to my ear. I could not get this headset to work. **Ease of Use** Item delivery just as described! Its made of the cheapest of materials and the bluetooth has a hard time staying connected. My only gripe is that sometimes there's a small lapse between my voice.

Figure 5.7: *General* and *aspect-specific* summaries for the "Bluetooth Headsets" domain generated by four systems (OPOSUM+ dataset).

ACESUM	**Cleanliness** and **Location** of a hotel The hotel is clean and the rooms are very clean. The location is great, right on the beach, and close to the Eiffel Tower. **Food** and **Rooms** of a hotel The breakfast was good, the food was good and the staff was very friendly. The breakfast buffet was good with a variety of choices. **Quality** and **Size** of a laptop bag It's a good size for a laptop. It is not a heavy bag, it is made of a soft material. **Ease of Use** and **Suction Power** of a vacuum I've had this vacuum for a few months now and it's very easy to use. I don't like the fact that it is a little heavy, but it does a great job of picking up the hair. **Comfort** and **Looks** of a pair of boots They are a little tight, and they are not comfortable. They look great with jeans and skirts. If you are looking for a comfortable shoe that will last a long time, do not order this.
QT	**Cleanliness** and **Location** of a hotel Overall we had a nice stay at the hotel. It's well worth the extra money. For the price I paid it underwhelmed ($350 for 1 night). Doesn't get more LA than this have a drink at the roof top. **Food** and **Rooms** of a hotel (Note that breakfast isn't necessarily included in the price.) On the first floor there is a small breakfast room but no restaurant. Also a small but cosy terrace with swimming pool. Rooms are a decent size but walls are paper thin. **Quality** and **Size** of a laptop bag The hand straps have not ripped or torn so really I think the problem was that I put too much weight in the bag. Barely fit a 14 inch HP sleek notebook. I would not recommend this bag. **Ease of Use** and **Suction Power** of a vacuum I even tried putting ear plugs in to vacuum with it, but it still hurts my ears. I looked at every small but powerful vacuum I could find in stores and on line. **Comfort** and **Looks** of a pair of boots Once the weather got cold the shoes became more stiff and they really hurt now so it looks like I wasted $40. I am wondering if they are worth returning or just passing off to someone.

Figure 5.8: Opinion summaries generated by ACESUM and QT focusing on two aspects (SPACE and OPOSUM+ datasets)

Chapter 6

Conclusions and Future Work

6.1 Conclusions

In this thesis, we focused on the task of summarizing opinions in multiple reviews, which is a difficult text generation problem due to the absence of large-scale training datasets. We examined three hypotheses: (a) training datasets for supervised learning can be synthesized from freely available data; (b) content planning allows modeling of opinion variation across multiple reviews and synthesis of naturalistic datasets; and (c) integrating aspect controllers into opinion summarization models enables summary customization based on user preferences.

For the first hypothesis, we introduced a new framework for synthetic dataset creation for opinion summarization in Chapter 3. The framework is based on two steps. The first step is to sample a review and treat it as the pseudo-summary, given a corpus of reviews. The second step is to create a set of reviews to be paired with the summary. Repeating this two-step process creates a large-scale synthetic dataset consisting of review-summary pairs. Using this framework, we proposed to construct synthetic datasets motivated by how humans generate summaries, i.e., by removing non-salient information from reviews. Treating this information as *noise* allows us to simulate this process via noise generating functions to add noise to the pseudo-summary. This process creates a set of noisy reviews, which are then paired with the pseudo-summary. Furthermore, based on this synthetic dataset creation, we designed a summarization model called DENOISESUM that explicitly denoises the input and generates a summary based on it. Experimental results showed that the proposed model improves the overall quality of the summary, outperforming competitive systems by a wide margin.

Reviews in real-world datasets naturally include conflicting opinions from differ-

ent users, which poses new challenges for the creation of synthetic datasets tha resemble real-world data and modeling consensus amongst possibly conflicting opinions. In Chapter 4, we introduced *content planning* for opinion summarization, a process that deals with selecting and structuring content (Kukich, 1983). We used aspect and sentiment probability distributions as content plans, which are automatically induced by learning to reconstruct reviews from aspect and sentiment embeddings. The content plan is then used for both synthetic dataset creation and opinion summarization. Firstly, it is set as a parameter for the Dirichlet distribution, which is used to sample multiple reviews with the right amount of variance in opinions. Secondly, it is incorporated in a summarization model called PLANSUM to guide the model to focus on opinions that reach consensus. We found that the incorporation of content plans improves the quality of synthetic datasets. Our proposed model achieves new state-of-the-art results across multiple datasets from different domains.

Although it is generally assumed that a single generic summary contains sufficient information to answer all user queries, this assumption does not apply in opinion summarization since users may have different preferences and needs. To this end, in Chapter 5, we explored a new formulation of opinion summarization called aspect-controllable opinion summarization, where the goal is to generate summaries focused on one or more target aspects provided by the user. Using the framework proposed in Chapter 3, we created synthetic datasets specifically for the new task. These datasets include multiple *aspect controllers* with different levels of granularity, and are automatically predicted using the multi-instance learning paradigm (Rumelhart et al., 1986). The synthetic datasets are used to fine-tune a pre-trained language model, where the aspect controllers collectively act as the controlling mechanism of the model. At inference time, setting the controllers based on the aspect query allows the model to produce both aspect-specific and generic summaries. On two datasets from different domains, we showed that our model outperforms all previous competing systems in generic and aspect-specific opinion summarization. Further evaluation results also revealed that our model can effectively generate summaries focusing on *multiple* aspects.

Overall, our findings can be summarized as follows:

1. Training datasets for opinion summarization can be synthesized from information freely available in online reviews with minimal domain knowledge.

2. Modeling conflicting opinions across reviews through the use of content plans allows for generation of summaries that capture opinion consensus.

3. Aspect controllers of different granularities enable summarization systems to produce personalized summaries to accommodate different user preferences.

6.2 Future Work

The models presented in this thesis are evaluated in settings that diverge from actual usage in real life. Firstly, while our models can handle up to 100 reviews (the average number of reviews in Rotten Tomatoes and SPACE; see Section 2.2), reviews in real world scenarios can be thousands. Scaling the opinion summarization task would entail identifying many fake reviews, and more generally user-authored content that is not opinion-specific. Secondly, in our experiments, we assumed that personalized queries take the form of a list of pre-defined aspects the user provides. However, this limits users in what they can ask of a system. These limitations on scalability and personalization can be avenues for future research, and we discuss them in detail below.

Massively Multiple Reviews A limitation of recent opinion summarization systems which prevents their deployment in industry is scalability – these systems cannot handle massively multiple reviews as input. This issue is not task-specific; due to memory limitations of modern hardware, neural models cannot be trained with long inputs. There are multiple solutions proposed from the angle of multi-document summarization and long document summarization (see Beltagy et al., 2021 for an overview). One common method is to cut the input length by means of truncation (Fabbri et al., 2019) or sentence selection (Liu et al., 2018a; Liu and Lapata, 2019a). Truncation considers the first few inputs that fit the length limit of models, while sentence selection uses a sentence extraction model that decides which sentences should be included in the input. Another method is chunking (Joshi et al., 2019), where the input is split into multiple chunks (in our case, into individual reviews), which are then processed independently and finally aggregated into one. More recent methods improve the Transformer architecture to model long documents by replacing the attention mechanism with space-efficient alternatives such as averaging instead of weight-summing encodings (Zhang et al., 2018a), introducing global- and local-level attention mechanisms (Beltagy et al., 2020), and low-rank approximations Wang et al. (2020b). Adopting these methods, as well as creating a new benchmark that challenges models with massively multiple reviews as input, would help improve the scalability of opinion summarizers.

Handling Noisy Reviews Another problem that arises when dealing with reviews at scale is the amount of noise they have. Noise in reviews can be classified into three different categories. Firstly, there are reviews that contain grammatical, typological, and structural errors which have arisen due to the colloquialization of online social media language (Dey and Haque, 2009), which have shown to negatively affect the output of neural models (Kumar et al., 2020). Thankfully, there are normalization and grammar correction techniques (Aw et al., 2006; Wang et al., 2020c) to deal with these errors. Secondly, there are also reviews that are highly irrelevant to the target product, such as opinions towards the vendor or a different product (McGlohon et al., 2010). Using anaphora resolution (Jakob and Gurevych, 2010) to identify which target the opinion is directed may help filter out this type of noise. Finally, spam and reviews with fake content are also abundant in online forums (Paul and Nikolaev, 2021). Examples include reviews based on rumors, propaganda news, as well as reviews written with the explicit purpose of harming or boosting the reputation of a product. The detection of such noise has been widely studied, where proposed approaches include a wide variety of features such as content similarity (i.e., comparing reviews with known fake ones; Jindal and Liu, 2007), behavioral patterns (i.e., detecting abnormalities in the user's writing behavior; Mukherjee et al., 2012), and network footprints (i.e., looking at relations and dependencies between review content, reviewer behavior, and product characteristics; Wang et al., 2012). A combination of these methods could remove noise from the set of input reviews and consequently help summarizers produce more informative and correct summaries.

Better Modeling of Aspects and Users In Chapter 5, we assumed that there exists a predefined set of aspects for a specific product. There are two limitations that stem from this assumption. For one thing, the assumption that aspects can be represented as a *list* is not always applicable in review domains. Aspects in products are known to have hierarchies (Kim et al., 2013; e.g., in hotels, the bathroom aspect is part of the room aspect) and dependencies (Zhao et al., 2020; in televisions, the sound quality may depend on connectivity). Topic models that can automatically induce these structures (Kim et al., 2013; Li et al., 2016) can be incorporated with aspect detection modules in our models. Moreover, the assumption that aspects are predefined apriori constrains customization and control in summary generation. Vendors can introduce new features and aspects to their products, which are usually what customers want to have information about (Nowlis and Simonson, 1996). To solve this, Fu et al. (2015)

applied ideas from non-parametric hierarchical Dirichlet process (HDP; Teh et al., 2006) to remove the requirement of setting the number of aspects prior to training. Dirichlet processes can also be extended to neural models that make use of powerful dense representations (Palencia-Olivar et al., 2021). Incorporating these features to our aspect detection models would improve the aspect modeling and personalization of opinion summarizers.

When user-specific data is available, another possible improvement is to incorporate user information into our models. This would potentially restrict user intervention when generating summaries, since the models can directly infer user preferences based on prior user data. Previous work on sentiment classification has explored various methods to effectively represent and inject user information into neural models (Tang et al., 2015; Chen et al., 2016). Tang et al. (2015) proposed that user representations can be learned through what they write in their reviews, while Kim et al. (2019) conducted thorough investigations on which layers in the neural network, such as the embedding and BiLSTM layers, should the user representation be injected into. Other work (Amplayo et al., 2018a; Zhou et al., 2021) aims to resolve issues regarding cold-start users, or new users with few or no prior data, by leveraging information from similar users with many data. Finally, ideas from recommender systems can also be incorporated to improve user modeling by making use of non-textual features such as metadata (Lakiotaki et al., 2011; Kim et al., 2011) and by efficiently learning them on a large scale (Aly et al., 2012).

Textual Queries Natural language is the most intuitive form of user queries, as commonly used in search engines and question answering systems. Replacing the aspect queries used in Chapter 5 with textual queries would give users the freedom to verbalize their information seeking needs and consequently allow opinion summarization systems to produce better personalized summaries. In the question answering domain, ideas from prior work on understanding textual queries (Ishwari et al., 2019) and translating them into machine-readable logical form (Kamath and Das, 2019) can be applied to our task. Specifically, we could automatically transform textual queries into the aspect controllers introduced in Chapter 5. We could also use ideas from query-focused summarization (Daumé III and Marcu, 2006), especially on leveraging distant supervision from question answering (Xu and Lapata, 2020) and creating proxy queries for generic summarization datasets (Xu and Lapata, 2021), which are relevant in opinion summarization since we do not have access to gold-standard query-focused opinion

summaries.

Appendix A

Instructions for Human Evaluation

In the following sections, we present the experimental instructions we provided the human annotators with. All the experiments were judged in Amazon Mechanical Turk (AMT) by turkers with HIT Approval Rate greater than or equal to 98%, Number of HITs Approved greater than or equal to 1000, and who are from one of the following locations: Australia, Canada, New Zealand, United Kingdom, and United States. In each task below, we provide an example data point that was actually shown to annotators during our study.

A.1 Best-Worst Scaling

Instructions

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

Please **read the human summary** first and try to get an overall idea of **opinions** expressed therein.

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

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**.

Example Task

Human Summary
This case is really beautiful great product for the price. Very nice leather and quality product it fits beautifully and the seams began to tear within a week of receiving it. The laptop doesn't fit inside, the quality isn't the best

Automatic Summaries

Summary 1
The case is so small that it's difficult for me to put Macbook air into it and I could hardly take it out of the case. This might be a good case for other 13. I purchased this case for a 13 in Macbook Air, and it doesn't even begin to fit.

Summary 2
It's a nice size for a Macbook pro, but it isn't a big deal. It is very well made, and the material is very nice. The strap is a little stiff and slouchy. If you're looking for something that will fit a laptop, you'll need to be careful with the quality.

Summary 3
Recommended. Great price. The case was too small. I ordered this sleeve for my Lenovo Ideapad. 3 (2013 model). This case is really beautiful and fits my computer snug. I loved this case, it was so inexpensive and my Macbook air fit perfectly in it.

Questions

1. Informativeness
How well does the automatic summary align with the **majority of the aspects** mentioned in the human summary (e.g., image/sound quality of a television, cleanliness/location of a hotel, etc.)?

Best
○ Summary 1 ○ Summary 2 ○ Summary 3
Worst
○ Summary 1 ○ Summary 2 ○ Summary 3

2. Conciseness
The summary should be both brief and **include useful information in a concise manner**.

Best
○ Summary 1 ○ Summary 2 ○ Summary 3
Worst
○ Summary 1 ○ Summary 2 ○ Summary 3

3. Fluency
The summary sentences should be **grammatical, easy to read and understand**.

Best
○ Summary 1 ○ Summary 2 ○ Summary 3
Worst
○ Summary 1 ○ Summary 2 ○ Summary 3

4. 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.

Best
○ Summary 1 ○ Summary 2 ○ Summary 3
Worst
○ Summary 1 ○ Summary 2 ○ Summary 3

A.2 Summary Veridicality

Instructions

You will be presented with a set of user reviews about products. Please **skim through** the reviews below and try to get an overall idea of **opinions** expressed in them.

Then, read a summary of these reviews produced by a computer-based system. For each summary sentence you should decide whether its content is supported by the user reviews. Here you will need to re-read reviews more carefully.

There are three options for you to choose from:

- **Full support**
  All the content is **reflected** in the **reviews**. For example, if a sentence is: "prices were reasonable", then you should be able to find at least one mention of prices or costs being reasonable, or it should be clear from the context.
- **Partial support**
  Only some content is **reflected** in the **reviews**. It's applicable to cases like "great 10x zoom" where the reviews state that the zoom is indeed 10x but only in a negative sentiment.
- **Not supported**
  Content is **not reflected** in the **reviews**. In other words, you can't find any explicit or implicit mentions of the information.

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** .

Example Task

Reviews

Review 1
It was very small and tight. I really did not like it. The pink flowers look really fake and cheap. I have a case on my 13 inch Macbook and its very hard to close it.

Review 2
When I got this case I got it because it said it would fit a 13-14 inch notebook laptop computer. I have a 14 inch and it does not fit. I can not zip it shut. otherwise the sleeve looks nice, is pretty and seems to be in good condition but I gave it 2 stars because I was misled into purchasing this item and it does n't even fit my laptop.

Review 3
When I received the carrying case I thought it was very cute. It looks exactly as it appears in the picture. If you like bright pink then you'll love this! it Sorta looks like a sticker with Japanese design to it. I have to get use to the bright pink... it looks like it Cheapens the look in a way. But I did receive some compliments about the bag- the Japanese design is an eye catcher with the bright pink. I purchased the package deal ($67 which is a good deal compared to the one item for $50 you get from the apple store)... the carrying side bag, case sleeve, pink rubber keyboard cover (which I love! it's a purple pink not too bright).

Review 4
We have a Macbook air and a pro, both 13 inch, and we bought two of this sleeve. fits very well, especially for the pro, looks pretty, and very light weight. One Zipper is n't as smooth when it meets the corner, but that' s not a problem if handled with care.

Review 5
I love my pink floral carrying case for my Mac book. It fits perfectly and provides padding for travel. It' s also stylish, and matches my pink plastic cover for the laptop. It fits nicely into a backpack without a lot of bulk.

Review 6
I swapped my Mac for a smaller Dell laptop Xps, so I needed a smaller case/sleeve. The design on the site looked cute. However, it turned out about the same size as my higher end Marc Jacobs sleeve (which I love, but didn't need all the space) and the quality of this case is terrible, like made in China, bought in Chinatown terrible. I suppose you get what you pay for

Review 7
It's pretty in the picture, but I don't care for it much in person. It's poor quality and I don't trust my laptop to be protected in it. a few months after I bought this I found an amazing brand name laptop case for half this price. too bad it's too late to return this one, but I guess I'll just keep it in my closet.

Review 8
It's really good looking and I feel like it protects my laptop well but sometimes the Zipper gets stuck. I have really push my Macbook in. I have a speck hard cover on my laptop, which may be the reason why, but I saw some other reviews with the same issue. It does fit with a case on it though, you just have to make sure it's really in there.

Review 9
If all you need to do is protect your Macbook, this stylish bag is Well-Constructed and just the thing. wish it had an outer pocket to carry the power cord.

Review 10
I purchased this case for my wife for her new Mbair and she loves it! the

case arrived promptly, it is the correct size, and works as advertised. As I have never purchased a laptop case before, I was a bit surprised how thin it was. It does n't add much bulk to the Super-Thin Mbair and yet provides enough protection during normal use. It fits very nicely in my wife' s purse and looks great!

Summary

It's really good looking like it protects my laptop well it is the correct size, and works as advertised.

Support
○ Full ○ Partial ○ No

It fits perfectly it was very small and tight.

Support
○ Full ○ Partial ○ No

The pink flowers look really fake and cheap.

Support
○ Full ○ Partial ○ No

It's poor quality I have a 14 inch and it does not fit.

Support
○ Full ○ Partial ○ No

But sometimes the zipper gets stuck.

Support
○ Full ○ Partial ○ No

wish it had an outer pocket

Support
○ Full ○ Partial ○ No

A.3 Aspect Controlability

A.3.1 Single Aspect Summaries

Instructions

In this task you will be presented with a number of summaries produced by different systems based on user reviews. Your task is to identify **whether a given aspect of a product is mentioned** in the summary. An **aspect** is a property of a product (e.g., sound quality of a television, location of a hotel).

There are three options for you to choose from:

- **Exclusively mentioned**
  The summary **only mentions** information about the given aspect. For example, if the given aspect is sound quality, the summary "the sound quality is perfect" should be marked with this option.
- **Partially mentioned**
  The summary also mentions **information other than the given aspect**. For example, if the given aspect is sound quality, the summary "the picture and sound quality are nice" should be marked with this option since it also mention information about the picture quality.
- **Not mentioned at all**
  The summary does **not contain any information** regarding the given aspect. For example, if the given aspect is sound quality, the summary "the picture quality is the best" should be marked with this option.

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** .

Example Task

Given Aspect
Feel/Comfort of a **Keyboard**

Summaries

Summary 1
I love this keyboard. It's a great size for my K9, but it doesn't work for me! It is OK if you have to type on your wrists and it works great for the price and the keys are very comfortable and easy to set up and use...

○ Exclusively mentioned
○ Partially mentioned
○ Not mentioned at all

Summary 2
The Rosewill Website helpfully says the warranty is “repair only”. I love Amazon! I liked the feel of the keys however the keyboard died for no reason after three weeks.

- ○ Exclusively mentioned
- ○ Partially mentioned
- ○ Not mentioned at all

Summary 3
This keyboard is comfortable to use, it has a very low key profile, the keys have quick response and it is very quiet.

- ○ Exclusively mentioned
- ○ Partially mentioned
- ○ Not mentioned at all

Summary 4
I’ve been using this keyboard for a couple of months now and it feels great. The keys are comfortable, and the feel of the keys is great!

- ○ Exclusively mentioned
- ○ Partially mentioned
- ○ Not mentioned at all

A.3.2 Double Aspect Summaries

Instructions

In this task you will be presented with a number of summaries produced by different systems based on user reviews. Your task is to identify **whether a given aspect of a product is mentioned** in the summary. An **aspect** is a property of a product (e.g., sound quality of a television, location of a hotel). For each HIT, you will be given **two** aspects.

There are three options for you to choose from:

- **Exclusively both aspects mentioned**
  The summary mentions information **only about the two given aspects**. For example, if the given aspects are sound quality and picture quality, the summary "the picture and sound quality are nice" should be marked with this option.
- **Exclusively one aspect mentioned**
  The summary mentions information regarding **only one of the given aspects**. For example, if the given aspects are sound quality and picture quality, the summary "the picture quality is good" should be marked with this option.
- **Partially mentioned**
  The summary mentions information regarding **at least one of the given aspects**, but **also other aspects that are not given**. For example, if the given aspects are sound quality and picture quality, the summary "the picture quality and the ports are good" should be marked with this option since the review mentions an opinion regarding the connectivity aspect.
- **None of the aspects mentioned**
  The summary does **not contain any information** regarding the given aspects. For example, if the given aspect are sound quality and picture quality, the summary "the TV has many useful ports" should be marked with this option since it only mentions information about the connectivity aspect.

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** .

Example Task

Given Aspect
Looks and **Size** of a **Pair of Boots**

Summaries

Summary 1
They are very comfortable. They are a little tight. The boots aren't too big, and they are so warm.

- ○ Exclusively both aspects mentioned
- ○ Exclusively one aspect mentioned
- ○ Partially mentioned
- ○ None of the aspects mentioned

Summary 2
They do have good ankle support however which is important in icy conditions. They are made by two layers of thick fabric. I will keep them in hoping that they'll loosen up some.

- ○ Exclusively both aspects mentioned
- ○ Exclusively one aspect mentioned
- ○ Partially mentioned
- ○ None of the aspects mentioned

Appendix B

Example LDA Topics for DENOISESUM

Topic 1 (*acting aspect*)	Topic 2 (*franchise genre*)	Topic 12 (*funny comedy*)	Topic 28 (*thriller genre*)
acting	series	funny	thriller
actors	installment	comedy	story
performance	franchise	entertaining	bond
adams	first	surprising	polanski
gibson	potter	extremely	haunting
demme	star	amusing	drama
keaton	harry	story	gripping
amy	trek	family	action
mirren	twilight	sweet	experience
diane	wars	warm	unpredictable

Table B.1: Four example topic-word distributions (out of 50) from the LDA model trained using the **Rotten Tomatoes** dataset. Topic titles in *italics* are manually annotated.

Topic 5 (*mediterranean*)	Topic 8 (*mexican*)	Topic 12 (*great service*)	Topic 14 (*great views*)
hummus	tacos	service	view
pita	mexican	great	great
chicken	salsa	time	romantic
food	burrito	job	anniversary
mediterranean	chips	work	amazing
wrap	margaritas	professional	fountains
filafel	guacamole	company	dinner
shawarma	carne	customer	beautiful
salad	chicken	experience	vegas
kabob	asada	friendly	excellent

Table B.2: Four example topic-word distributions (out of 100) from the LDA model trained using the **Yelp** dataset. Topic titles in *italics* are manually annotated.

Appendix C

Automatic Seed Words

Aspect	"Hotels"
building	lobby beautiful pool decor gym
cleanliness	clean room comfortable nice modern
food	breakfast food buffet restaurant good
location	location walk great close station
rooms	room bed bathroom small clean
service	staff service friendly helpful desk

Table C.1: SPACE seed words automatically induced for six aspects.

Aspect	"Laptop Bags"
looks	looks color pink stylish looked
quality	quality material poor broke durable
size	fit fits size macbook big

Aspect	"Bluetooth Headsets"
comfort	ear fit comfortable fits buds
ease of use	easy button simple setup control
sound quality	sound quality hear noise volume

Aspect	"Boots"
comfort	comfortable fit foot hurt ankle
looks	cute look looked great fringe
size	size ordered half order big

Aspect	"Keyboards"
build quality	working months build stopped quality
feel/comfort	feel comfortable feels keyboard mushy
layout	key keys delete backspace size

Aspect	"Televisions"
connectivity	hdmi computer port usb internet
image quality	picture color colors quality bright
sound quality	sound speakers good quality loud

Aspect	"Vacuums"
accessories	filter brush attachments attachment turbo
ease of use	easy cord push corners vacuuming
suction power	suction picks pick powerful power

Table C.2: OPOSUM+ seed words automatically induced for various "domains" and their aspects.

Bibliography

Akbik, A., Bergmann, T., Blythe, D., Rasul, K., Schweter, S., and Vollgraf, R. (2019). FLAIR: An easy-to-use framework for state-of-the-art NLP. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59, Minneapolis, Minnesota. Association for Computational Linguistics.

Aly, M., Hatch, A., Josifovski, V., and Narayanan, V. K. (2012). Web-scale user modeling for targeting. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12 Companion, page 3–12, New York, NY, USA. Association for Computing Machinery.

Amplayo, R. K., Angelidis, S., and Lapata, M. (2021a). Aspect-controllable opinion summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Amplayo, R. K., Angelidis, S., and Lapata, M. (2021b). Unsupervised opinion summarization with content planning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12489–12497.

Amplayo, R. K., Kim, J., Sung, S., and Hwang, S.-w. (2018a). Cold-start aware user and product attention for sentiment classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2535–2544, Melbourne, Australia. Association for Computational Linguistics.

Amplayo, R. K. and Lapata, M. (2020). Unsupervised opinion summarization with noising and denoising. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1934–1945, Online. Association for Computational Linguistics.

Amplayo, R. K. and Lapata, M. (2021). Informative and controllable opinion summarization. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2662–2672, Online. Association for Computational Linguistics.

Amplayo, R. K., Lim, S., and Hwang, S.-w. (2018b). Entity commonsense representation for neural abstractive summarization. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 697–707, New Orleans, Louisiana. Association for Computational Linguistics.

Amplayo, R. K. and Song, M. (2017). An adaptable fine-grained sentiment analysis for summarization of multiple short online reviews. *Data & Knowledge Engineering*, 110:54 – 67.

Andrews, S., Tsochantaridis, I., and Hofmann, T. (2002). Support vector machines for multiple-instance learning. In *Proceedings of the 15th International Conference on Neural Information Processing Systems*, NIPS'02, page 577–584, Cambridge, MA, USA. MIT Press.

Angelidis, S., Amplayo, R. K., Suhara, Y., Wang, X., and Lapata, M. (2021). Extractive Opinion Summarization in Quantized Transformer Spaces. *Transactions of the Association for Computational Linguistics*, 9:277–293.

Angelidis, S. and Lapata, M. (2018a). Multiple instance learning networks for fine-grained sentiment analysis. *Transactions of the Association for Computational Linguistics*, 6:17–31.

Angelidis, S. and Lapata, M. (2018b). 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*, pages 3675–3686, Brussels, Belgium. Association for Computational Linguistics.

Aw, A., Zhang, M., Xiao, J., and Su, J. (2006). A phrase-based statistical model for SMS text normalization. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 33–40, Sydney, Australia. Association for Computational Linguistics.

Ba, J. L., Kiros, J. R., and Hinton, G. E. (2016). Layer normalization. *arXiv preprint arXiv:1607.06450*.

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations*, San Diego, California.

Banerjee, S. and Lavie, A. (2005). 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*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Beltagy, I., Cohan, A., Hajishirzi, H., Min, S., and Peters, M. E. (2021). Beyond paragraphs: NLP for long sequences. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Tutorials*, pages 20–24, Online. Association for Computational Linguistics.

Beltagy, I., Peters, M. E., and Cohan, A. (2020). Longformer: The long-document transformer. *arXiv:2004.05150*.

Bengio, Y., Courville, A., and Vincent, P. (2013). Representation learning: a review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798—1828.

Blair-Goldensohn, S., Hannan, K., McDonald, R., Neylon, T., Reis, G., and Reynar, J. (2008). Building a sentiment summarizer for local service reviews. In *WWW Workshop on NLP Challenges in the Information Explosion Era (NLPIX)*.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Bowman, S. R., Vilnis, L., Vinyals, O., Dai, A., Jozefowicz, R., and Bengio, S. (2016). Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21, Berlin, Germany. Association for Computational Linguistics.

Bražinskas, A., Lapata, M., and Titov, I. (2019). Unsupervised multi-document opinion summarization as copycat-review generation. *arXiv preprint arXiv:1911.02247*.

Cao, S. and Wang, L. (2021). Inference time style control for summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5942–5953, Online. Association for Computational Linguistics.

Cao, Z., Wei, F., Dong, L., Li, S., and Zhou, M. (2015). Ranking with recursive neural networks and its application to multi-document summarization. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, page 2153–2159. AAAI Press.

Carbonell, J. and Goldstein, J. (1998). The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, page 335–336, New York, NY, USA. Association for Computing Machinery.

Carenini, G. and Moore, J. D. (2006). Generating and evaluating evaluative arguments. *Artif. Intell.*, 170(11):925–952.

Carenini, G., Ng, R., and Pauls, A. (2006). Multi-document summarization of evaluative text. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, Trento, Italy. Association for Computational Linguistics.

Chatterjee, P. (2001). Online reviews: Do consumers use them? *Advances in Consumer Research*, 28:129–133.

Chen, H., Sun, M., Tu, C., Lin, Y., and Liu, Z. (2016). Neural sentiment classification with user and product attention. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1650–1659, Austin, Texas. Association for Computational Linguistics.

Chen, Y.-S. and Shuai, H.-H. (2021). Meta-transfer learning for low-resource abstractive summarization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(14):12692–12700.

Chevalier, J. A. and Mayzlin, D. (2006). The effect of word of mouth on sales: Online book reviews. *Journal of Marketing Research*, 43(3):345–354.

Choi, E., Palomaki, J., Lamm, M., Kwiatkowski, T., Das, D., and Collins, M. (2021). Decontextualization: Making sentences stand-alone. *Transactions of the Association for Computational Linguistics*, 9:447–461.

Chu, E. and Liu, P. (2019). MeanSum: A neural model for unsupervised multi-document abstractive summarization. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 1223–1232, Long Beach, California.

Coavoux, M., Elsahar, H., and Gallé, M. (2019). Unsupervised aspect-based multi-document abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 42–47, Hong Kong, China. Association for Computational Linguistics.

Cronen-Townsend, S., Zhou, Y., and Croft, W. B. (2002). Predicting query performance. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '02, page 299–306, New York, NY, USA. Association for Computing Machinery.

Dang, H. T. (2006). DUC 2005: Evaluation of question-focused summarization systems. In *Proceedings of the Workshop on Task-Focused Summarization and Question Answering*, pages 48–55, Sydney, Australia. Association for Computational Linguistics.

Daumé III, H. and Marcu, D. (2006). Bayesian query-focused summarization. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 305–312, Sydney, Australia. Association for Computational Linguistics.

Dave, K., Lawrence, S., and Pennock, D. M. (2003). Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th International Conference on World Wide Web*, WWW '03, page 519–528, New York, NY, USA. Association for Computing Machinery.

Denkowski, M. and Lavie, A. (2014). Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the 9th Workshop on Statistical Machine Translation*, pages 376–380, Baltimore, Maryland. Association for Computational Linguistics.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Dey, L. and Haque, S. M. (2009). Opinion mining from noisy text data. *International Journal on Document Analysis and Recognition (IJDAR)*, 12(3):205–226.

Di Fabbrizio, G., Stent, A., and Gaizauskas, R. (2014). A hybrid approach to multi-document summarization of opinions in reviews. In *Proceedings of the 8th International Natural Language Generation Conference (INLG)*, pages 54–63, Philadelphia, Pennsylvania. Association for Computational Linguistics.

Dietterich, T. G., Lathrop, R. H., and Lozano-Pérez, T. (1997). Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1-2):31–71.

Duan, W., Gu, B., and Whinston, A. B. (2008). Do online reviews matter? — an empirical investigation of panel data. *Decision Support Systems*, 45(4):1007–1016. Information Technology and Systems in the Internet-Era.

Elman, J. L. (1990). Finding structure in time. *Cognitive science*, 14(2):179–211.

Elsahar, H., Coavoux, M., Rozen, J., and Gallé, M. (2021). 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*, pages 1646–1662, Online. Association for Computational Linguistics.

Eppler, M. J. and Mengis, J. (2008). *The Concept of Information Overload - A Review of Literature from Organization Science, Accounting, Marketing, MIS, and Related Disciplines (2004)*, pages 271–305. Gabler, Wiesbaden.

Erkan, G. and Radev, D. R. (2004). Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22(1):457–479.

Fabbri, A., Li, I., She, T., Li, S., and Radev, D. (2019). 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*, pages 1074–1084, Florence, Italy. Association for Computational Linguistics.

Fabbri, A. R., Kryściński, W., McCann, B., Xiong, C., Socher, R., and Radev, D. (2021). SummEval: Re-evaluating Summarization Evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409.

Fan, A., Grangier, D., and Auli, M. (2018). Controllable abstractive summarization. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 45–54, Melbourne, Australia. Association for Computational Linguistics.

Fevry, T. and Phang, J. (2018). Unsupervised sentence compression using denoising auto-encoders. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 413–422, Brussels, Belgium. Association for Computational Linguistics.

Freitag, M. and Roy, S. (2018). Unsupervised natural language generation with denoising autoencoders. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3922–3929, Brussels, Belgium. Association for Computational Linguistics.

Fu, X., Yang, K., Huang, J. Z., and Cui, L. (2015). Dynamic non-parametric joint sentiment topic mixture model. *Knowledge-Based Systems*, 82:102–114.

Gamon, M., Aue, A., Corston-Oliver, S., and Ringger, E. (2005). Pulse: Mining customer opinions from free text. In Famili, A. F., Kok, J. N., Peña, J. M., Siebes, A., and Feelders, A., editors, *Advances in Intelligent Data Analysis VI*, pages 121–132, Berlin, Heidelberg. Springer Berlin Heidelberg.

Ganesan, K., Zhai, C., and Han, J. (2010). Opinosis: A graph based approach to abstractive summarization of highly redundant opinions. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 340–348, Beijing, China. Coling 2010 Organizing Committee.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., March, M., and Lempitsky, V. (2016). Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(59):1–35.

Gehrmann, S., Dai, F., Elder, H., and Rush, A. (2018). End-to-end content and plan selection for data-to-text generation. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 46–56, Tilburg University, The Netherlands. Association for Computational Linguistics.

Glorot, X., Bordes, A., and Bengio, Y. (2011a). Deep sparse rectifier neural networks. In Gordon, G., Dunson, D., and Dudík, M., editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 315–323, Fort Lauderdale, FL, USA. PMLR.

Glorot, X., Bordes, A., and Bengio, Y. (2011b). Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, pages 513–520, Bellevue, Washington. Omnipress.

Grusky, M., Naaman, M., and Artzi, Y. (2018). 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)*, pages 708–719, New Orleans, Louisiana. Assocation for Computational Linguistics.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

He, R., Lee, W. S., Ng, H. T., and Dahlmeier, D. (2017). An unsupervised neural attention model for aspect extraction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 388–397, Vancouver, Canada. Association for Computational Linguistics.

Hellinger, E. (1909). Neue begründung der theorie quadratischer formen von unendlichvielen veränderlichen. *Journal für die reine und angewandte Mathematik (Crelles Journal)*, 1909(136):210–271.

Hermann, K. M., Kocisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., and Blunsom, P. (2015). Teaching machines to read and comprehend. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R., editors, *Advances in*

Neural Information Processing Systems 28, pages 1693–1701. Curran Associates, Inc.

Hinton, G., McClelland, J., and Rumelhart, D. (1986). Distributed representations. In *Parallel distributed processing: explorations in the microstructure of cognition, vol. 1: foundations*, pages 77–109.

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

Holtzman, A., Buys, J., Forbes, M., and Choi, Y. (2019). The curious case of neural text degeneration. *CoRR*, abs/1904.09751.

Hu, M. and Liu, B. (2004a). Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, page 168–177, New York, NY, USA. Association for Computing Machinery.

Hu, M. and Liu, B. (2004b). Mining opinion features in customer reviews. In *Proceedings of the 19th National Conference on Artifical Intelligence*, AAAI'04, page 755–760. AAAI Press.

Hu, M. and Liu, B. (2006). Opinion extraction and summarization on the web. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 2*, pages 1621–1624, Boston, Massachusetts. AAAI Press.

Hu, N., Liu, L., and Zhang, J. J. (2008). Do online reviews affect product sales? the role of reviewer characteristics and temporal effects. *Inf. Technol. and Management*, 9(3):201–214.

Hua, X. and Wang, L. (2019). 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)*, pages 591–602, Hong Kong, China. Association for Computational Linguistics.

Ishwari, K., Aneeze, A., Sudheesan, S., Karunaratne, H., Nugaliyadde, A., and Mallawarrachchi, Y. (2019). Advances in natural language question answering: A review. *arXiv preprint arXiv:1904.05276*.

Jakob, N. and Gurevych, I. (2010). Using anaphora resolution to improve opinion target identification in movie reviews. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 263–268, Uppsala, Sweden. Association for Computational Linguistics.

Jindal, N. and Liu, B. (2007). Analyzing and detecting review spam. In *Seventh IEEE International Conference on Data Mining (ICDM 2007)*, pages 547–552.

Joshi, M., Levy, O., Zettlemoyer, L., and Weld, D. (2019). BERT for coreference resolution: Baselines and analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5803–5808, Hong Kong, China. Association for Computational Linguistics.

Kamath, A. and Das, R. (2019). A survey on semantic parsing. In *1st Conference on Automated Knowledge Base Construction, AKBC 2019, Amherst, MA, USA, May 20-22, 2019*.

Kan, M.-Y. and McKeown, K. R. (2002). Corpus-trained text generation for summarization. In *Proceedings of the International Natural Language Generation Conference*, pages 1–8, Harriman, New York, USA. Association for Computational Linguistics.

Karamanolakis, G., Hsu, D., and Gravano, L. (2019). Training neural networks for aspect extraction using descriptive keywords only. In *Proceedings of the The 2nd Learning from Limited Labeled Data (LLD) Workshop*.

Keeler, J. and Rumelhart, D. E. (1991). A self-organizing integrated segmentation and recognition neural net. In *Proceedings of the 4th International Conference on Neural Information Processing Systems*, page 496–503, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Kikuchi, Y., Neubig, G., Sasano, R., Takamura, H., and Okumura, M. (2016). Controlling output length in neural encoder-decoders. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1328–1338, Austin, Texas. Association for Computational Linguistics.

Kim, H.-N., Alkhaldi, A., El Saddik, A., and Jo, G.-S. (2011). Collaborative user modeling with user-generated tags for social recommender systems. *Expert Systems with Applications*, 38(7):8488–8496.

Kim, J., Amplayo, R. K., Lee, K., Sung, S., Seo, M., and Hwang, S.-w. (2019). Categorical metadata representation for customized text classification. *Transactions of the Association for Computational Linguistics*, 7:201–215.

Kim, S., Zhang, J., Chen, Z., Oh, A., and Liu, S. (2013). A hierarchical aspect-sentiment model for online reviews. AAAI'13, page 526–533. AAAI Press.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *Proceedings of the 4th International Conference on Learning Representations*, San Diego, California.

Kingma, D. P. and Welling, M. (2014). Auto-encoding variational Bayes. In *Proceedings of the 3rd International Conference on Learning Representations*, Banff, Alberta.

Kiritchenko, S. and Mohammad, S. (2017). Best-worst scaling more reliable than rating scales: A case study on sentiment intensity annotation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 465–470, Vancouver, Canada. Association for Computational Linguistics.

Kolen, J. F. and Kremer, S. C. (2001). *Gradient Flow in Recurrent Nets: The Difficulty of Learning LongTerm Dependencies*, pages 237–243.

Kotzias, D., Denil, M., Blunsom, P., and de Freitas, N. (2014). Deep multi-instance transfer learning. *arXiv preprint arXiv:1411.3128*.

Kotzias, D., Denil, M., De Freitas, N., and Smyth, P. (2015). From group to individual labels using deep features. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 597–606, New York, NY, USA.

Ku, L.-W., Liang, Y.-T., and Chen, H.-H. (2006). Opinion extraction, summarization and tracking in news and blog corpora. In *AAAI Symposium on Computational Approaches to Analysing Weblogs (AAAI-CAAW)*, pages 100–107, Palo Alto, California.

Kukich, K. (1983). Design of a knowledge-based report generator. In *21st Annual Meeting of the Association for Computational Linguistics*, pages 145–150, Cambridge, Massachusetts, USA. Association for Computational Linguistics.

Kumar, A., Makhija, P., and Gupta, A. (2020). Noisy text data: Achilles' heel of BERT. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, pages 16–21, Online. Association for Computational Linguistics.

Kwon, B. C., Kim, S.-H., Duket, T., Catalán, A., and Yi, J. S. (2015). Do people really experience information overload while reading online reviews? *International Journal of Human–Computer Interaction*, 31(12):959–973.

Lakiotaki, K., Matsatsinis, N. F., and Tsoukiàs, A. (2011). Multicriteria user modeling in recommender systems. *IEEE Intelligent Systems*, 26(2):64–76.

Lerman, K., Blair-Goldensohn, S., and McDonald, R. (2009). Sentiment summarization: Evaluating and learning user preferences. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 514–522. Association for Computational Linguistics.

Li, J., Liao, M., Gao, W., He, Y., and Wong, K.-F. (2016). Topic extraction from microblog posts using conversation structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2114–2123, Berlin, Germany. Association for Computational Linguistics.

Li, P., Wang, Z., Lam, W., Ren, Z., and Bing, L. (2017). Salience estimation via variational auto-encoders for multi-document summarization. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 3497–3503. AAAI Press.

Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Lin, C.-Y. and Hovy, E. (2003). Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 150–157.

Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167.

Liu, B., Hu, M., and Cheng, J. (2005). Opinion observer: Analyzing and comparing opinions on the web. In *Proceedings of the 14th International Conference on World Wide Web*, pages 342–351, New York, New York. ACM.

Liu, P. J., Saleh, M., Pot, E., Goodrich, B., Sepassi, R., Kaiser, L., and Shazeer, N. (2018a). Generating Wikipedia by summarizing long sequences. In *Proceedings of the 7th International Conference on Learning Representations*, Vancouver, Canada.

Liu, Y. and Lapata, M. (2019a). Hierarchical transformers for multi-document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5070–5081, Florence, Italy. Association for Computational Linguistics.

Liu, Y. and Lapata, M. (2019b). 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)*, pages 3730–3740, Hong Kong, China. Association for Computational Linguistics.

Liu, Y., Luo, Z., and Zhu, K. (2018b). Controlling length in abstractive summarization using a convolutional neural network. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4110–4119, Brussels, Belgium. Association for Computational Linguistics.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Loshchilov, I. and Hutter, F. (2019). Decoupled weight decay regularization. In *International Conference on Learning Representations*, New Orleans, LA, USA.

Louviere, J. J., Flynn, T. N., and Marley, A. A. J. (2015). *Best-Worst Scaling: Theory, Methods and Applications*. Cambridge University Press.

Lu, Y., Zhai, C., and Sundaresan, N. (2009). Rated aspect summarization of short comments. In *Proceedings of the 18th International Conference on World Wide Web*, pages 131–140, Madrid, Spain. ACM.

Malhotra, N. K. (1984). Reflections on the information overload paradigm in consumer decision making. *Journal of Consumer Research*, 10(4):436–440.

Maron, O. and Ratan, A. L. (1998). Multiple-instance learning for natural scene classification. In *Proceedings of the Annual International Conference on Machine Learning*, volume 98, pages 341–349.

Maynez, J., Narayan, S., Bohnet, B., and McDonald, R. (2020). On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.

McAuley, J., Targett, C., Shi, Q., and van den Hengel, A. (2015). Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, page 43–52, New York, NY, USA. Association for Computing Machinery.

McGlohon, M., Glance, N., and Reiter, Z. (2010). Star quality: Aggregating reviews to rank products and merchants. In *Proceedings of Fourth International Conference on Weblogs and Social Media (ICWSM)*, Washington, DC.

McKeown, K. (1985). *Text Generation*. Studies in Natural Language Processing. Cambridge University Press.

Mei, Q., Ling, X., Wondra, M., Su, H., and Zhai, C. (2007). Topic sentiment mixture: Modeling facets and opinions in weblogs. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, page 171–180, New York, NY, USA. Association for Computing Machinery.

Mihalcea, R. and Tarau, P. (2004). TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, page 3111–3119, Red Hook, NY, USA. Curran Associates Inc.

Moryossef, A., Goldberg, Y., and Dagan, I. (2019). 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)*, pages 2267–2277, Minneapolis, Minnesota. Association for Computational Linguistics.

Moussa, M. E., Mohamed, E. H., and Haggag, M. H. (2018). A survey on opinion summarization techniques for social media. *Future Computing and Informatics Journal*, 3(1):82–109.

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

Mukherjee, A., Liu, B., and Glance, N. (2012). Spotting fake reviewer groups in consumer reviews. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, page 191–200, New York, NY, USA. Association for Computing Machinery.

Mukherjee, R., Peruri, H. C., Vishnu, U., Goyal, P., Bhattacharya, S., and Ganguly, N. (2020). *Read What You Need: Controllable Aspect-Based Opinion Summarization of Tourist Reviews*, page 1825–1828. Association for Computing Machinery, New York, NY, USA.

Murray, G., Hoque, E., and Carenini, G. (2017). Chapter 11 - opinion summarization and visualization. In Pozzi, F. A., Fersini, E., Messina, E., and Liu, B., editors, *Sentiment Analysis in Social Networks*, pages 171–187. Morgan Kaufmann, Boston.

Narayan, S., Zhao, Y., Maynez, J., Simões, G., Nikolaev, V., and McDonald, R. (2021). Planning with learned entity prompts for abstractive summarization. *Transactions of the Association for Computational Linguistics*, 9:1475–1492.

Nenkova, A. and McKeown, K. (2012). *A Survey of Text Summarization Techniques*, pages 43–76. Springer US, Boston, MA.

Nenkova, A. and Vanderwende, L. (2005). The impact of frequency on summarization. *Microsoft Research, Redmond, Washington, Tech. Rep. MSR-TR-2005*, 101.

Ni, J., Li, J., and McAuley, J. (2019). 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)*, pages 188–197, Hong Kong, China. Association for Computational Linguistics.

Nowlis, S. M. and Simonson, I. (1996). The effect of new product features on brand choice. *Journal of Marketing Research*, 33(1):36–46.

Palencia-Olivar, M., Bonnevay, S., Aussem, A., and Canitia, B. (2021). Neural embedded dirichlet processes for topic modeling. In Torra, V. and Narukawa, Y., editors, *Modeling Decisions for Artificial Intelligence*, pages 299–310, Cham. Springer International Publishing.

Pan, S. J. and Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359.

Pang, B. and Lee, L. (2004). A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics.

Pang, B. and Lee, L. (2005). Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, page 115–124, USA. Association for Computational Linguistics.

Pang, B. and Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135.

Pang, B., Lee, L., and Vaithyanathan, S. (2002). Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 79–86. Association for Computational Linguistics.

Paul, H. and Nikolaev, A. (2021). Fake review detection on online e-commerce platforms: a systematic literature review. *Data Mining and Knowledge Discovery*, pages 1–52.

Paul, M., Zhai, C., and Girju, R. (2010). Summarizing contrastive viewpoints in opinionated text. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 66–76, Cambridge, MA. Association for Computational Linguistics.

Paulus, R., Xiong, C., and Socher, R. (2018). A deep reinforced model for abstractive summarization. In *International Conference on Learning Representations*, Vancouver, Canada.

Pennington, J., Socher, R., and Manning, C. (2014). GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Popescu, A.-M. and Etzioni, O. (2005). Extracting product features and opinions from reviews. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 339–346, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Puduppully, R., Dong, L., and Lapata, M. (2019). Data-to-text generation with content selection and planning. In *AAAI Conference on Artificial Intelligence*, pages 6908–6915.

Radev, D. R., Jing, H., and Budzikowska, M. (2000). Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation, and user studies. In *NAACL-ANLP 2000 Workshop: Automatic Summarization*.

Radev, D. R., Jing, H., Styś, M., and Tam, D. (2004). Centroid-based summarization of multiple documents. *Information Processing & Management*, 40(6):919–938.

Radford, A., Józefowicz, R., and Sutskever, I. (2017). Learning to generate reviews and discovering sentiment. *CoRR*, abs/1704.01444.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Ramachandran, P., Liu, P., and Le, Q. (2017). Unsupervised pretraining for sequence to sequence learning. In *Proceedings of the 2017 Conference on Empirical Methods in*

Natural Language Processing, pages 383–391, Copenhagen, Denmark. Association for Computational Linguistics.

Ray, S. and Craven, M. (2005). Supervised versus multiple instance learning: An empirical comparison. In *Proceedings of the 22nd International Conference on Machine Learning*, ICML '05, page 697–704, New York, NY, USA. Association for Computing Machinery.

Reimers, N. and Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Rohrbach, A., Hendricks, L. A., Burns, K., Darrell, T., and Saenko, K. (2018). Object hallucination in image captioning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4035–4045, Brussels, Belgium. Association for Computational Linguistics.

Rossiello, G., Basile, P., and Semeraro, G. (2017). Centroid-based text summarization through compositionality of word embeddings. In *Proceedings of the MultiLing 2017 Workshop on Summarization and Summary Evaluation Across Source Types and Genres*, pages 12–21, Valencia, Spain. Association for Computational Linguistics.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, 323(6088):533–536.

Rush, A. M., Chopra, S., and Weston, J. (2015). A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, Lisbon, Portugal. Association for Computational Linguistics.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

See, A., Liu, P. J., and Manning, C. D. (2017). 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), pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Sennrich, R., Haddow, B., and Birch, A. (2016). 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)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Sharma, E., Huang, L., Hu, Z., and Wang, L. (2019). An entity-driven framework for abstractive summarization. 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)*, pages 3280–3291, Hong Kong, China. Association for Computational Linguistics.

Silberer, C. and Lapata, M. (2014). Learning grounded meaning representations with autoencoders. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 721–732, Baltimore, Maryland. Association for Computational Linguistics.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., and Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Soto-Acosta, P., Molina-Castillo, F. J., Lopez-Nicolas, C., and Colomo-Palacios, R. (2014). The effect of information overload and disorganisation on intention to purchase online: The role of perceived risk and internet experience. *Online Information Review*.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958.

Suhara, Y., Wang, X., Angelidis, S., and Tan, W.-C. (2020). OpinionDigest: A simple framework for opinion summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5789–5798, Online. Association for Computational Linguistics.

Surdeanu, M., Tibshirani, J., Nallapati, R., and Manning, C. D. (2012). Multi-instance multi-label learning for relation extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 455–465, Jeju Island, Korea. Association for Computational Linguistics.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 3104–3112, Cambridge, MA, USA. MIT Press.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Tang, D., Qin, B., and Liu, T. (2015). Learning semantic representations of users and products for document level sentiment classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1014–1023, Beijing, China. Association for Computational Linguistics.

Tay, W., Joshi, A., Zhang, X., Karimi, S., and Wan, S. (2019). 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*, pages 52–60, Sydney, Australia. Australasian Language Technology Association.

Teh, Y. W., Jordan, M. I., Beal, M. J., and Blei, D. M. (2006). Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581.

Thet, T. T., Na, J.-C., and Khoo, C. S. (2010). Aspect-based sentiment analysis of movie reviews on discussion boards. *Journal of Information Science*, 36(6):823–848.

Titov, I. and McDonald, R. (2008). A joint model of text and aspect ratings for sentiment summarization. In *Proceedings of ACL-08: HLT*, pages 308–316, Columbus, Ohio. Association for Computational Linguistics.

van den Oord, A., Vinyals, O., and Kavukcuoglu, K. (2017). Neural discrete representation learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, page 6309–6318, Red Hook, NY, USA. Curran Associates Inc.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, u., and Polosukhin, I. (2017). Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P.-A. (2008). Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, pages 1096–1103, Helsinki, Finland.

Vinyals, O., Fortunato, M., and Jaitly, N. (2015). Pointer networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, pages 2692–2700, Montréal, Canada.

Wang, G., Xie, S., Liu, B., and Yu, P. S. (2012). Identify online store review spammers via social review graph. 3(4).

Wang, K., Chang, B., and Sui, Z. (2020a). A spectral method for unsupervised multi-document summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 435–445, Online. Association for Computational Linguistics.

Wang, L. and Ling, W. (2016). Neural network-based abstract generation for opinions and arguments. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 47–57, San Diego, California. Association for Computational Linguistics.

Wang, S., Li, B. Z., Khabsa, M., Fang, H., and Ma, H. (2020b). Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*.

Wang, Y., Wang, Y., Liu, J., and Liu, Z. (2020c). A comprehensive survey of grammar error correction. *arXiv preprint arXiv:2005.06600*.

Wei, X.-S., Wu, J., and Zhou, Z.-H. (2014). Scalable multi-instance learning. In *Proceedings of the 2014 IEEE International Conference on Data Mining*, pages 1037–1042, Shenzhen, China.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Wu, H., Gu, Y., Sun, S., and Gu, X. (2016a). Aspect-based opinion summarization with convolutional neural networks. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 3157–3163.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al. (2016b). Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. (2019). How powerful are graph neural networks? In *International Conference on Learning Representations*.

Xu, Y. and Lapata, M. (2019). Weakly supervised domain detection. *Transactions of the Association for Computational Linguistics*, 7:581–596.

Xu, Y. and 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)*, pages 3632–3645, Online. Association for Computational Linguistics.

Xu, Y. and Lapata, M. (2021). Generating query focused summaries from query-free resources. 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)*, pages 6096–6109, Online. Association for Computational Linguistics.

Zhang, B., Xiong, D., and Su, J. (2018a). Accelerating neural transformer via an average attention network. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1789–1798, Melbourne, Australia. Association for Computational Linguistics.

Zhang, J., Tan, J., and Wan, X. (2018b). Adapting neural single-document summarization model for abstractive multi-document summarization: A pilot study. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 381–390, Tilburg University, The Netherlands. Association for Computational Linguistics.

Zhao, C. and Chaturvedi, S. (2020). Weakly-supervised opinion summarization by leveraging external information. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9644–9651.

Zhao, P., Hou, L., and Wu, O. (2020). Modeling sentiment dependencies with graph convolutional networks for aspect-level sentiment classification. *Knowledge-Based Systems*, 193:105443.

Zhou, D., Zhang, M., Zhang, L., and He, Y. (2021). A neural group-wise sentiment analysis model with data sparsity awareness. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14594–14601.

Zhou, Z.-H., Sun, Y.-Y., and Li, Y.-F. (2009). Multi-instance learning by treating instances as non-iid samples. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1249–1256, Montreal, Quebec, Canada.

Zhuang, L., Jing, F., and Zhu, X.-Y. (2006). Movie review mining and summarization. CIKM '06, page 43–50, New York, NY, USA. Association for Computing Machinery.