Neural Aspect-based Text Generation

Hiroaki Hayashi

CMU-LTI-21-017

Language Technologies Institute
School of Computer Science
Carnegie Mellon University
5000 Forbes Ave., Pittsburgh, PA 15123
www.lti.cs.cmu.edu

Thesis Committee:

Graham Neubig (Chair)	Carnegie Mellon University
Yulia Tsvetkov	Carnegie Mellon University
Alan W. Black	Carnegie Mellon University
Dragomir Radev	Yale University

*Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy.*

Keywords: Neural Language Generation, Aspect-based Natural Language Generation, Text Summarization

To my father and my mother,
To my wife, Tania.

Abstract

Advances in neural sequence models and large-scale pre-trained language models have made a great impact on natural language generation, achieving impressive performance on different tasks. However when users of such systems have a specific interest in what content to generate, these models fail to address such needs. To control the content of the generated text more accurately, one could specify an *aspect* of interest, a semantic property of a current topic that specifies a particular subset of content, and perform aspect-based generation. For example, a user may be interested in knowing more about price from the collection of product reviews. Despite the potential utility of such methods, aspect-based generation has received relatively little research attention. One of the reasons is the lack of available data resources to train and apply models in a variety of domains. In addition, what little work exists on aspect-based generation describes these aspects in simple forms: a set of labels that specifies the overall subtopics in the generated text. However, in reality the content of a text should be specified with different granularities and with respect to other aspects of interest when necessary.

In this thesis, we propose methods to address these issues, allowing for better control of the generated texts. This thesis consists of three parts. First, we address the lack of domain diversity in data sources for aspect-based summarization by reformulating Wikipedia article generation as multi-document aspect-based summarization. We examine the summarization performance on 20 domains and highlight domain-specific challenges. Leveraging this dataset, we then explore zero-shot domain transfer of aspect-based summarization models, with models capable of handling arbitrary aspects at testing time. Next, we focus on incorporating structures into aspects. In this part, we turn our focus to scientific survey articles, which are organized by human authors using section structures. Using this structure, we formulate survey article generation as a summarization task and investigate the use of structural prompts for aspect-based summarization. In the last part, we aim to achieve fine-grained content control with fine-grained aspects. In particular, we learn a language model on Wikipedia texts about single entities by conditioning on a local knowledge base that stores information about various aspects of the entities. We design a model capable of automatically switching between token-by-token and aspect-based generation based on the likelihood of the text.

Acknowledgments

I would like to thank my PhD advisor Graham Neubig for his continued guidance and support in every aspect of this thesis. He always put me back on track and helped me realize the next steps when I tend to focus on non-significant matters. From avoiding NaN in softmax calculations to how to conduct a research, I learned so many levels of skills from his advice.

I would also like to thank my committee members, Yulia Tsvetkov, Alan Black, and Dragomir Radev, for accepting to be a part of my committee and for their insightful and constructive feedback to improve this thesis.

I am deeply grateful to have the opportunity to work with my bright collaborators and mentors: Pengfei Liu, Jayanth Koushik, Eduard Hovy, Teruko Mitamura, Zecong Hu, Chenyan Xiong, Zi-Yi Dou, Zhengbao Jiang, Junjie Hu, Shonosuke Ishiwatari, and Yixin Liu. In particular, Pengfei was always available and shared his time to discuss research and other things with me during the latter half of my PhD, for which I cannot thank enough. As a master's student, it was fun to go on a search of an empty meeting room with Jayanth to spend a night working together. I would also like to thank my internship collaborators: Wojciech Kryściński and Bryan McCann for their mentoring.

I am thankful to have strong support from many friends and peers who listened to me and gave me guidance, including Dongyeop Kang, Frederick Liu, Naoki Otani, Hector Liu, Varun Gangal, Dheeraj Rajagopal, Chitanya Ahuja, Kazuya Kawakami, Yoichi Matsuyama, and NeuLab members.

Finally, I want to acknowledge those who had the most important role during my studies. I want to thank all my family: my wife Tania, my father Shunyu, my mother Hiromi, my sister Yuki, my niece Suzuka, señor Julio, señora Nora, Diana, Uriel, Perrito, Tarrito, for their unconditional love and support. Messages, calls, and letters from my parents gave me incredible strength to go through hardships. Most importantly, my PhD journey would have never been possible without Tania sharing every emotion with me, supporting me, and loving me always. This thesis is dedicated to all of you.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Natural Language Generation (NLG) is a field of research that studies automatic generation of natural language utterances with or without conditioning on particular inputs. In the absence of conditioning input, NLG can be applied to free-form text generation such as language modeling or story generation. Perhaps the more popular use case is NLG with conditioning inputs, such as machine translation, dialogue response generation, or data-to-text generation tasks, whose goal is to reflect the conditioning context in the generated output based on the task requirements. Historically, NLG systems took the form of pipeline systems that consist of modules like content selection, lexicalization, and linguistic realization (Gatt and Krahmer, 2018). While these systems are transparent in terms of what *content* is going to be included in the generated texts, generated texts lacked in fluency due to limited lexical diversity in the outputs. More recently, data-driven NLG systems have observed great success by incorporating neural network-based methods (Bahdanau et al., 2015; Bengio et al., 2003; Sutskever et al., 2014) especially on tasks with clear associations between inputs and outputs (Castro Ferreira et al., 2017; Lebret et al., 2016; Wen et al., 2015). The state-of-the-art has been further pushed forward with the pre-training of large scale sequence models on abundant data (Devlin et al., 2019; Lewis et al., 2020). Neural methods particularly excel at learning from rich representation of inputs and providing high fluency in the generation outputs. Despite these improvements, a number of new challenges have emerged due to the way these models generate text. For example, they lack the transparency of more conventional systems, making it not only difficult to understand the model behaviors but also difficult to control the model to generate texts with desired content.

One of the most important situations that requires such control is when we consider multiple possible outputs different users may be interested in given common inputs. For example, when summarizing human-authored reviews of a TV, users might have different things they would want to know about the TV when reading a review, such as information regarding the price, quality, and so on. When discussing a certain topic such as Barack Obama, users may be interested in different subtopics such as information regarding his political career, biography, legacy, and so on. In both cases, users would benefit from the ability to ask NLG models to generate responses that focus on the subset of semantic information about an underlying object, where the information reflects their interests. In this thesis, we refer to these varieties of semantic information that characterize the focused object as "aspects." Considering the previous examples, price and quality, political career and legacy are the aspects of TVs and Barack Obama, respectively. NLG responses that can focus on specified aspects could allow for faster and easier information access for large audiences with diverse interests because of the ability to tailor the output to individual users.

In this context, there is an extensive literature on aspect-based NLG over customer review data for products or restaurants, where features such as "ease of use" or "price" were regarded as aspects. On these data, the task of aspect-based summarization was extensively studied both with extractive and abstractive methods (Angelidis and Lapata, 2018; Krishna and Srinivasan, 2018; Wang and Ling, 2016). The outputs generated by these models convey the specified aspects of a given subject reasonably well. Despite the success with neural network-based models above, there still remain fundamental challenges in aspect-based NLG.

(1) Aspect-based NLG tasks lack domain diversity. As described above, most aspect-based NLG tasks have focused on customer review summarization (Angelidis and Lapata, 2018) or coarse topic-based summarization (Fremann and Klementiev, 2019; Perez-Beltrachini et al., 2019) by synthesizing data. This is primarily due to the lack of suitable supervised data that contains texts with associated aspects. Considering the potential practical use of aspect-based NLG, the lack of domain diversity both limits the applicability of existing systems to certain domains, and potentially overlooks interesting scientific questions that may arise when these systems are applied to new domains. Thus, we argue that it is of great importance to model and evaluate aspect-based NLG systems in more general domains.

(2) Aspects are related in a structural way. The majority of aspect-based models have focused on an unstructured set of aspects. However, aspects themselves often exhibit structural relations with respect to each other. For example, a hierarchical taxonomy of aspects can be defined when compiling a survey of a scientific discipline, where high-level aspects are more general and low-level aspects discuss specific details. An example of high-to-low aspects that transition from general to specific content would be: "Economics of Data Pricing", " Cost Reduction in Information Goods", and "Search Costs" from a survey article in Economics (Pei, 2020). In this way, it would be advantageous for aspect-based NLG models to be able to take advantage of the structural relationships in aspects when available.

(3) Aspects influence the generated response only coarsely. When humans discuss a specific topic, we often change the granularity of discussion from general to specific content, such as only touching on a topic or diving into more fine-grained attributes of the topic or entities in relation to the topic. For example, a political discussion on Barack Obama can broadly cover his education (topic) while pinpointing his alma mater such as Columbia University and Harvard Law School. Existing aspect-based generation tasks have mostly focused on coarse aspects, which do not explicitly specify the content. This is particularly problematic in the context of NLG where a model could comply with the aspect and generate unrelated content at the same time (*e.g.* generating a text about food but unrelated to the food served by the restaurant about which the text discusses). Thus, models for fine-grained aspects are necessary to achieve fine-grained content control. Previous work on data-to-text NLG tasks also are concerned with generation of fine-grained data entries, however, their main goal is to verbalize the data in natural language and not to produce coherent text by incorporating data as the users want. Controlling which entity to focus on in summarization was also investigated previously (Fan et al., 2018), but to limited extent.

1.1 Main Contributions

We summarize the main contributions of this thesis below.

Domain-diverse Dataset for Aspect-based Summarization (Chapter 3) Aspect-based summarization is the task of generating focused summaries based on specific points of interest. Such summaries aid efficient analysis of text, such as quickly understanding reviews or opinions from different angles. However, due to large differences in the type of aspects for different domains (e.g., sentiment, product features), the development of previous models has tended to be domain-specific. In this chapter, we propose WikiAsp, a large-scale dataset for multi-domain aspect-based summarization that attempts to spur research in the direction of open-domain aspect-based summarization. Specifically, we build the dataset using Wikipedia articles from 20 different domains, using the section titles and boundaries of each article as a proxy for aspect annotation. We propose several straightforward baseline models for this task and conduct experiments on the dataset. Results highlight key challenges that existing summarization models face in this setting, such as proper pronoun handling of quoted sources and consistent explanation of time-sensitive events.

Transferring Aspect-based Generation Models Across Domains (Chapter 4) Conventionally, aspect-based summarization models are trained on data consisting of aspect-based summaries for each salient aspect to be summarized. However, it is impractical to prepare data for every domain of interest and its corresponding aspects. Therefore, in this chapter, we investigate approaches for *zero-shot transfer* for aspect-based summarization in unseen domains, where the salient aspects to be summarized may differ from any we have seen before in our training data. Transferring both domains and aspects without labeled data requires controllability of models both at domain and aspect levels, a non-trivial problem understudied in the literature. To tackle these challenges, we propose a two-stage extractive-abstractive model that encodes domains and aspects as natural language inputs to the model, which allows for handling arbitrary aspects as long as they are expressible in natural language. Experiments demonstrate the efficacy of the model on a task of generating Wikipedia article sections in new domains.

Structured Aspect-based Generation on Scientific Documents (Chapter 5) In the previous chapter, aspects were treated as sets, *i.e.* there is no assumption regarding relationships between them. While this may be a valid assumption for applications that require single or independent aspect-based summaries, real-life applications often require generation of multiple aspects in the right order, which raises the need for modeling the relationships of aspects. Modeling such relationships could help generation in different ways, such as additional aspect contexts providing more complex content control (*e.g.* promoting and demoting certain content), or encouraging more coherent summaries with respect to other summaries. To this end, in this chapter, we propose the task of generating scientific survey articles as the aspect-based summarization of related reference articles. In this task, each section of the survey articles is treated as a summary of cited reference articles with respect to particular aspects (sections) that are sequentially ordered. Following the modeling approaches used in the previous chapter, we experiment with state-of-the-art summarization models with the ability to incorporate target aspects as well as preceding aspects. We found that incorporating aspect contexts improves the individual aspect-based summary quality.

Fine-grained Aspect-based Generation (Chapter 6) So far in the previous chapters, we investigated adapting generation models to the content control at topic level with aspects. However, the topical aspects still grossly under-specify the exact content to generate, and a more detailed content specification requires a model that is capable of effectively incorporating such information. In this context, we propose Latent Relation Language Models (LRLMs), a class of language models that parameterizes the joint distribution over the words in a document and the entities that occur therein via knowledge base relations. This model has a number of attractive properties: it not only improves language modeling performance, but is also able to annotate the posterior probability of entity spans for a given text through relations. Experiments demonstrate empirical improvements over both word-based language models and a previous approach that incorporates knowledge graph information. Qualitative analysis further demonstrates the proposed model's ability to learn to predict appropriate relations in context.

1.2 Thesis Outline

The following chapter discusses definitions and background concepts for the thesis, as well as prior work. After that, Chapters 3 and 4 address the problem of lack of domain diversity in aspect-based summarization and zero-shot application based on the proposed dataset. Next, Chapter 5 tackles the problem of incorporating structures among the aspects in the form of survey article generation, then Chapter 6 changes the focus to modeling fine-grained aspects. Finally, conclusions of this thesis and future directions are discussed in Chapter 7.

Chapter 2

Background

In this chapter, we overview key background concepts used in this thesis. We start by defining the central theme of this thesis, *aspect*, in § 2.1. Previous relevant literature on aspect-based models and tasks are then discussed in § 2.2. Finally, relevant methods and problems are discussed in § 2.3.

2.1 Definition

What is an *aspect*? In natural language processing (NLP) research literature, the term is most often associated with either aspect-based sentiment analysis or aspect-based summarization. However, to the best of our knowledge, *aspect* has not clearly been defined in the previous literature. Therefore, we define *aspect*, as used in this thesis, based both on the underlying meaning of the word and based on how it has been used in early and recent literature. The commonalities of aforementioned aspect-based tasks are two-fold: (1) introduction of an additional input ("aspect") compared to non aspect-based counterparts, which (2) specifies semantic properties of a particular object in the generated response. For instance, aspect-based summarization adds an additional specification regarding which subset of the semantic information of the source text should be reflected in the output.

Based on this observation, we define an aspect as *semantic property that specifies a subset of information about an object*. A similar yet different task from aspect-based generation is query-focused summarization (QFS). QFS also takes an additional input "query" that specifies the summary to be the answer constructed from the source text, but queries to the same text may not be associated with the common object.

Similarly, aspect-based generation is defined as the task of generating texts while reflect-ing the specified aspect. In other words, this is a controllable text generation task, where the intended control is on the content of the text.

2.2 Prior Work

We first introduce and discuss the precursor to aspect, *focus*, in § 2.2.1. Next, § 2.2.2 covers the use of aspects in text summarization tasks, which is the primary instance of aspect-based generation. Then, related similar tasks in the NLP community are briefly discussed in § 2.2.3.

2.2.1 Precursor to Aspect: Focus

Early literature on natural language generation (NLG) and dialogue understanding discussed a similar concept to aspect: *focus of attention*, which was divided into *global focus* (Grosz, 1977) and *immediate focus* (McKeown, 1983, 1992). In the context of dialogue, global focus narrows speaker's and listener's knowledge about the world to a relevant part to achieve a discourse goal efficiently, while immediate focus refers to narrowing "to a single object (or set of objects) in its pool of relevant information" (McKeown, 1983) at the utterance-level. Because these studies targeted dialogues, the primary unit of focus was the entities (*e.g.* person, item). In this thesis, however, we generalize aspects with different granularity levels through the aforementioned definition, which allows us to study different concepts in a unified manner.

2.2.2 Aspect-based Summarization

Aspect-based generation was first introduced as "feature-based summarization" on customer reviews (Hu and Liu, 2004), where the primary objective is to allow users to focus on the parts they are interested in thanks to summaries about particular product features. Similarly, different terms were used to describe the same concept in customer review analysis literature, such as keywords (Gamon et al., 2005), fine-grained features (Popescu and Etzioni, 2007), or topical facets (Mei et al., 2007). Later, while not a generation task, Snyder and Barzilay (2007) introduced the term *aspect*, referring to a similar concept to the "feature" in the previous study. Following these works, Titov and McDonald (2008) unified the inconsistent terminology into *aspect* and defined the aspect-based summarization task, where the authors learned an aspect-aware extractive summarization model. Text summarization is a suitable task for aspect-based generation, because generating text about certain aspects of a longer text would most likely compress the original text in terms of length, achieving the summarization. Thus, a major subfield of aspect-based generation is aspect-based summarization. We first summarize previous work on aspect-based summarization and discuss other important studies.

To carry out the task, related works on aspect-based summarization developed datasets in review domains, such as reviewing e-commerce sellers (Lu et al., 2009), movies (Wang and Ling, 2016) or Amazon product reviews (Angelidis and Lapata, 2018; Yang et al., 2018). As evident from the listed datasets, customer reviews are suitable resource for aspect-based generation, because reviews most likely involve (1) the same underlying object (*e.g.* product, movie, etc.) and (2) people writing about different properties and attributes of the object consistently. Relaxing the definition of aspect, multiple studies (Fremann and Klementiev, 2019; Krishna and Srinivasan, 2018) have synthesized aspect-based summarization datasets on the news-wire domain, where each sample is composed of concatenated multi-topic data samples from CNN/DailyMail (Nallapati et al., 2016). TAC 2010 also held a shared task of guided-based summarization on news domain, which resembles aspect-based summarization in terms of topic guidance. Thus, data resources have mainly been developed in either customer reviews or in synthesized news domains, thereby limiting the utility of the trained models beyond those datasets. This thesis aims to address this issue with two approaches, one by constructing a domain-rich aspect-based summarization dataset (Chapter 3), and the other by learning a model that can operate across domains in a zero-shot manner (Chapter 4).

Early attempts at neural aspect-based summarization took extractive approaches. For example, Angelidis and Lapata (2018) proposed a ranking-based extractive model that uses extracted aspect-relevant phrases (with polarity estimation). Neural abstractive models take advantage of sequence-to-sequence models (Sutskever et al., 2014) by incorporating target aspect information in various ways, such as feature embeddings (Fremann and Klementiev, 2019; Kikuchi et al., 2016; Krishna and Srinivasan, 2018; Michel and Neubig, 2018), extra contexts (Li et al., 2018), auxiliary objectives (Perez-Beltrachini et al., 2019), or by prompting (Fan et al., 2018). These methods took advantage of recent advances in neural summarization methods (Cheng and Lapata, 2016; Chopra et al., 2016; Gehrmann et al., 2018; Paulus et al., 2018; Rush et al., 2015; See et al., 2017). More recently with pre-trained language models, previous attempts have shown that prompting is one of the most effective methods for controlling the outputs (Dathathri et al., 2020; He et al., 2020; Keskar et al., 2019; Raffel et al., 2020). Adding aspect information only at test time by replacing a model component with aspect-specific features has also been studied (Amplayo and Lapata, 2021). We point out that models described above consider aspects as additional labels accompanied with instances, and did not consider the relationships between aspects. In addition, aspects introduced in the aforementioned models only loosely constrain the outputs in a way that the topic is consistent (*i.e.* "the summary should talk about *aspect*"). One of the goals in this thesis is to model relationships that aspects possess, which have been either overlooked or unavailable (Chapter 5). Also, we aim to achieve finer-grained aspect-based generation in Chapter 6.

2.2.3 Query-based Generation Models

In § 2.1, we discussed that aspects serve as the signal to control the content with the constraint of an underlying object. While not exactly fitting to the aforementioned definition, there are a few highly related research topics to aspect-based generation: query-focused summarization and question answering. We briefly describe the tasks and the relevance to aspect-based generation below.

- **Query-focused Summarization**. Originally introduced at DUC 2005 (Dang, 2005), this task's objective is to generate a summary according to queries. In other words, aspect-based generation conforms to this definition by considering the aspects as the queries. However, there is no restriction on what and how the queries are formulated, unlike aspect-based generation where the aspect should be the semantic properties of the underlying object. While differing in definition, much of the methodology can be translated across tasks, such as re-ranking the source texts according to aspect (query) relevance (Su et al., 2020). Much focus is on the relevance assessment between queries and text units, which is followed by the summarization step. We refer the readers to (Xu and Lapata, 2020) for more related work on query-focused summarization.
- **Question Answering**. This task is different from aspect-based generation in the same way that query-focused summarization is. The input and output of this task are similar to that of query-focused summarization; it takes as input source texts and a query, and returns the output. The key difference lies in the intended output, where question answering focuses on getting the most probable answer, while summarization tasks focus on getting all relevant information within a token budget (Reddy et al., 2019; Xu and Lapata, 2020).

2.3 Relevant Concepts

In this section, we summarize common concepts and problem settings tackled in this thesis. § 2.3.1 and § 2.3.2 describe key methods and problems, respectively.

2.3.1 Methods

This thesis leverages recent advances in deep neural network-based models for NLP. Specifically, all of the chapters in this thesis employ Transformers (Vaswani et al., 2017) as building blocks. Besides, Transformer-based architectures employed in this thesis are mostly pre-trained; the parameters are optimized for pre-training objectives on a large amount of data before use. We cover the key ideas of these methods below.

Sequence Modeling with Transformers Sequence modeling is a central part of NLP, where the objective is to obtain context-rich representations of a sequence. Ever since the neural network-based models have proven to be effective, recurrent neural networks (RNN; (Elman, 1990; Hochreiter and Schmidhuber, 1997)) have widely been adopted to various NLP tasks including NLG. In this context, Vaswani et al. (2017) introduced the Transformer, a neural sequence model composed of layers of multi-head self-attention and multi-layer perceptron blocks. With the direct access to any position in the sequence, Transformers have taken the place of RNNs and are now the standard method for featurizing sequences. The originally proposed Transformer architecture suffers from a computational bottleneck as the sequence becomes longer. To mitigate this, a number of methods have been proposed including efficient attention mechanisms (Beltagy et al., 2020; Kitaev et al., 2020), compressed contexts (Rae et al., 2020), and recurrent conditioning on previous contexts (Dai et al., 2019).

Pre-trained Language Models With ample amounts of data, Transformers have exhibited gains over RNNs in multiple NLP tasks, such as machine translation (Vaswani et al., 2017) or question and answering (Yu et al., 2018). However, Radford et al. (2018) (and later on Devlin et al. (2019)) showed the effectiveness of pre-training on Transformer-based architectures, where the models are pre-trained on a large-scale unlabeled corpora such as Wikipedia or CommonCrawl. With various methods to learn from unlabeled corpora such as mask filling or left-to-right token predictions, resulting pre-trained language models provide sequence representations effective for most of NLP tasks (Devlin et al., 2019; Lewis et al., 2020). We refer the details to (Liu et al., 2021).

Among various forms of pre-training methods, a particularly relevant architecture to this thesis is BART (Lewis et al., 2020), a pre-trained sequence-to-sequence model on Wikipedia. BART is pre-trained with the denoising autoencoding objective, which is to reconstruct the input from noised versions of it. Since BART is a sequence-to-sequence model, it has a great affinity with sequence-to-sequence generation tasks such as machine translation or summarization.

2.3.2 Problem Settings

Different parts of this thesis tackle the same or similar problem settings. Specifically, we highlight language modeling and aspect-based summarization, two major tasks discussed in this thesis below.

Language Modeling Given a sequence of n tokens: $x = x_1 x_2 \ldots x_n$, Language modeling refers to the task of estimating the probability of the sentence $P(x)$, *i.e.* the joint probability of individual tokens. The learned probability distribution or the parameters for it is called a **language model**. The joint probability is typically decomposed in a left-to-right manner with the chain rule as follows:

$$P(x) = P(x_1)P(x_2 \mid x_1) \ldots P(x_n \mid x_1 x_2 \ldots x_{n-1}). \tag{2.1}$$

With respect to neural sequence models introduced above, each of the factored probabilities (*e.g.* $P(x_1), P(x_2 \mid x_1)$) is calculated to represent the sequence probability as a whole. To train these models on a corpus, the likelihood over the training portion of the corpus is maximized.

A variant of language models that is more relevant to this thesis is **conditional language models**. Conditional language modeling concerns with learning a probability distribution of a sequence given conditional information: $P(x \mid C)$. The conditioning variable C could be of a variety of forms such as another sequence, structured data, or indicators, depending on the down-stream task.

Aspect-based Summarization Given a source document S and a target aspect a, aspect-based summarization aims to learn a summarization model that generates an aspect-based summary t_a. An aspect-based summary is required to comply with the content specified by aspect a. In the context of neural network-based models, summarization tasks are mostly modeled as sequence-to-sequence learning where the source and the target are source document and summary, respectively. Specifically, aspect-based summarization models typically maximize the conditional probability $P(t_a \mid S, a)$, which corresponds to the conditional language modeling formulation: t_a is equivalent to x, and $\langle S, a \rangle$ serve as the conditioning variable C.

Chapter 3

Multi-domain Large-scale Aspect-based Summarization Dataset

In this and the next chapter, our focus will be on aspect-based NLG at a conventional granularity, where aspects specify topical properties of a given domain. Aspect-based NLG tasks have historically focused on a small number of domains such as product reviews due to the lack of natural data with aspects. Because of that, the aspects for such domains tend to be similar and lacking diversity. We address this challenge by proposing a new dataset WikiAsp for multi-domain aspect-based summarization which consists of a diverse set of aspects. Specifically, we cast Wikipedia articles as the aspect-based description of an entity where different sections corresponds to the aspects of the entity. Adopting the idea that the article text can be considered a summary of cited sources (Liu et al., 2018a), we view our task as aspect-based summarization of these cited sources. We collect 20 diverse domains with each of which containing 10 salient aspects. For this task, we experiment with a two-stage model that first clusters input documents according to potential aspects to be included and summarizes for each aspect. While the proposed model was able to generate aspect-sensitive summaries, automatic evaluation results and the analyses indicate that certain domains exhibit unique challenges.

The content in this chapter has been reported in the following work:

- Hiroaki Hayashi, Prashant Budania, Peng Wang, Chris Ackerson, Raj Neervannan, Graham Neubig. WikiAsp: A Dataset for Multi-domain Aspect-based Summarization. Transactions of the Association for Computational Linguistics 2021; 9 211–225.

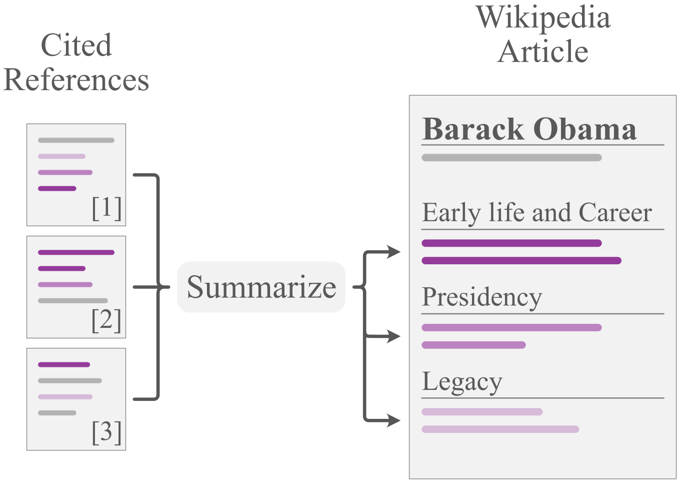


Figure 3.1: In WikiAsp, given reference documents cited by a target article, a summarization model must produce targeted aspect-based summaries that correspond to sections.

3.1 Overview

To tackle aspect-based summarization, a number of datasets have been proposed, but they are somewhat narrowly focused. For example a great majority of the work focuses specifically on the domain of product or restaurant reviews. In contrast, generic summarization models are tested on a much wider variety of genres, from news (Grusky et al., 2018; Nallapati et al., 2016), to academic papers (Kang et al., 2018; Kedzie et al., 2018), to movie scripts (Gorinski and Lapata, 2015). For each genre, the types and characteristics of aspects that will need to be touched upon in a good summary will differ greatly.

One natural source of such multi-domain articles is Wikipedia, and the section boundaries and titles in each article form natural annotations of aspects and corresponding text. There have recently been a number of attempts to generate the *lead* section of Wikipedia articles from the linked external sites in the reference section (Fan et al., 2019; Liu et al., 2018a; Liu and Lapata, 2019a), an approach that does not explicitly consider the different aspects covered by the article. Perez-Beltrachini et al. (2019) also examine domain differences in Wikipedia text summarization. However, existing datasets and analyses lack structure, broad domain coverage, or both. We argue that (1) generating *structured* summaries is of inherent interest, as these will allow humans consuming the information to browse specific aspects of interest more readily, and (2) the structure will *vary across domains*, with different domains demonstrating very different characteristics.

In this chapter, we construct a dataset for multi-domain aspect-based summarization that allows us to train models for this unique variety of summarization task, and examine the challenges posed therein. Figure 3.1 illustrates the overview of our task. Specifically, we turn to *section titles* of Wikipedia articles and construct sets of "aspects" through steps of automatic extraction, curation, and filtering. The section texts then serve as corresponding aspect-based summaries.

We devise a baseline two-stage method consisting of aspect identification and summarization using extractive and abstractive models, and conduct experiments on the proposed dataset. The analysis of experimental results and the generated summaries reveals the unique challenges posed by our multi-domain and multi-document setting. For example, aspects that require summarizing contents in a particular order (*e.g.*, time series events) in a multi-document setting adds extra difficulty because of the need for correctly ordering scattered (and possibly duplicate) pieces of information from different sources. Certain domains that involve interviews or quotes of people also exhibit challenges in correctly modifying pronouns based on the relationship to the topic of interest.

3.2 Related Work

Wikipedia as a Summarization Dataset

Wikipedia has been studied as a target resource for generation. An early attempt on generating *full* Wikipedia articles relied on web search results for target entities as inputs (Sauper and Barzilay, 2009), which simulates an authoring process of humans searching information over the Internet. Liu et al. (2018a) formulate a sub-task of generating *lead* sections as summarization of reference web pages to target articles. The resulting WikiSum dataset is accompanied by rich metadata about articles and inspired different uses of the dataset (Perez-Beltrachini et al., 2019). Our work also builds upon the WikiSum dataset, and aims to evaluate aspect-based summarization models using different sections from Wikipedia articles. Similarly to our work, WikiRef (Zhu et al., 2019) use section titles as queries to formulate a query-based summarization task that aims to summarize cited references into citation statements. Compared to Sauper and Barzilay (2009), our dataset is an order of magnitude larger, both in the amount of articles and in the number of domains covered.

Title: Barack Obama
Aspect: *Early life and Career*
Obama was born on August 4, 1961, at Kapiolani Medical Center for Women and Children in Honolulu, Hawaii. . . .
Aspect: *Presidency*
The inauguration of Barack Obama as the 44th President took place on January 20, 2009. In his first few days in office, Obama issued . . .
Aspect: *Legacy*
Obama's most significant legacy is generally considered to be the Patient Protection and Affordable Care Act (PPACA), . . .

Table 3.1: Example Wikipedia article about Barack Obama. Our goal is to generate texts given the cited references and the specified aspects.

Multi-Document Summarization

Extractive methods have shown effective for multi-document summarization in previous work (Cao et al., 2015; Nenkova et al., 2006; Yasunaga et al., 2017; Zhang et al., 2018), but abstractive methods have increasingly adopted for the task (Fabbri et al., 2019; Lebanoff et al., 2018). Our task is based on the idea of (Liu et al., 2018a) which treats references as source documents for the multi-document summarization task, and we experimented with both types of summarization models in our experiments.

3.3 Generating Wikipedia as Aspect-based Summarization

Wikipedia articles exhibit a specific way of organizing information about a focused topic. An article S consists of two parts: section titles a, and their contents p. The contents are further split into sections, where each section describes information about the main topic from different viewpoints. Table 3.1 shows an example article about the topic "Barack Obama", with several sections "Early life and Career," "Presidency," and "Legacy". In practice, the contents included in each section can take many forms, from text, tables, and images, to more specialized content such as brackets of a tournament. In this chapter, we focus only on sections that mainly consist of textual content (see Section 3.4 for how we define this).

Importantly, the content in Wikipedia articles is required to be *verifiable*: "other people using the encyclopedia can check that the information comes from a reliable source".[1] To ensure this, articles contain citations from a set of references $\mathcal{R}$ so that readers can check the validity of the content. In other words, citations supposedly contain the majority of the information written in the articles. Liu et al. (2018a) took advantage of this fact by proposing a summarization task using cited references as source documents for summarization. Citations include published material (such as books) and websites, but because only web-based citations can easily and automatically be mined via crawling, we consider only web-based citations as source documents in this chapter and ignore the rest of non-web based citations following Liu et al. (2018a).

The goal of our task is to learn a model $f : \mathcal{R} \rightarrow S$, which can 1) identify and gather information from cited references and 2) generate a section-by-section summary where each section contains the appropriate type of information. Formally, let $\mathcal{R} = \{R_1, R_2, \dots, R_M\}$ be a collection of M cited references for an article $S = \{s_1, s_2, \dots, s_N\}$ of N sections. Each section s_i is essentially a tuple of a section title and one or more paragraphs: $s_i = \langle a_i, p_i \rangle$.

While there is a fair amount of variety in section titles across different articles, articles that belong to the same domain tend to share aspects that are particularly salient for that domain. Because of this, we select a fixed-size subset of all section titles that appear in each domain as the set of aspects $\mathcal{A}$ that we will target; details on how we select this subset will be elucidated in the following section. Hence, our task is cast as multi-document aspect-based summarization.

3.4 The WikiAsp Dataset

In this section, we describe our concrete steps to create our dataset.

3.4.1 Data Collection

As the base data, we build upon the data collection strategy from the WikiSum dataset (Liu et al., 2018a), a dataset for generating lead sections of Wikipedia from referenced web pages. Following the WikiSum data generation script,[2] we first crawled cited references covered by CommonCrawl for each Wikipedia article. We then recover all the sections[3] of the target Wikipedia

[1] https://en.wikipedia.org/wiki/Wikipedia:Verifiability

[2] Tensor2tensor's WikiSum generator was used.

[3] Due to the design of WikiSum dataset, the first section title of any article is automatically renamed to "LEAD". Therefore, we could not recover first sections of the Wikipedia articles. We suggest editing the data generation scripts for future WikiSum users if section title information is necessary.

Infrastructure		Software	
history	13293	reception	8196
route description	5627	gameplay	8095
facilities	2792	development	3983
services	1955	plot	3697
future	784	history	2465
route	689	features	1799
location	613	story	991
construction	577	release	750
connections	497	overview	570
description	463	legacy	564

Table 3.2: Frequency of filtered aspects that are *textual* in 2 domains. Due to space constraint, the statistics for the rest of domains will be available in the Appendix A.3.

articles from WikiSum (which was unused in WikiSum dataset) and obtain pairs of (section title, section paragraph). An example for this is shown in Table 3.1.

3.4.2 Domain Separation

Articles in different domains focus on different salient topics, as observed by Perez-Beltrachini et al. (2019). For example, the "discography" section is common for articles about singers, but is not appropriate for articles about infrastructure. To characterize such structural differences, we separate the set of articles obtained in the previous step into sets in particular domains. Specifically, we follow Perez-Beltrachini et al. (2019) in assigning one category for each article using DBPedia (Auer et al., 2007). DBPedia stores structured information for each Wikipedia article, including the domain labels and info boxes. Additionally, it defines a topical hierarchy of the domains (ontology classes). We first map between articles and the domain labels from the corresponding DBPedia dump. Obtained domain labels, however, have mixed granularity (e.g., Person and its sub-class Dancer) which causes imbalance in the number of examples in each domain, as well as domain overlap between high-level and low-level domains in the domain hierarchy. We mitigate this by recursively merging domains at leaf-level into coarser ones according to the aforementioned topical hierarchy from the ontology classes.[4] We repeat the merging procedure until a branch in the hierarchy includes more than 15,000 articles, and picked 20 domains at the leaf of the merged hierarchy.[5]

[4] http://mappings.dbpedia.org/server/ontology/classes/

[5] Many articles are labeled directly as Person, in which case the domain is high-level at the hierarchy. We do not select this domain because lower-level domains such as Artist or SoccerPlayer already have enough number

Dataset	Domain	#Dom.	#Train	Doc. Length	Sum. Length	#Asp.	#Asp./Ex.
OpoSum	Product Review	6	359,048	138	49	9	2.00
Amazon	Product Review	7	240,000	82	-	-	-
RottenTomatoes	Movie Review	1	2,458	2369	24	*2	*1.00
MA-News	News	1	284,701	1350	54	6	2.98
WikiAsp	Encyclopedia	20	320,272	13,672	213	10	1.77

Table 3.3: Training set statistics comparisons against previous aspect-based summarization datasets. For multi-domain datasets, the sum of all the examples are reported. #Asp./Ex. represents the average number of aspects that a model has to summarize on each example. (* Review saliency is treated as aspects. #Asp. represents the number of aspects per domain if the number of domains is more than one. Compared datasets are the work of Angelidis and Lapata (2018); Frermann and Klementiev (2019); Wang and Ling (2016); Yang et al. (2018), respectively.

3.4.3 Aspect Selection

Next, we perform aspect selection on each set of articles in the domains extracted during the previous step. As previously noted, articles in the same domain tend to share similar set of section titles. Motivated by this observation, we construct the set of aspects from the **most frequent section titles**.

From the frequency distribution of section titles in a domain, we manually filter ones that are not *textual*, that is, more than half portion of section consists of text. For each section title, we take 20 randomly sampled sections and include it in the set of aspects only if 80% of samples consist of *textual* paragraphs. Following the steps above, we construct the 10 most frequent aspects for each domain. However, the choice of words in section titles vary depending on the editors within the same domain, which leads to missing relevant aspects that are moderately frequent but not present in Top-10. For example, one of the common section titles in WrittenWork domain are "summary" and "plot summary," which should be merged together to form a single aspect. We handle these cases by inspecting the frequent distribution further down and manually identifying semantically equivalent titles to merge.

The resulting dataset consists of instances in 20 domains where each domain has 10 predefined aspect classes. We show statistics comparisons of the dataset to existing aspect-based summarization datasets in Table 3.3 and examples of obtained aspects for two domains in Table 3.2.

Appendix A.1 and A.3 summarizes the data size for each domain and the obtained aspects

articles.

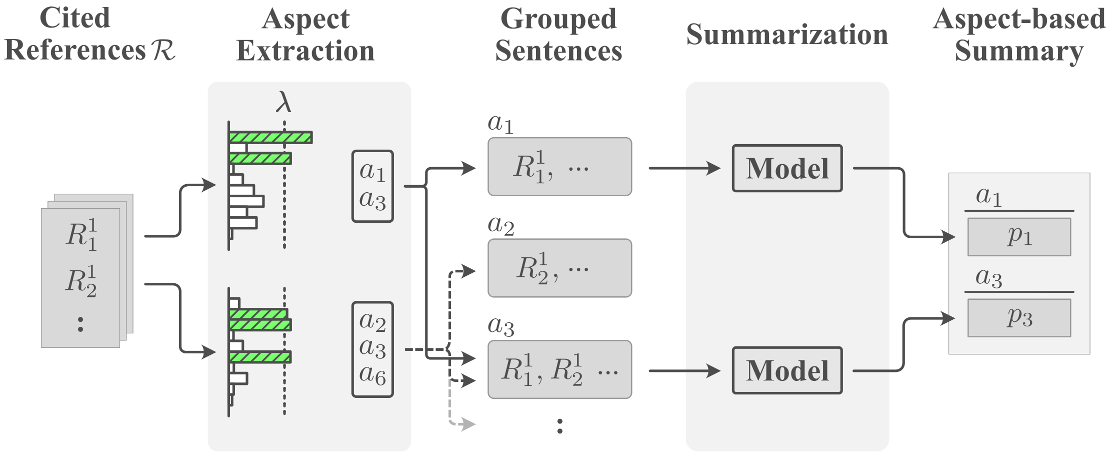


Figure 3.2: Two-stage model diagram. The aspect classifier assigns aspect labels for each reference sentence R_j^i from references $\mathcal{R}$ with a threshold λ. Sentences are then grouped according to the assigned labels, which are fed to the summarization model. Groups about irrelevant aspects (*i.e.*, a_2) is ignored. Finally, the summarization model outputs summaries for each relevant aspect.

for the rest of 18 domains respectively.

3.5 Baseline Models

Next, in this section we describe two baseline models for solving this task. Both of these models decompose the overall process into two stages: aspect discovery and aspect-based summarization of classified sentences. Both baseline models share the same methodology for aspect discovery, but differ in terms of summarization models. The model overview is shown in Figure 3.2.

3.5.1 Aspect Discovery

The first stage consists of labeling sentences in *cited reference texts* according to aspects. Having training data that contains sentences in the reference documents labeled with target aspects would be the ideal case, but these do not exist *a priori*. Therefore, we instead create training data by assigning each sentence in the target articles with aspect labels corresponding to the aspect to which the sentence belongs. For example, the article about Barack Obama in Table 3.1 yields training instances consisting of sentences labeled with *Early life and career*, *Presidency* and *Legacy* depending on which paragraph a sentence comes from. This data makes it possible to train a classifier that learns to predict aspects from the texts at sentence-level. At test time, cited reference sentences are fed into the learned classifier and are labeled with their most likely aspects.

However, the discrepancy of inputs at train/test time is problematic because the model is not exposed to any *noisy* sentences that do not belong to any of the relevant aspects at training time, while cited reference texts do contain such sentences. For example, an article in the *Company* domain may have a citation to the company website itself, which contains commercial messages that may not be appropriate in encyclopedic text such as Wikipedia. We manage such cases by introducing an auxiliary label *Other* at training time and let the model learn to identify noisy sentences as well. To do so, sentences labeled with *Other* are randomly sampled from texts in different domains and added to training data. We fine-tune the pre-trained ROBERTa (Liu et al., 2019) model on this classification dataset for each domain. Logits obtained from the model are then passed through the sigmoid function to obtain probabilities of each aspect for a given sentence. Finally, we assign labels to a sentence by taking the aspects a_i whose probabilities are greater than the threshold λ: $P(a_i) > \lambda$. The lower we set the threshold, the more but potentially noisy sentences we include as the input to the summarization model. We tune λ independently for each domain based on the performance on validation sets and set 0.5 for *Group*, 0.8 for *Album*, *Animal*, *Building*, *Film*, and 0.9 for the remaining domains as the threshold values.

3.5.2 Summarization

Sentences that are labeled with the same aspect are then grouped in order of occurrence in cited references to form a chunked paragraph that discusses the same aspect. This forms aspect-based clusters of relevant sentences, which become the input to a summarization model. On the contrary, aspects that are never labeled (due to low probabilities) are deemed irrelevant and thus will not be summarized. We consider both an extractive and an abstractive summarization model in our baseline implementation. For the extractive model, we use TextRank (Barrios et al., 2016; Mihalcea and Tarau, 2004), a graph-based ranking model for extracting important sentences. For the abstractive model, we use PreSumm (Liu and Lapata, 2019b), a Transformer-based summarizer with fine-tuned BERT as the source encoder. For each domain, PreSumm is fine-tuned and trained on the pairs of (grouped sentences, target aspect paragraph) to learn to produce summaries given the aspect-relevant sentences.

3.6 Evaluation

We evaluate models along two axes: aspect discovery and summarization. We note that the primary task in this dataset is aspect-based summarization, thus aspect discovery evaluation discussed below is only for diagnostic purposes. Since the aspect sets differ in different domains, evaluation is performed separately for each domain.

Aspect Discovery Models have to correctly predict the right set of aspects about which they generate summaries. The aspect discovery criterion aims to evaluate the similarity between the set of aspects about which a model decides to generate summaries and the set of aspects that appear in the target article.[6] For comparing these two sets, we use precision, recall and F1 scores.

Aspect-based Summarization Gold standard summaries only exist for each of the aspects that appear in an article. Therefore in this evaluation, we focus on evaluating the model's ability to summarize inputs particularly on these aspects. Specifically, generated summaries are paired to corresponding reference summaries with the same aspects and are evaluated using ROUGE (Lin, 2004). Since ROUGE is a recall-based measure, the number of tokens in the model outputs directly affect the performance. Controlling the length is particularly important for our dataset because average summary length for each aspect in different domains varies (*e.g.*, "description" and "location" from HistoricPlace domain has 396 and 90 average tokens, respectively). We take this into account by explicitly setting the maximum number of words for extractive and abstractive summaries to be the average number of words in the target summaries in the training set for each aspect and for each domain.

3.7 Experiments

We provide two baseline models for the task and evaluate on the proposed dataset.

[6] Note that there are two potential reasons an aspect does not appear in the target article: (1) it may not be appropriate for that particular entity (e.g. the "controversy" aspect in the "company" domain should not exist if that company has legitimately never had a controversy), or (2) the article may not be complete. For this evaluation, we make the simplifying assumption that all articles are complete and thus missing aspects are an indication of failure to recall information, but relaxing this assumption in some way may result in more accurate evaluation.

3.7.1 Implementation Details

For aspect classification, we used `roberta-base`[7] model and fine-tuned for 5 epochs on the created surrogate dataset above for each domain, with the learning rate 2×10^{-5}. For the extractive summarization, we specify the summary length for TextRank according to the mean length of target summaries for each aspect in each domain. We re-train the PreSumm summarizer on our dataset for each domain: the encoder is initialized with the weights of pre-trained BERT (Devlin et al., 2019) and the decoder is trained from scratch. The total number of training steps is 300,000. For some domains, we further tuned the decoder dropout rate to 0.3 to stabilize training. At inference time, we specify maximum summary lengths for each *aspect* for each domain using the average summary lengths from computed from the training set.

3.7.2 Results

In this section, we discuss the experimental results on each stage.

Aspect Discovery

We show the aspect discovery results in Table 3.4. We see a general trend of high recall predictions made by the model. While varying thresholds could balance precision and recall, the results exhibited high recall after hyperparameter search. This suggests that the learned classifier is poorly calibrated. Class imbalance also plays a role here; predicting the major classes give high recall due to skew aspect frequency distributions. Among others, the classifier performed best with the Town domain by achieving the highest precision and the F1 score.

Summarization

The automatic evaluation results are shown in Table 3.5. Neither baseline unanimously outperformed the other on all domains, but we observe that PreSumm (abstractive) performs better than TextRank (extractive) on average.The low R-2 and R-L scores by both models despite the oracle being relatively higher suggest that important phrases to be summarized do not appear rarely.[8]

To understand the upper-bound of model performance for the task, we also show summarization results of the extractive oracle model in Table 3.5. Sentences were chosen directly

[7] We used Hugging Face's implementation (Wolf et al., 2019) for obtaining and fine-tuning the weights.

[8] Note that TextRank connects nodes according to content overlap, thus isolated sentences are not selected.

Domain	Prec	Rec	F-1
Album	19.64	86.43	30.64
Animal	34.69	84.08	45.52
Artist	26.32	75.24	36.72
Building	31.46	91.25	42.92
Company	28.97	91.50	41.06
EducationalInstitution	25.64	93.82	37.66
Event	28.99	96.44	42.36
Film	32.84	91.46	45.17
Group	17.46	95.56	28.18
HistoricPlace	33.38	90.22	42.98
Infrastructure	28.38	94.00	41.00
MeanOfTransportation	23.24	83.13	33.88
OfficeHolder	21.22	73.25	30.62
Plant	31.25	83.17	42.10
Single	25.36	88.33	37.16
SoccerPlayer	28.54	67.18	37.16
Software	31.52	94.65	45.10
TelevisionShow	20.44	81.76	31.28
Town	42.61	71.85	50.12
WrittenWork	21.50	94.29	33.71

Table 3.4: Aspect discovery results on the test set.

from cited reference texts to maximize the ROUGE score against summaries, thus bypassing the aspect classification stage. The oracle performance shows that a summarization model can indeed perform competitively on the dataset if the model is given with the full input information. The contrasting results between the oracle and two stage models suggests the importance of accurate content selection before performing summarization.

3.8 Analysis

We discuss the model outputs and analysis below.

3.8.1 Aspect-by-aspect Evaluation

Not all the aspects are equally hard to summarize; some might require summarization of a broad range of information, while others require only specific concepts to be summarized. We further investigate this by looking into summarization performance for both models on per-aspect

	TextRank			PreSumm			Extractive Oracle		
	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L
Album	19.56	2.81	17.26	22.76	6.31	20.27	37.72	12.58	33.19
Animal	18.00	3.16	16.05	27.11	8.08	25.01	34.82	10.52	31.01
Artist	17.22	2.49	15.58	21.79	3.76	20.00	41.49	15.04	37.64
Building	23.91	4.96	21.85	24.99	5.97	23.24	41.95	14.31	38.28
Company	22.92	3.70	20.65	22.28	4.08	20.50	40.20	12.30	36.16
EducationalInstitution	21.47	4.29	19.24	24.17	6.70	21.96	39.11	14.04	35.18
Event	26.64	5.67	24.08	28.31	7.69	26.20	46.17	16.90	41.87
Film	21.25	3.81	19.14	20.58	5.34	18.86	40.24	13.78	36.14
Group	22.30	3.62	20.20	25.51	4.97	23.51	41.36	13.23	37.56
HistoricPlace	18.96	3.71	17.51	27.40	8.08	25.69	37.78	10.83	34.65
Infrastructure	20.40	3.27	18.39	27.86	9.24	25.80	36.04	10.00	32.25
MeanOfTransportation	21.20	3.93	19.31	24.52	7.04	22.72	41.13	13.70	37.45
OfficeHolder	18.45	3.15	16.77	19.63	5.24	18.12	39.60	14.70	36.04
Plant	18.73	3.02	16.84	25.29	6.30	23.20	34.93	9.66	31.31
Single	17.96	2.67	15.86	22.06	6.78	19.98	36.51	11.57	31.88
SoccerPlayer	14.79	2.36	12.89	12.89	1.86	12.05	31.06	8.00	27.08
Software	24.54	4.56	22.05	20.51	5.15	18.82	42.79	13.96	38.30
TelevisionShow	19.77	3.21	17.68	19.20	3.53	17.42	40.35	13.47	35.67
Town	17.89	3.56	16.50	19.76	4.39	16.87	33.21	10.31	30.70
WrittenWork	23.39	3.89	21.14	22.19	4.33	20.15	42.66	13.93	38.16
AVG	20.47	3.59	18.45	22.94	5.74	21.02	38.95	12.64	35.03

Table 3.5: Aspect-based summarization results on the test set. The last row shows the average performance.

basis. Table 3.6 shows the best-performing aspects sorted in descending order by ROUGE-1 scores for two summarization models on the validation set. Through manual investigation of the generated samples for each aspect, we observed that the aspects where the *abstractive* model performed well tend to have common templates and similar choice of vocabulary, more so than other aspects. For example, 58% (out of 183 samples) of the target summaries for *government* in Town shared the identical summaries despite the fact that articles discuss different townships. Similar but less prevalent patterns were observed in other aspects as well.

Aspects where the *extractive* summarization model performed better contain much larger numbers of tokens in the summaries than average. Specifically, the average summary length for 10 aspects where TextRank performed the best was 303, while that for 10 aspects where PreSumm performed the best was 166. Naturally, abstractive models have issues with maintaining coherence over long decoding results, but the extractive model has few issues gathering relevant sentences at the cost of incoherent transitions from sentence to sentence. As for

Dom.	Aspect	PreSumm	TextRank
		↓ R-1	R-1
Tow.	government	**55.10**	21.20
Eve.	format	44.94	24.73
Inf.	facilities	42.46	14.75
Bui.	exterior	41.81	25.60
Mea.	background	39.00	23.72
His.	heritage listing	36.58	10.25
Ani.	habitat	32.91	12.95
Pla.	taxonomy and nm.	32.70	9.39
Edu.	rankings	31.80	26.92
Alb.	commercial perf.	31.71	15.51
Dom.	Aspect	R-1	↓ R-1
Eve.	battle	28.00	**32.00**
Eve.	report	24.77	30.11
Sof.	gameplay	24.17	28.53
Eve.	background	30.01	27.42
Eve.	aftermath	27.54	27.27
Bui.	history	25.32	27.13
Sof.	plot	20.50	27.00
Edu.	rankings	31.80	26.92
Wri.	plot summary	22.08	26.85
Fil.	plot	19.43	26.66

Table 3.6: List of aspects sorted in descending order of ROUGE-1 score according to PreSumm (top half) and TextRank (bottom half). "performance" and "naming" are abbreviated to "perf." and "nm.", respectively. Domain names shortened to the first three letters.

the content, extractive summaries exhibited the advantage of being able to correctly include mentions related to numbers and dates.

3.8.2 Quality of Generated Summaries

We then examined the generated summaries from the two models and compared them qualitatively. Samples are shown[9] in Table 3.7 from some of the domains listed in Table 3.2.

Manual inspection of the generated summaries revealed pros and cons of the two models:

- **Both models are successful at discussing on-topic content.** For all the summaries inspected, both models were able to generate on-topic content in spite of the source documents potentially being noisy.

[9] Samples from other domains are in Appendix A.2.

Domain / Title: Software / Cyberpunk 2077
Aspect: *Gameplay*
Gold: cyberpunk 2077 is a role - playing video game played from either a first - person or third - person perspective . it is set in an open world metropolis called night city . the game will feature non - english speaking characters . players who do not speak the languages can buy translator implants to better comprehend them ; . . . **Ext.**: cyberpunk 2077 takes place in , you guessed it , the year 2077 . for just a few hours , you can be rich , successful , and popular with your preferred gender . cyberpunk 2077 will be a aaa rpg , but whether it will come to resemble the witcher in any way remains unclear . how braindances will be used by or on the protagonist is . . . **Abs.**: the game is an action role - playing game in which the player assumes the role of an unnamed protagonist , runabout who lives in a large metropolis known as the “ black city “ . the game ’ s narrative is told through procedurally generated procedurally generated . during the course of the game , the player is briefed into . . .
Domain / Title: Event / Battle of the Saintes
Aspect: *Aftermath*
Gold: the british lost 243 killed and 816 wounded , and two captains out of 36 were killed . the french loss in killed and wounded has never been stated , but six captains out of 30 were killed . it is estimated that the french may have lost as much as 3 , 000 , and more than 5 , 000 french soldiers and sailors were captured , . . . **Ext.**: battle of the saintes , (april 9 – 12 , 1782) , in the american revolution , major naval victory for britain in the west indies that restored british naval mastery in the area and ended the french threat to nearby british possessions . the valiant was subsequently present at the relief of gibraltar on 12 april 1781 . . . **Abs.**: the immediate aftermath of the battle of the niles meant the british fleet could not be able to deliver a great strategic victory in the kingdom of france , and had also declared war on great britain with the royal navy to france . the outcome was the subject of a strategic victory for the british . in the meantime , the british navy had captured the french navy . . .
Aspect: *Battle*
Gold: on 9 april 1782 , the copper - hulled british fleet soon caught up with the french , who were surprised by their speed . de grasse ordered the french convoy to head into guadeloupe for repair , forcing him to escort two fifty - gun ships (fier and experiment) and placing his fleet in line of battle in order to cover the retreat . . . . **Ext.**: " after some initial maneuvers and minor clashes , a full - scale battle was joined on april 12 , by which time the british had thirty - six ships of the line in action against thirty french ones . he turned his ships ninety degrees and sailed through the broken french line of battle , splitting the french line into four segments . in doing this , the guns on each side of the british ships were brought to bear on the french with little risk of return fire . . . . **Abs.**: the battle started as the shock . the battle progressed , when the british forces reached the north - eastern flank of the battle of weidman (kingdom) in a battle . he had begun to return to the field , and moved his forces toward the mouth of the river . in the battle , the first contingent of the french navy ships got off from a small contingent of british soldiers as well as the third - rate , under the command of general sir henry sturgis . . . .

Table 3.7: Generated summaries from multiple domains. Ext. and Abs. represent summaries from TextRank and PreSumm.

- **Abstractive summaries underperform at generating exact entity mentions.** Almost all the samples require generation of entities because the task targets at generating encyclopedic texts. Except for the title (topic) entity, abstractive models either generated no entities or wrong ones.

3.8.3 Aspect Classification Accuracy

We observed a general trend of low precision for aspect discovery. We hypothesize that this is due to limited target aspects for each article; correctly extracted aspects affect negatively to precision if they do not exist in the target article. To quantify this, 10 random articles are selected from the validation set in Software domain. For each article, we extract 10 sentences

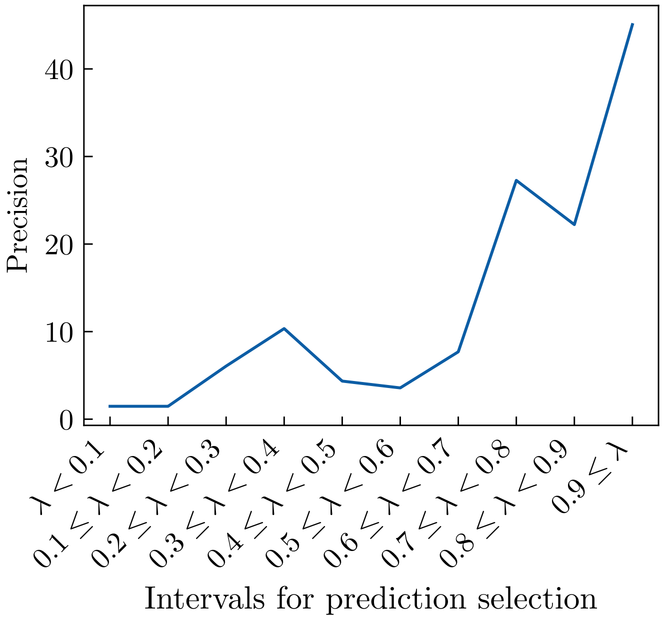


Figure 3.3: Precision differences in varying threshold ranges.

labeled with the highest confidence for each of the 10 aspects, resulting in 1,000 sentences in total. Each sentence is annotated with binary labels indicating whether it is correctly associated with the aspect or not.[10] With the threshold λ set to 0.9, we achieved the precision of 45.1, which shows that the aspect discovery has the ability to extract aspects, but not as good at extracting *relevant* aspects for the article. We observed that the model predictions tend to be polarized to extreme values (*i.e.*, near 0 or 1). We also show the relationship between λ ranges and the precision in Figure 3.3, which indicates that the classifier is not well-calibrated.

3.8.4 Domain-specific Challenges

One of the benefits of having many domains for the same task is to be able to characterize the differences and challenges that are unique to certain domains. We analyzed the generated summaries from both of the summarization models and identified some of them below.

Pronoun Resolution for Opinion-based Inputs

This is particularly important in domains and aspects with subjective reviews such as music(Album, Artist, Group, and Single) or Software. Source documents in these domains often

[10] Sometimes, the entity in discussion by the sentence is not clear. In this case, we annotate it correct if the sentence could correspond to the target aspect of any entity.

include quotes by artists or critics, which are often written from different person perspective. These are usually converted by the Wikipedia editors into more encyclopedic text, citing the source of the information and writing in the third person. By design, extractive summaries have issues with this problem because of the lack of ability to transform the input sentences in any way. For example, the first extractive summary in Table 3.7 describes a game in a subjective way. We verified this by randomly selecting 20 summaries for *gameplay* aspect in Software domain. We inspected pronouns in extractive summaries and mark ones with first- or second-person pronouns if the gold summaries do not contain them. We found 65% of the samples contained those undesirable pronouns that do not align with the format of gold summaries.

Chronological Explanation

This variety of content is often found in certain aspects such as *history* and *event*, which tend to appear across multiple domains but are most prevalent in Event, HistoricPlace, and non-human entities like Company and Building. It is essential in these aspects to describe key information in the right chronological order for better readability. This would not be a hard task for single document summarization, as the model could perform reasonably by following the order of the original document. However, since our input is of multi-document form, maintaining chronological order when aggregating information across multiple domains becomes non-trivial. Indeed, neither of the models were successful at being truthful to the order even when there are enough clues in the original references. For example, multiple sentences start with “In [year], . . .”, but the generated summary jumps around in time. We randomly picked 20 samples of extractive summaries with *history* aspect from Company domain and found that 25% of the samples have inconsistent timeline explanations.

Chapter 4

Zero-shot Model Adaptation on Unseen Domains and Aspects

We have learned from the previous chapter that a supervised aspect-based summarization models perform reasonably well. However, as with any model trained in a supervised fashion, the trained models will fail to generate high quality summaries on out-of-domain samples. To assist human editors in such cases, we utilize WikiAsp and propose a transferable aspect-based summarization model that learns from collections of training domains and can take arbitrary aspects at test time to generate summaries. Specifically, we follow a similar approach to the baseline models in the previous chapter and propose a two-stage model consisting of (1) a zero-shot aspect classification and (2) aspect-based summarization.

The content in this chapter is written in:

- Hiroaki Hayashi, Pengfei Liu, Graham Neubig. ZerASum: Zero-shot Domain Transfer for Aspect-based Summarization. (*Under review*)

4.1 Overview

A model trained on aspect-based summarization data might generate summaries for the aspects in the data. However, in many cases there is text from multiple potential domains that needs to be summarized, and the salient aspects may vary from domain to domain. For example, Wikipedia articles cover domains including `Software` and `Animal`, and the things a reader may want to know about software (*e.g.* about its "development") and animals (*e.g.* "habitat") are very different. Gathering training data for every possible domain to train aspect-based

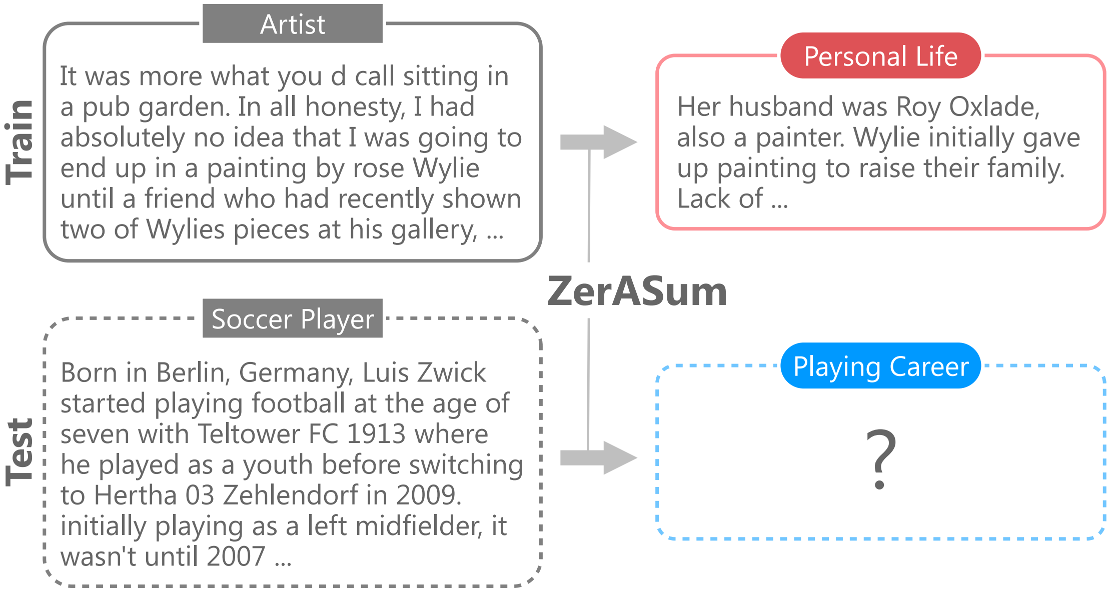

Train
Artist
It was more what you d call sitting in a pub garden. In all honesty, I had absolutely no idea that I was going to end up in a painting by rose Wylie until a friend who had recently shown two of Wylies pieces at his gallery, ...
Personal Life
Her husband was Roy Oxlade, also a painter. Wylie initially gave up painting to raise their family. Lack of ...
ZerASum
Test
Soccer Player
Born in Berlin, Germany, Luis Zwick started playing football at the age of seven with Teltower FC 1913 where he played as a youth before switching to Hertha 03 Zehlendorf in 2009. initially playing as a left midfielder, it wasn't until 2007 ...
Playing Career
?

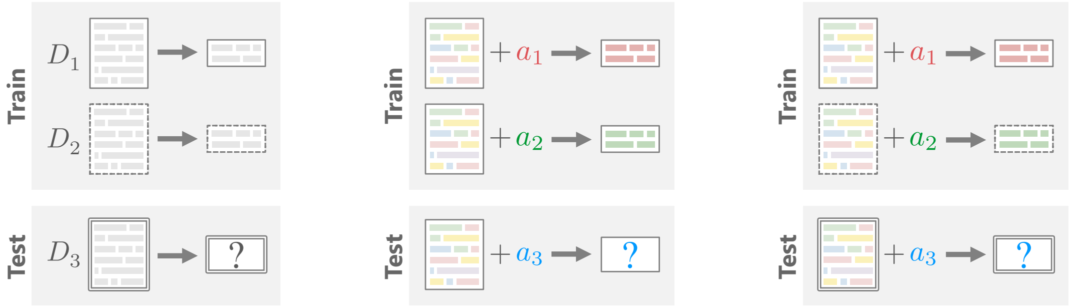
Figure 4.1: ZerASum learns an aspect-based summarization that can be transferred across domains.

summarization models on respective salient aspects is infeasible.

To address this issue, previous attempts studied transfer learning over domains (Brown et al., 2020; Fabbri et al., 2020; Wang et al., 2019; Zhang et al., 2019b) or aspects (Tan et al., 2020) separately. However handling both has been unexplored, for which two challenges may account: (i) the lack of suitable resources that contain different domains and aspects at the same time, and (ii) the lack of effective approaches to allow for transfer across domains and aspects.

Given this background, in this chapter, we investigate the problem of transferring aspect-based summarization models across domains in a zero-shot manner, overcoming the aforementioned challenges. To address the first, we develop a zero-shot transfer setting from the WikiAsp dataset introduced in Chapter 3, a multi-domain aspect-based summarization dataset that contains 20 domains with 10 aspects each from English Wikipedia. To address the second, we incorporate **domain** and **aspect** as encodable features, and study the ability of models to summarize aspects they've never seen in domains they were not trained on.

To tackle this task, we adopt a two-stage model (Chen and Bansal, 2018; Liu et al., 2018a) consisting of a cascade of extractive and abstractive sub-models. The two-stage architecture is beneficial for two reasons: (i) it makes it possible for the model to filter irrelevant content explicitly at the extractive stage, and (ii) it allows for processing of long source documents that would otherwise be too expensive for a single neural abstractive model. We experiment with unsupervised PacSum (Zheng and Lapata, 2019) and supervised BART (Lewis et al., 2020)-based



(a) Domain Transfer

(b) Aspect Transfer

(c) Domain & Aspect Transfer

Figure 4.2: Different settings of zero-shot transfer for text summarization. Square borders represent domains.

models for the extractive stage. For the abstractive stage, we extend BART by incorporating domain and aspect information in several different ways. Experimental results show that incorporating domain and aspect information as textual prompts results in the best performance, achieving reasonable transfer accuracy compared to best and worst case scenarios. Ablation studies revealed that of the two stages, incorporating these signals in the abstractive stage has the largest effect. Analyses also show that the best model still suffers from over-generalization for some aspects; summaries for some aspects discussed content from training domains despite different inputs.

4.2 Problem Setting

We illustrate the transfer over different domains and aspects in Fig. 4.2. One avenue of zero-shot transfer for summarization (Fig. 4.2a) focuses on transferring the models over multiple *domains* using multi-task learning (Wang et al., 2019), demonstrations (Zhang et al., 2019b), or intermediate fine-tuning on auxiliary corpus (Fabbri et al., 2020). In terms of adaptation on new *aspects* (Fig. 4.2b), previous work on abstractive aspect-based summarization constructed weakly-supervised data that covers a wide range of entity-based aspects in news domain, where the model fine-tuned on such data can accept arbitrary (entity-based) aspects (Tan et al., 2020). The problem setting we consider involves two variables that may differ at test time, domains and aspects, where both variables could interact with each other (Fig. 4.2c).

In this study, we consider the problem of *transferring an aspect-based summarization model across domains*. Let $\mathcal{D} = \{D_1, D_2, \ldots, D_N\}$ be the set of N domains, and $A^i = \{a^i_1, a^i_2, \ldots, a^i_K\}$ be the set of aspects for D_i. We aim to learn a summarization model $f : \langle S, a \rangle \rightarrow t_a$ that takes

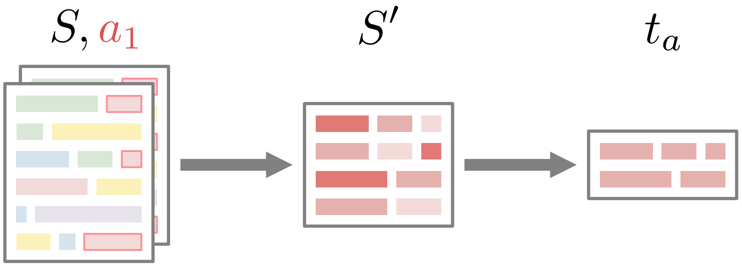


Figure 4.3: Two-stage model, where the source documents are summarized into an aspect-aware intermediate document, then further summarized into the aspect-based summary.

source documents S and the target aspect a as input and generate an aspect-based summary t_a. The model can only observe samples from a subset of domains $\mathcal{D}_{\text{train}}$ at training time, and is evaluated on another disjoint subset $\mathcal{D}_{\text{test}}$ ($\mathcal{D}_{\text{train}} \cap \mathcal{D}_{\text{test}} = \varnothing$).

4.3 Methodology for ZerASum

In this section, we describe our methodology to solve the problem of zero-shot transfer. To start, we decompose the aspect-based summarization task into two sub-tasks, (i) stage one: *extraction of aspect-relevant content* and (ii) stage two: *summarization with extracted content.* Fig. 4.3 presents the overview of such a two-stage model architecture. Instead of directly mapping the source text S to the aspect-based summary t_a, ZerASum creates an intermediate document S' which will be the input to stage 2.

The potential advantages are: (1) by reducing the large input in the first stage's extraction, it becomes more feasible to use compute- and memory-intensive models for abstractive summarization in the second stage. (2) By task decomposition, the challenge of adapting generation models to unseen domains and aspects are broken down into two sub-problems that can be discussed flexibly and relatively independently in their respective sub-tasks. Next, we will show how two sub-tasks are formulated and problems are addressed there.

4.3.1 Extraction as Sentence Entailment

The challenge in the extraction stage is to identify textual information from a long document based on given domains and aspects that are not seen in the training set. To address this problem, we present a novel formulation of *extractive aspect-based summarization as entailment.* We also introduce aspect-agnostic unsupervised extractive methods for comparison and for more

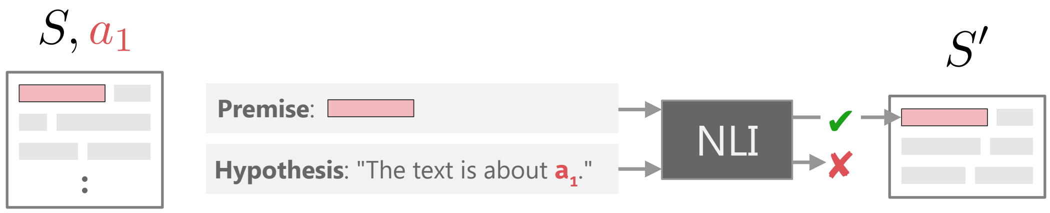


Figure 4.4: Under the NLI formulation, each sentence in the source text and an aspect a are converted into a premise and a hypothesis. If the NLI predicts entailment, the sentence is incorporated into the extracted summary.

application scenarios.

Supervised

Based on the resemblance to the zero-shot text classification, we adopt the natural language inference (NLI) paradigm for zero-shot text classification (Yin et al., 2019). Given a sentences pair: premise and hypothesis, NLI is defined as a classification task that predicts whether a premise entails the hypothesis or not. Yin et al. (2019) showed that formulating the hypothesis with a statement discussing the relevance to a label allows for the use of NLI as a text classifier. We show the overview in Fig. 4.4. For a given sentence s and the aspect a to evaluate, the premise p is directly the sentence s and the hypothesis is a natural language sentence that encodes a. For example, the simplest form could be “The text is related to a”. Through preliminary experiments, we found that providing the model with richer hypothesis representations improve the classification accuracy. Specifically, we follow (Zhang et al., 2019a) and compose the hypothesis with 1) aspect name, 2) aspect synonyms derived from WordNet (Miller, 1995), and 3) abstract-concept terms using ConceptNet (Speer et al., 2017), respectively. The *entailment* prediction from the model is then associated to the sentence (premise) being labeled with the aspect a.

Since training data for this model will not be naturally available from aspect-annotated datasets, we propose a method to construct it by applying labels to source document sentences. For each sentence in the source document,

- An **entailment** example is created if the sentence has high similarity to one of aspect-based summaries for the instance. The corresponding aspect is assigned to the sentence as a label.
- A **non-entailment** example is created in two ways: wrong-label and irrelevant. For each entailment-class instance, a random aspect (different from the entailed one) is randomly sampled and added as non-entailment example. An irrelevant example is sampled from

sentences with low similarity to any of the target summaries.

We employed ROUGE-2 > 0.1 as the threshold for similarity assessment. At inference time, the trained model is applied on all the (source sentence, aspect) candidate pairs, which are then grouped by aspects and are sorted in the descending order of entailment probability.

The number of tokens for extraction is set by the maximum number of tokens that the stage 2 model can take as input. If entailment-predicted sentences do not fill the token budget, we further add borderline not-entailment sentences according the same order. We then re-sort the selected sentences in the original order of sentence appearance to retain some semblance of discourse consistency.

Unsupervised

A popular and light-weight approach for extractive summarization is to employ an unsupervised approach. We also consider two flavors of unsupervised methods.

PacSum (Zheng and Lapata, 2019) is an unsupervised extractive summarization algorithm, which refined the graph-based ranking algorithm with BERT-based representations and directed edges. Specifically, we employed PacSum over sentences featurized with TF-IDF vectors, a more light-weight variant compared to using BERT for sentence representations. Note that this method does not concern with any specified target aspects, but instead returns salient sentences in general. We leave a more elaborate alternative (*e.g.* query-based unsupervised extractive summarization) as a future work.

LEAD In a similar spirit, we also adopt the leading text baseline which simply takes the first part of the source documents.

4.3.2 Summarization with Guidance

The challenge in this stage is how to design an abstractive summarization system that can make good use of: (i) extracted content from stage one (ii) domain and aspect labels associated the documents.

Specifically, after the first stage, the source text S is shortened to S_a, an aspect-aware source document. We fine-tune a pre-trained sequence-to-sequence summarization model on the pairs of shortened document S_a and target aspect-based summary t_a. To guide the model to generate outputs about the target aspect, we consider different methods to consider domain and aspect information in the model.

Prompt Prompting is a simple yet effective approach to condition the sequence models on a variable-length control signals. Prefixing the signals to control the outputs has been successfully adopted in controllable text generation tasks (Dathathri et al., 2020; He et al., 2020; Keskar et al., 2019). We follow this intuition and modify the input to the model by prefixing the domain and aspect names as the prompt to the encoder[1] so that the model will condition the generation on these variables. We keep the prompt in a simplest form, *i.e.* `[domain] [aspect] [source]`, as we did not observe critical differences when employing different prompt designs.

Logit Intuitively, domain and aspect information influences the vocabulary choices of the output, and thus could be a useful signal to directly control the vocabulary distribution. To do so, we follow Michel and Neubig (2018) and incorporate domain and aspect information into the calculation of logits. Specifically, we directly modify the projection layer of the decoder with additional terms that represents domain and aspect information:

$$P(\mathbf{y}_i \,|\, \mathbf{x}) = \sigma(\mathbf{W}[\mathbf{h}_i \oplus \mathbf{e}_d \oplus \mathbf{e}_a] + \mathbf{b}_h), \tag{4.1}$$

where $\oplus$ denotes the concatenation operation. $\mathbf{W}$ and $\mathbf{b}_h$ are learnable parameters, $\mathbf{h}_i$ is the hidden representation at i-th decoding step, $\mathbf{e}_d$ and $\mathbf{e}_a$ are the domain and aspect embeddings, respectively. We represent domains using the definitions mined from Wikidata and DBPedia, while the aspects are represented in the same way as the featurization of hypotheses for the NLI model.

4.4 Zero-shot WikiAsp

In order to evaluate zero-shot transfer of aspect-based summarization, we need an appropriate data setting of a multi-domain and multi-aspect corpus. To test aspect-based summarization models in this setting, we adopt WikiAsp, a multi-domain aspect-based summarization dataset in the encyclopedic domain from Chapter 3. WikiAsp is built upon WikiSum (Liu et al., 2018a), a dataset that formulated cited references of a Wikipedia article as source text and the lead section

[1] Preliminary experiments found prefixing the decoded sequence less effective than doing so for the encoder.

Name	Stats
Domains	20
Aspects per domain	10
Examples	320,272
Avg. Source Tokens	13,672
Avg. Summary Tokens	213
Avg. Aspects per example	1.77

Table 4.1: WikiAsp training set statistics.

of it as the summary. Cited references are sorted by (Liu et al., 2018a) in descending order according to the TF-IDF score of each document against the article title and concatenated into single text. Notably, WikiAsp considers other sections as the target text as well and categorizes articles into distinct domains defined in DBPedia (Auer et al., 2007), which make it a multi-domain aspect-based summarization dataset. Data statistics are shown in Tab. 4.1.

In Chapter 3, we developed a two-stage model consisting of aspect classification and summarization, where both parts are trained *separately for each domain*. While this approach allows for learning models specific to the domains of interest, models are not applicable to a new domain without fine-tuning due to over-fitting on the trained domains.

We randomly split 20 domains in WikiAsp into 80 : 20% split, resulting in 16 training and 4 test domains. In this split, 58% of aspects (23 out of 40) from test domains also appear in the training domains. We use combined training splits from $\mathcal{D}_{\text{train}}$ for training and combined test splits from $\mathcal{D}_{\text{test}}$ for evaluation.

4.5 Experiments

4.5.1 Evaluation Methodology

We evaluate the model performance at each stage on metrics that focus on different factors. Fig. 4.5 visualizes evaluation measures employed for comparing different parts of the model.

Stage 1 As illustrated in Fig.4.5, we adopt two evaluation methods for stage one models.

- **F1**: To measure the quality of aspect classification, we evaluate the model via F1 score for the positive class. This corresponds to the *entailment* class F1 score for the NLI model. For the unsupervised baselines, we set the number of sentences to extract as 50% of the

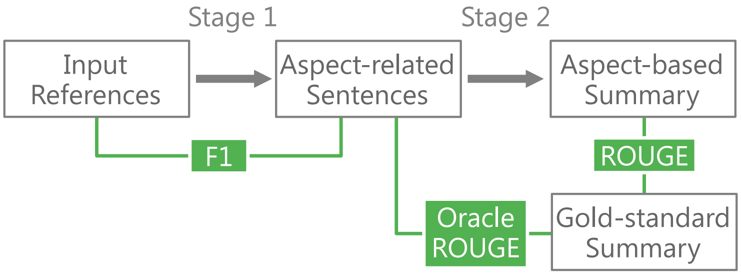


Figure 4.5: Illustration of our evaluation methodologies at different stages.

original input. We simulate that selected sentences are relevant to all the aspect candidates.

- **Oracle ROUGE (OR)**: The primary role of the first stage is to provide as informative inputs for the stage 2 as possible. We calculate oracle ROUGE (OR) scores to evaluate greedily selected extractive oracle sentences against the target summary for all the aspect-based summaries. Oracle ROUGE characterizes the upper-bound performance of stage 2 models in an extractive setting.

Stage 2 We evaluate the model performance by ROUGE, which is a primary metric for our task.

4.5.2 Implementation Details

Stage 1 We found through preliminary experiments that directly adopting the training configurations explained in (Yin et al., 2019) led to suboptimal accuracy. To remedy this, we employ 1) a larger pre-trained model (`bart-large-mnli`), 2) keep the original classification head with three labels (entailment, neutral, not-entailment) with a mask to suppress neutral prediction, and 3) scale the losses for different classes proportionally to the number of instances. As we mention below, we extract up to 1024 tokens.

Stage 2 We use BART-large (Lewis et al., 2020) as the base model, which is capable of processing up to 1024 source tokens. We start fine-tuning on a pre-trained checkpoint on CNNDM dataset (Nallapati et al., 2016) and fine-tune for three epochs. Due to not knowing the target summary size for test domains and different aspects tend to vary in the average summary length in WikiAsp, we decode using beam search with the beam size 4, the maximum number of tokens 512, length penalty 2.0, and trigram blocking.

More details are available in the Appendix.

Stage 1	Stage 2	Stage 1					Stage 2			
		F-1	OR-1	OR-2	OR-3	OR-L	R-1	R-2	R-3	R-L
LEAD	Prompt	39.10	31.41	7.73	2.62	27.96	25.26	6.12	1.87	23.30
	Logit						20.62	3.81	0.99	19.05
PacSum	Prompt	36.75	30.71	7.22	2.37	27.40	25.08	5.89	1.77	23.15
	Logit						20.21	3.62	0.93	18.76
NLI†	Prompt	**54.98**	**34.22**	**9.30**	**3.36**	**30.63**	**25.41**	**6.26**	**1.91**	**23.49**
	Logit						22.47	4.67	1.33	20.69

Table 4.2: Stage 1 and Stage 2 results. Largest scores for each column are bold-faced. † indicates results of NLI-based models are significant ($p < 0.05$) better than Lead and PacSum except the result 1.91 (R-3 in stage two).

4.5.3 Results

We report experimental results below. Paired-bootstrapping is used to assess statistical significance (Koehn, 2004).

Stage 1 In this stage, we compare aforementioned PacSum, LEAD, and NLI. LEAD serves as the baseline where it takes the first 1024 tokens from the highest-ranked documents. The results for both F1 and OR is shown in Tab. 4.2.

As a more powerful model, NLI out-performed the others both in terms of F1 and OR despite the fact that it only considered sentence-level context. We also found that the entailment probabilities reflected the likelihood that the model was actually correct. On the other hand, F1 of LEAD and PacSum are expectedly low due to inability to distinguish aspect relevance. Relatively high OR despite this aspect insensitivity suggests that 1) keywords useful for summarization are already included lead portion of the source texts and 2) generally salient texts are also salient for aspect-specific content partly because the entity of interest remains the same. While simple, LEAD out-performed PacSum in terms of OR. We suspect that this is due to the data bias introduced by Liu et al. (2018a) where high TF-IDF documents are ordered earlier.

Stage 2 Next, we experimented two stage 2 variants, Prompt and Logit, on the stage 1 model outputs mentioned above. Since each variant can follow either of the three stage 1 baselines, we obtain the total of 6 combinations. The results for ROUGE scores computed against the gold standard aspect-based summaries are shown in the right half of Tab. 4.2. While being statistically significant, ROUGE score differences at stage 2 among baselines are much small

Control		Stage 2		
Stage 1	Stage 2	R-1	R-2	R-L
✓*	✓*	30.83	9.59	28.36
✓	✓	25.41	6.26	23.49
-	✓	25.08	5.89	23.15
✓	-	23.37	5.02	21.48
-	-	20.15	3.72	18.67

Table 4.3: Comparison of Prompt against different data and control settings. For ceiling performance, we include a supervised setting (marked in $*$) that the model is trained on data from the training and testing domains.

compared to OR at stage 1. We perform an in-depth analysis comparing the models per aspect in Section 4.6.3.

Comparing the Prompt and Logit, we found that Prompt consistently out-performed. In other words, even though Logit model directly influences logits with domain and aspect information, rich feature interaction at token-level using via prompting utilized the features better.

4.6 Discussion

We discuss in-depth analyses below.

4.6.1 Feasibility of Zero-shot Transfer

To better understand the significance of the results, we compare NLI-Prompt against the best and worst case scenario: (i) supervised setting, where the model observes the training splits from the test domains as well during the training, and (ii) no-control setting, where the model does not rely on any transferable control signals. We employed NLI-Prompt for (i), and PacSum followed by vanilla BART with no domain or aspect information for (ii). We show the ROUGE score comparison on stage 2 in Tab. 4.3. While there still exists a large gap between the best model and supervised setting (5 point ROUGE-1), we observe significant increase in ROUGE scores as domain and aspect signals are incorporated into different stages.

4.6.2 Controllability at Different Stages

At which stage do we obtain the most benefit out of control signals? To answer this question, we compare the models according to the application of domain/aspect controls at different stages.

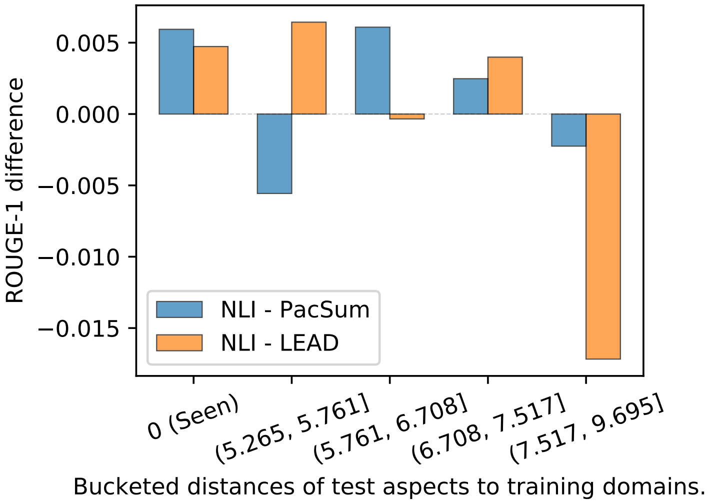


Figure 4.6: ROUGE-1 differences between NLI and other models according to aspect distances. Per-aspect ROUGE-1 scores in each bucket are averaged.

We treat PacSum as the baseline without control signals for stage 1, and vanilla BART for stage 2. Tab. 4.3 shows the ROUGE scores when employing control signals in each stage. In terms of individual contributions, we observe that enriching stage 2 with domain and aspect features clearly has an advantage in terms of compared to stage 1 with control. Combined together, we observe that the two stage are not taking the advantage of each other well; having both controlled does not improve over the baseline with only stage 2 being controlled.

4.6.3 Influence of Stage 1 on Per-aspect Generalization

58% of aspects in the test domains also appear in the training domains. This suggest that the model could capture domain-agnostic features for particular aspects and re-use them at test time. To analyze the effectiveness on observing aspects during training, we calculate per-aspect ROUGE scores of model variants on test domains by focusing on stage 1 model variants and fix stage 2 with Prompt. We also group aspects into different groups based on the distance to training domains:

$$\text{dist}(a, \mathcal{D}_{\text{train}}) = \begin{cases} 0 & \text{seen} \\ \min\limits_{a' \in \mathcal{D}_{\text{train}}} ||e_a - e_{a'}|| & \text{unseen} \end{cases}, \tag{4.2}$$

Error Type	Stage 1	
	NLI	PacSum
Topic	6	10
Domain	18	8
Aspect	5	11
Length	13	12
Misc.	8	9

Table 4.4: Error type counts on generated summaries. Prompt is used for stage 2.

where e_a, $e_{a'}$ are embeddings[2] of the aspect a. Intuitively, distant aspects from any aspects in training domains $\mathcal{D}_{\text{train}}$ is harder to adapt due to the lack of exposure to the model at training time. We then bucket the nonzero distance scores further into quartiles and calculate the average ROUGE-1 *differences* between NLI and other baselines in each bucket. The result is shown in Fig. 4.6.

NLI's advantage is most visible on seen aspects (leftmost in Fig. 4.6) thanks to observing the same aspect (with different domains) at training time. For unseen aspects, we observe a weak trend of decreasing NLI performance as they become more distant, which is overtaken by LEAD and PacSum.

4.6.4 Qualitative Analysis

While we found that the stage 2 models perform similar in terms of ROUGE, we also observed that the summaries are not necessarily similar. To gain an insight on the characteristics and influence of stage 1 on the final summaries, we conduct an error analysis by classifying poor summaries into the following error types:

- **Topic:** discusses the right domain and aspect as specified, but deviates from the main topic.
- **Domain:** discusses the right aspect but does not generate target domain content.
- **Aspect:** content lies in the right domain but does not discuss the target aspect.
- **Length:** discusses the right topic, domain, and aspect, but has a large gap in summary length (short or long).
- **Miscellaneous:** None of above.

We took 50 summaries from two model variants each (NLI-Prompt and PacSum-Prompt) that

[2] The sum of GloVe (Pennington et al., 2014) embeddings is used to compute the representation of an aspect.

Error	Reference	Summary
Topic	found mostly in the lowlands , chamaeleo laevigatus lives throughout much of sub - saharan africa	PacSum: galago demidoffii is found in the eastern cape province of south australia . it is also found in new south wales , queensland , tasmania , and victoria . . . .
Domain	toulouse has represented the united states as a member of the united states women ' s national under - 23 soccer team .	NLI: toulouse began her career as a model in the early 1990s . she appeared on the cover of the french edition of vogue magazine in 1994 . . . .
Aspect	approximately 50 % of the gray catbird ' s diet is fruit and berries . they also eat mealworms , earthworms , beetles , and other bugs	PacSum: the gray catbird is known for mew - like song , which is reminiscent of the " mew " made by a cat . they can even mimic other birds , tree frogs , and other mechanical sounds that they hear . . . .
Length	bahadur is married to khin than myint.	NLI: suk bahadur roka was born in the village of jelbang in the myagdi district of western Nepal. he is married and has a son and a daughter. he was . . .

Table 4.5: Example errors, with underlines specifying the error locations. For the aspect error, the summary discusses about behavior when the target aspect is the diet.

performed below average and annotated according to the error types above. Tab. 4.4 shows the results.

We observe notable differences in error types between the two models. Using NLI for stage 1 results in more domain errors compared aspect errors; with the ability to filter irrelevant and extract aspect-relevant content, the model can focus more on the right aspect and therefore is more accurate on the aspect. While beneficial for aspects, the large portion of domain errors suggest that the model is over-generalizing domains due to the presence of aspects. Specifically, the model might be associating certain training domains to aspects more strongly, which results in coupling certain aspects to familiar training domain contents.

PacSum, on the other hand, exhibited the opposite trend. The aspect-insensitive first stage ends up with more noise included as the output of stage 1, but it prevented the model from remembering the patterns (*e.g.* “description” is associated to “Plant” because they often appear).

Both models suffer from generating the right lengths, as each aspect has different tendency in summary lengths. While we could estimate for aspects available for training time, it is generally unpredictable on test domains. Once the target lengths are set, we could employ methods to control the lengths such as (Kikuchi et al., 2016), which we leave for future work. We show primary error type examples (topic, domain, aspect) in Tab. 4.5.

4.7 Related Work

Multi-domain Summarization Most studies on neural text summarization experiment their models on multiple datasets from different domains *individually*, such as news (Nallapati et al., 2016), scientific documents (Cohan et al., 2018), patents (Sharma et al., 2019), or Wikipedia (Liu et al., 2018a). Wang et al. (2019) proposed learning techniques that leverage multi-domain instances from multiple news datasets.

Toward Few- or Zero-shot Summarization However, there has been less focus on learning models across domains. The most similar work to our work is Fabbri et al. (2020), where the authors transferred summarization models in a zero-shot manner by fine-tuning on Wikipedia texts that share a similar data bias to the target domains. Hua and Wang (2017) studied domain adaptation between news and opinion subset of NYT corpus (Sandhaus, 2008). Low-resource training or domain adaptation for summarization was tackled by leveraging templates (Magooda and Litman, 2020), span-based plausibility and salience modeling (Desai et al., 2020), or pre-trained language models (Yu et al., 2021). Compared to the studies above, our work studies transferring aspects in addition to transferring domains.

Unsupervised Abstractive Summarization In addition to extractive approaches, a growing number of works on unsupervised abstractive summarization has been proposed (Amplayo et al., 2021; Chu and Liu, 2019; Yang et al., 2020). At the sentence level, previous approaches employed autoencoders (Miao and Blunsom, 2016; Schumann, 2018) or language models Zhou and Rush (2019) to compress sentences in an unsupervised fashion. In theory these are applicable in zero-shot scenarios, the main focus for this line of work is to learn a summarization model without gold-standard summaries, rather than the application on new domains.

4.8 Implications and Future Work

Based on these results, we see that with prompting, one can transfer an aspect-based summarization model to an unseen domain and aspects to a reasonable extent. Despite this, the overall performance is far from the supervised setting and the error cases analyzed above suggest that there is a room for improvement with respect to several aspects. First, since our model does not share knowledge between two stages, the first stage cannot receive the gradient from the decoding results at the end. This may lead to suboptimal performance, which could potentially be avoided via a unified framework that shares the parameters. Second, we observed that in some cases prompt-based control was not sufficient to ensure that the target aspect was appropriately reflected in the output. Other training objectives that more explicitly encourage the model to generate text following the target aspect may help alleviate this problem. In addition, the model could further benefit from (1) better prompt design that maximizes the utility of pretraining task, or (2) modeling aspect relationships in a way that helps encourage or suppress the co-occurrence of certain aspect pairs.

Chapter 5

Language Generation with Structured Aspects

In the previous chapters, we have explored aspects as individual labels to guide summarization. In other words, there was no consideration on what other aspects of interest exist for a given source text. This lack of consideration is problematic for real-life use cases where users may have multiple different interests on the same source text. In fact, it is necessary and beneficial to incorporate relationships between aspects for better aspect-based generation. For example, knowing what aspects have been discussed prior to generating for the target aspect helps model produce summaries that are syntactically (*e.g.* pronoun usage) and semantically (*e.g.* content determination) more appropriate. To this end, we explore incorporating structures in aspects and aim to improve aspect-based generation by leveraging them.

The content in this chapter is written in:

- Hiroaki Hayashi, Pengfei Liu, Yixin Liu, Graham Neubig. Automatic Survey Generation as Aspect-based Generation. (*In Preparation.*)

5.1 Overview

While aspect-based summarization is useful when the user interest is limited to a single aspect, users may often be interested in multiple aspects. In this context, a summarization model would need to generate multiple aspect-based summaries that serve as a whole, which introduces new challenges to the summarization task, such as organization, content overlap, and so on. We call this task structured aspect-based summarization, a variant of aspect-based summarization

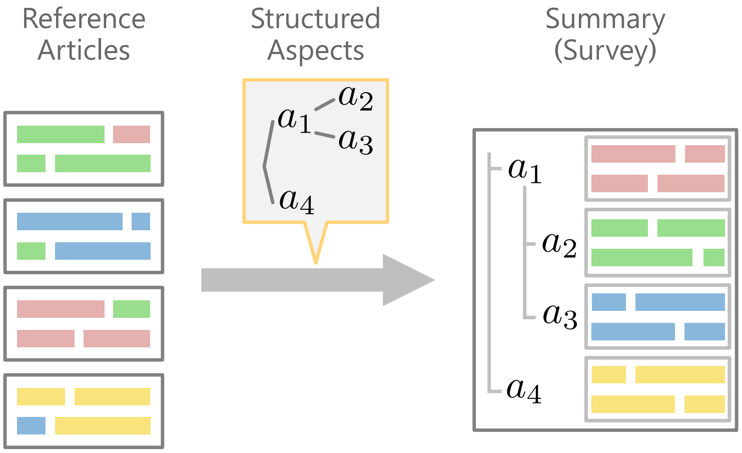


Figure 5.1: Proposed Task. References and *structured aspects* are transformed into aspect-based summaries that respect the provided structure.

where the model takes structurally organized aspects as input.

One instance of texts with structured aspects is scientific documents, where authors separate content in different parts in specific structures. For example, this chapter uses chapters, sections, sub-sections, and paragraphs to express different granularities and groups of content in a hierarchy. In the scientific document domain, previous works investigated different forms of summaries such as the abstract (Cohan et al., 2018), one-line summary (Cachola et al., 2020), or related work (Lu et al., 2020), while considering structures of the *input* (Cohan et al., 2018). However, there has been less attention on generating structured outputs, primarily due to the lack of suitable data. Perhaps the most similar idea is argumentative zoning (Teufel and Moens, 2002) which is the task to classify sentences in the abstract with key discourse structures employed in scientific documents (*e.g.* introduction, method, conclusion). Using zoning as features, previous work developed scientific document summarization systems to improve summary quality (Contractor et al., 2012).

To address the lack of resources and models for structured aspect-based generation, in this work, we reformulate the process of writing survey articles for a scientific discipline into a structured summarization task. Survey articles generally summarize the state of the research topic in a organized manner through a novel perspective, which can serve as an educational material for researchers unfamiliar with the research topic. Most importantly, a good survey provides comprehensive categorizations of the research topic into subtopic structures, in which relevant works are discussed. Previously, Jha et al. (2015a) had formulated survey generation as multi-document text summarization of reference articles, which aimed to generate a single content-guided summary. Extending this, we aim to generate the full survey article with reference articles and structured aspects. We illustrate the task in Figure 5.1.

With the goal of leveraging structured aspects for summarization, we formulate the generation of scientific survey articles as a structured aspect-based summarization task and develop a new dataset. We then propose a framework for structured aspect-based summarization and extend state-of-the-art summarization models to accommodate such information via structured prompts. Experiments show that structured aspects help achieve not only better summary quality for individual aspects but also better controllability. Analyses on the outputs also reveal that the proposed framework especially improves models on article-specific aspects (*i.e.* aspects unique to each article).

5.2 Survey as Aspect-based Summaries

To establish our task, we first define aspect-based summarization as survey generation as follows. Let a survey article D be the sequence of N section blocks, each of which consists of a section title a and section content y: $D = \{a_i, y_i\}_{i=1}^{N}$. Given a survey topic, section titles organize the topic content from different perspectives, hence they can be considered *aspects*. Each survey article includes a bibliography that compiles a set of M reference articles $R = \{R^1, R^2, \dots R^M\}$, which discuss the original content that the survey article cites and aggregates with respect to different aspects. Thus, we formulate the task of survey generation as aspect-based summarization: generate y given R with the guidance of a.

5.3 Dataset

Existing scientific document summarization datasets provide individual articles, which are then used to train a model that summarizes an article into an abstract (Cohan et al., 2018). For survey generation tasks, however, they are not directly applicable for two reasons: (1) bibliography information is not present for each paper and (2) there is no annotation as to which paper is a survey. To mitigate these issues, we construct a new dataset derived from S2ORC (Lo et al., 2020), a large-scale multi-domain scientific document dataset with rich metadata such as citation information and bibliography. Even though survey articles exist in many domains, every domain and journal likely differ in the style of the surveys. To simplify the task, we focus on computer science literature in this work. We outline the dataset construction process below.

5.3.1 Survey Article Collection

While S2ORC comes with metadata, it still does not provide the annotation about whether or not an article is a survey. To overcome the lack of such annotation, we apply a simple keyword-based filter to collect survey articles. We first conducted a preliminary study to determine keywords that are indicative of survey articles by measuring the coverage on a curated list of survey articles (Wang et al., 2021) after applying the filter. As a result, we found that having "survey" or "overview" in article *titles* is indicative regardless of research areas in Computer Science. Using this keyword-based rule, an initial set of 9025 articles were collected from Computer Science articles with PDF parses in S2ORC. However, this simple rule inevitably collects noisy entries due to its simplicity, for example "survey methods" and "aerial surveys" terms that are used with a different meanings. To improve the precision of collected articles, we further investigated noise patterns and applied a grammar-based filtering [1] on the set of articles, obtaining the final set of 4553 articles. We examined 100 random samples from these articles and verified that no non-survey articles were found.

5.3.2 Dataset Construction

Given the set of survey articles, we define each of the key variables as follows.

Input references R A natural resource to serve as the source for survey articles is the set of references cited by the surveys. For a given set of references, Jha et al. (2015a) defined the input to the survey generation task as articles that cite the references, with each article being represented by its introduction. While this approach provides other researchers' perspectives on the set of references, we observed that directly using the references lead to significantly better performance.[2] Thus in this work, we define the inputs as the references cited by the survey articles.

Survey aspect and summary a, y A survey article usually consists of multiple sections, each of which organizes subtopics. Based on the assumption that section titles can hint at such subtopics, we split each survey article into sections and treat section titles as the aspects, section

[1] For each candidate title that includes the aforementioned keywords, we only keep those whose the matched keyword is the root of a noun phrase. If the noun phrase is multi-word, we additionally filter those that do *not* include the allowed terms: "a, an, research, literature, comprehensive" to improve the precision.

[2] ROUGE-1/2 differences: 3.4/1.0.

Dataset	pairs	papers	aspects	references	source words	target words
Multi-News	44972	-	-	3	2103	264
Multi-XScience (train)	30369	-	-	4	778	116
Survey (train)	42904	3609	12	22	6238	489
Survey (valid)	5146	475	11	21	5471	528
Survey (test)	5911	469	13	22	6152	507

Table 5.1: Data statistics.

text as the aspect-based summary. Specifically, we rely on PDF parses from S2ORC, which split body of the articles into sections.Extracted aspects form a sequence in each survey article.

When human researchers write papers, they normally structure in a way such that the paper includes hierarchical section structures (*e.g.* section and sub-section). While PDF parses provide a scalable solution to obtaining the aspects, one drawback of using PDF parses is the inability to capture such complex aspect structures. For example, nested aspects (*e.g.* "2." and "2.1.") are returned as a flattened sequence. We also attempted to reconstruct structures from the sequence of aspects using textual clues like numbering, but observed that less than a third of the articles preserved such information. We leave the reconstruction of complete structures as future work.[3]

After removing pairs with low lexical overlap and non-text aspects with simple rules[4], survey articles are split into train, valid, and test splits. Table 5.1 shows statistics of the resulting dataset along with related aspect-based summarization datasets. Note that the number of samples represent the total number of pairs of aspect and aspect-based summaries, which is much larger than the total number of articles.

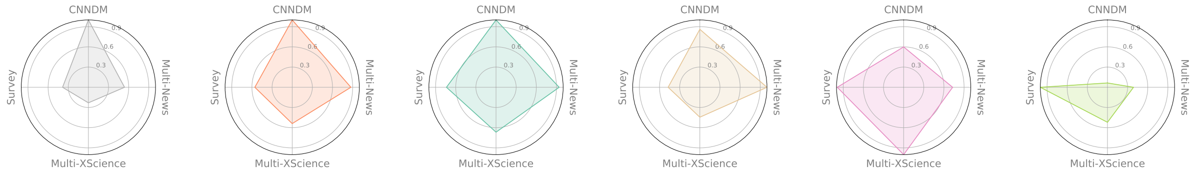
5.3.3 Data Analysis

We analyze the constructed dataset below and highlight its characteristics in comparison to the other relevant datasets.

Dataset Bias We first compare commonly used dataset statistics for our dataset against statistics from other datasets. Specifically, we compare against two multi-document summarization

[3] The updated GROBID parser for scientific articles now support the extraction of section numberings as of 8/16/2021. However, the latest S2ORC release (20200705v1) that we use does not include them yet.

[4] Non-text aspects include numbers (*e.g.* "1."), symbols, or empty strings. They consist of 1.2% of the extracted pairs.



(a) Compression. (b) Copy Length. (c) Coverage. (d) Fusion. (e) Novelty. (f) Repetition.

Figure 5.2: Spider charts for various measures for four datasets. North: CNN/DailyMail, South: Multi-XScience, East: Multi-News, West: Survey.

datasets: CNN/DailyMail (Nallapati et al., 2016), Multi-News (Fabbri et al., 2019) and Multi-XScience (Lu et al., 2020). Following (Chen et al., 2020), coverage, copy length, novelty (bigram), repetition (trigram), and sentence fusion scores are calculated. Note that these metrics can be influenced by source and summary text lengths. We refer the readers to (Chen et al., 2020) for the formal definition of individual metrics. Figure 5.2 shows the measures for four datasets. We first observe distinctive differences between news and scientific document domains. Specifically, Multi-XScience and Survey have less coverage: less copyable segments in the summaries, which is expected because related work and survey content are usually paraphrased in other words. Another characteristic is high novelty in both Multi-XScience and Survey, which indicates abstractiveness of the task. While similar in many measures, the key difference between Multi-XScience and Survey is repetition, which is the percentage of repeated trigrams in summaries. Interestingly, Survey has more than double the repetition compared to Multi-XScience. We suspect this is because related work texts (Multi-XScience) tend to be brief on each work, while Survey may discuss the same topic consistently over many sentences.

Novelty In this part, we particularly spotlight novelty, which exhibited a unique trend compared to other datasets. We plot the novel n-gram percentage up to $n = 4$ in Figure 5.3. Unlike other datasets, novel unigram percentage is as low as CNN/DailyMail (Nallapati et al., 2016), while higher-order novel n-gram percentages approach abstractive datasets such as Xsum (Narayan et al., 2018). This suggests that our task can benefit from both extractive and abstractive approaches.

General and Specific Aspects While no survey article authors follow the identical structure, there are similarities in how aspects are defined. To gain more insight, we define *general* and *specific* aspects, depending on if the aspect appears across survey articles or unique to a single article, respectively. For example, general aspects include common terms or sections such

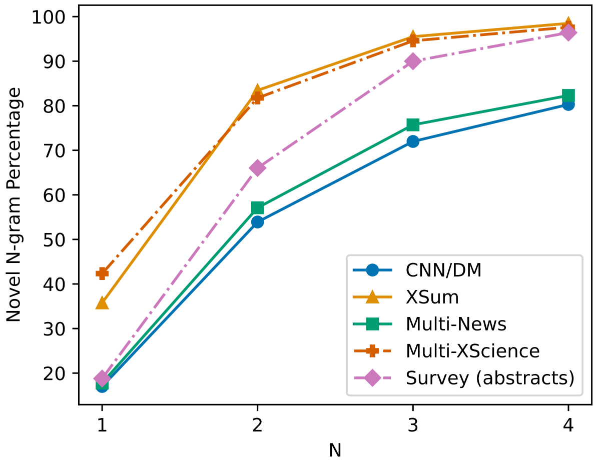


Figure 5.3: Novel n-gram percentage for different datasets. Our proposed dataset has a unique tendency of lower novel unigrams, but high novelty in higher-order ngrams.

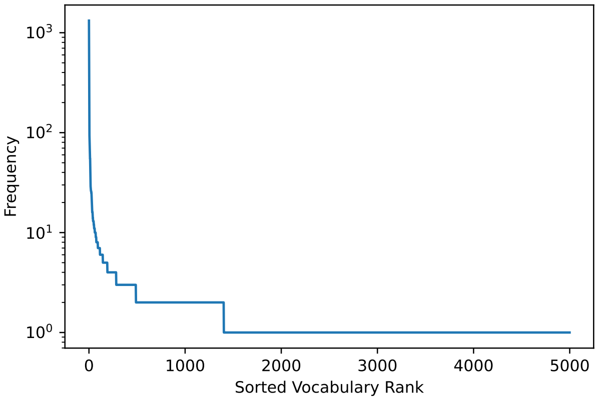


Figure 5.4: Sorted frequency distribution of top 5,000 aspects.

as “Introduction”, “Discussion” or “Conclusion”, while specific aspects include “Monte Carlo Simulation”, “What is NER?”, etc. Since there is no repetition in aspects within a single article, general and specific aspects can also be identified by the frequency of aspects. We visualize this by plotting the frequencies of the top 5,000 most common aspects out of 36288 unique aspects from the training set in Figure 5.4. The frequency distribution is highly skewed and long-tailed, with 81% of aspect types being specific, *i.e.* frequency count is 1. To perform reasonably well in this dataset, models have to (1) pick up patterns in the summaries for general aspects, as well as (2) adapt the summaries to specific aspects.

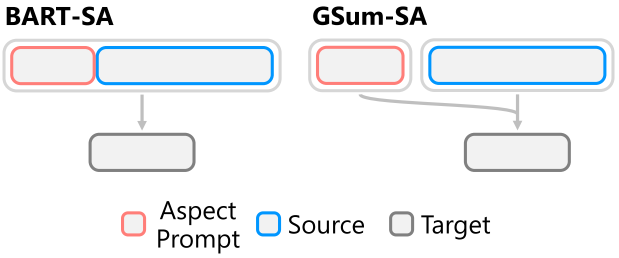


Figure 5.5: **BART-SA** encodes concatenated aspect and source, while **GSum-SA** encodes them separately as a multi-source input to the decoder.

5.4 Structured Aspect Model

Aspect-based summarization models normally only considers the target aspect when summarizing the aspect. As we analyzed in the previous section, however, aspects in survey articles are organized and structured in a meaningful manner by the authors. For example, sequential organization of aspects likely include content dependencies over the article. Thus, knowing the aspects besides the target aspects within the structure may provide clues to further guide the generated texts.

To leverage such structural information for generation, we propose to encode them as structured prompts to contextualize the target aspect with respect to the structure. Let A be structured aspects (SA) defined as $A = f\left(\{a_i\}_{i=1}^{N}\right)$, where f is a function that composes a structure around aspects for a single survey article, such as a tree or a sequence. Summarizing source reference articles R according to a target aspect a would then be:

$$\langle R, \text{prompt}(A, a)\rangle \rightarrow y, \tag{5.1}$$

where the function $\text{prompt}(A, a)$ takes the structured aspects and the target aspect as input and returns a textual prompt. For example, a prompt function for a tree-structured prompt from 5.1 and the target aspect a_3 may parse the tree in a breadth-first manner and represent as a string:

$$\text{prompt}(A, a_3) = a_1\,[\,\text{SEP}\,]\,a_4\,[\,\text{SEP}\,]\,a_2\,[\,\text{SEP}\,]\,a_3, \tag{5.2}$$

or a simple unstructured prompt function may convert to a string with the target aspect only "a_3: ". Specifically, we instantiate the prompt function as a simple concatenation with separators and A as K ($= 5$) preceding aspects in this work.

We apply this framework to two abstractive summarization models, BART (Lewis et al.,

2020) and GSum (Dou et al., 2021) to examine suitable modeling methods for structured aspects. The key difference between the two lies in how prompts interact with the source text. For BART, we prepend the prompt to the source which allows for interaction across the prompt and the source text during encoding. GSum takes the prompt as the guidance that is separately encoded, and is attended to by the decoder separately. Figure 5.5 illustrates the incorporation of the prompt by the two models. We denote the models with structured aspects as **BART-SA**, **GSum-SA**, respectively.

5.5 Experiments

5.5.1 Data Setup

A model would ideally take into account full texts from the set of reference articles to generate summaries. However, incorporating all information in the reference articles is challenging due to each article having a large number of words, not to mention the fact that survey articles tend to cite more articles than a regular research paper. In this context, previous work in scientific text generation adopted different parts of articles depending on tasks: Jha et al. (2015a) used introduction for survey generation, while Cachola et al. (2020) used abstract, introduction and conclusion for extreme summarization. Moreover, one could also select relevant parts of articles with respect to the target aspects. Despite all possibilities, we found that simply using abstracts is sufficient and concise, achieving the highest validation ROUGE scores in our preliminary experiments. Therefore, we extract abstracts from the reference articles to approximate the full article texts in this paper.

5.5.2 Evaluation

To evaluate the overall similarity of individual aspect-based summaries to gold-standard counterparts, we use the standard n-gram overlap based methods of **ROUGE**-1, 2, and L (Lin, 2004). However, per-summary evaluation do not fully reflect model quality in terms of responsiveness to the input aspect at article-level. Specifically, given multiple aspects on a single survey article, an ideal model is expected to generate an aspect-based summary that is the most similar to the gold standard summary for that aspect, making it more “responsive” to the input aspect. To quantify the responsiveness, we first obtain similarity scores[5] calculated between a

[5] ROUGE-2 is employed in this work.

particular aspect-based summary s_i and all the gold-standard summaries in the target survey article: sim(s_i, y_1), sim$(s_i, y_2), \ldots$ sim(s_i, y_k). These scores are then ranked in descending order. Using these ranks, we calculate and report Mean Reciprocal Rank (**MRR**; Voorhees, 2001)[6] by aggregating reciprocal ranks over all the summaries.

$$\frac{1}{N}\sum_{i}\frac{1}{\text{rank}\left(\text{sim}(s_i, y_i)\right)}, \tag{5.3}$$

where rank(sim(s_i, y_i)) returns the rank of the score among other scores calculated by comparing s_i to other gold-standard summaries. MRR is the highest when each generated summary is the most similar to the corresponding gold-standard summary. On the contrary, the score gets lower when a generated summary is more similar to gold-standard summaries for other aspects, which indicates poorer focus on the specified aspect. Thus, MRR considers the article-level summary quality with respect to how generated summaries correctly respond to the specified aspects.

5.5.3 Baselines

Previous approaches for summarization according to specified aspects include the use of aspect-augmented encoders (Fremann and Klementiev, 2019) or aspects as part of input prompts (Keskar et al., 2019). We compare summarization models from different categories.

Unsupervised A simple but effective heuristic method that exploits the lead bias is the **Lead** method, which simply regards the first K sentences of source texts as the summary. While the lead bias is most conspicuous in the news domain (Kryscinski et al., 2019), this method works reasonably well in previous works (Dou et al., 2021). To accommodate multi-document inputs, we take K sentences from *each* source article and concatenate them together to form a summary. Sentences that cause the summary to exceed the maximum summary length are discarded. Additionally, we also experiment with **TextRank** (Mihalcea and Tarau, 2004), a standard unsupervised extractive approach for summarization. Each sentence from multiple source articles is provided to the model, which gets ranked according to sentence importance scores calculated with a graph-based algorithm. An advantage of these methods is the ability to process long inputs without a computational bottleneck. On the other hand, both methods are unaware of the target aspects and generate only a single summary for each input.

[6] The value ranges $(0, 1]$, where 1 is the perfect situation.

Method	R-1	R-2	R-L	MRR
Oracle	39.85	8.94	36.24	0.625
Lead-5	30.50	5.26	28.59	0.212
TextRank	31.29	5.90	26.59	0.212
BertExt	32.26	6.02	30.14	0.258
GSum-SA	31.37	7.61	29.77	0.308
BART-SA	32.49	8.13	30.77	0.377

Table 5.2: Experimental results.

Supervised Unlike the unsupervised methods, state-of-the-art (extractive and abstractive) summarization models rely on encoding the source texts using pre-trained language models, which are less capable of handling long inputs. We mitigate this length bottleneck by heuristically compressing the input articles into shorter lengths. Specifically for each pair, articles are ranked according to TF-IDF between the target aspect and the articles, and top-10 of them are concatenated together.[7]

For comparison with the two aforementioned models (**BART-SA**, **GSum-SA**), we include **BertExt** (Liu and Lapata, 2019b), a supervised extractive summarization model. For the abstractive models, we noticed that decoding often suffers from endless generating citations or references (*e.g.* "[1], [2], . . ."). We introduce a post-processing step that truncates such citations and references made in the summaries.

5.5.4 Results

We show the summarization results in Table 5.2, as well as the extractive oracle. Having seen all the reference documents, light-weight methods such as **Lead** or **TextRank** performed competitively in terms of R-1, which reflects the low number of novel *unigram* (Figure 5.3) in the uncompressed inputs in the dataset. At the same time, higher order ROUGE scores and R-L are significantly worse as expected from the higher novel n-gram percentages for higher-orders of n.

Comparing the supervised models, **BertExt** achieved as good R-1 and R-L scores as abstractive models while suffering in R-2. Among abstractive models, we observed that **BART-SA** outperformed **GSum-SA** by a large margin, both in terms of ROUGE and MRR. We suspect that

[7] Each article text is equally truncated to leading sentences so that the concatenated text fits into the length budget, which is 1024 subword tokens.

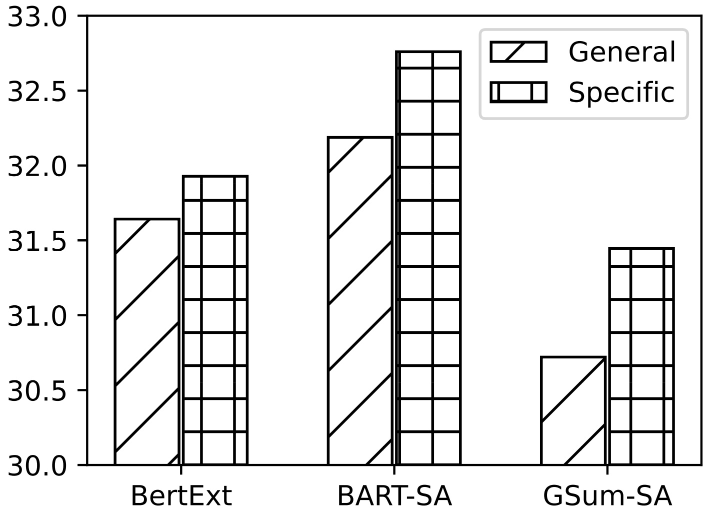


Figure 5.6: ROUGE-1 Comparison of supervised models according to Aspect Frequency. Specific aspect refers to aspects that appear uniquely in one survey article, while general aspects refer to aspects that appear across articles.

our task specifically benefits from aspect-source interaction much more than other guided summarization tasks tackled by Dou et al. (2021) with **GSum**, due to the necessity to semantically compare aspects and source texts for determining the right content to attend to and generate.

5.6 Discussion

We analyze the experimental results in depth below.

5.6.1 Relationship between Aspect Frequency and ROUGE

Aspects in the proposed dataset form a skewed frequency distribution, which means the models are exposed to them likewise. At the same time, frequent aspects tend to be general, while long-tail aspects tend to be more specific. To understand the influence of these aspect characteristics, we evaluate samples based on the two aspect groups defined earlier: *general* and *specific*. ROUGE-1 for the two groups by **BertExt**, **BART-SA**, and **GSum-SA** are plotted in Figure 5.6. We observe a global trend that all the abstractive models perform better with *specific* aspects, which may be counter-intuitive because of the rarity of these aspects. We suspect that this tendency is more due to the specificity of the aspect rather than the frequency; the more informative an aspect is the more accurate models can focus and summarize.

5.6.2 Human Evaluation

Automatic measures like ROUGE or MRR can only reveal surface-level qualities. To gain more insights on model differences, we conducted a human evaluation on three models (**BertExt**, **BART-SA**, **GSum-SA**) according to the four criteria below:

- **Coherence**: Does the text discuss the same content as before? (-1: Not coherent, 1: Coherent)
- **Fluency**: Is the text grammatically sound? (-1: Not fluent, 0: Somewhat fluent, 1: Fluent)
- **Aspect Relevance**: Is the text about the aspect? (-1: Not relevant, 0: Unsure, 1: Relevant)
- **Factuality**: Is the text factually correct? (-1: Not factual, 0: Unsure, 1: Factual)

Every criterion is assessed at sentence-level and then aggregated to represent at summary-level, which allows for more fine-grained analysis on the criteria. Normally, factuality of a summary is evaluated with respect to the source text, focusing on whether the content in the summary aligns with the source text (Kryscinski et al., 2020; Pagnoni et al., 2021). However, we found it difficult to ground each summary sentence to the source text for two reasons besides the difficulty in assessing factuality: (1) it is unreasonably costly to verify the factuality against the multi-document inputs consisting of large source texts, and (2) identifying the evidence from the source texts is challenging due to highly abstractive of survey texts. To tackle the annotation more reliably, we annotate the factuality of each sentence with respect to annotators' knowledge about the area. We note that this approach is only a step toward more accurate factuality annotation; it is still far from ideal in that the annotation quality highly depends on each annotator's knowledge-level.

Annotation data is selected by sampling 20 aspect-based summaries from NLP domain[8] instances in the test set. Each aspect-based summary is split into sentences, which results in 440.3 sentences on average across models.

We show the aggregated human annotation results in Table 5.3. All the models returned fluent sentences, achieving high fluency scores. With respect to coherence, **BertExt** was expectedly lower than the other two due to the lack of connections between sentences caused by the extractive approach. In terms of the aspect relevance, we found that **BART-SA** is significantly more responsive to the specified aspect than **GSum-SA**. Despite being the best, aspect relevance of **BART-SA** is still far from ideal and further model improvement on enforcing generation of relevant content is necessary.

[8] Before annotation, we ensure that automatic evaluation measures for these 20 samples do not deviate from the full evaluation results in Table 5.2.

Model	Coh.	Flu.	AspRel.	Fac.
BART-SA	0.908	0.927	0.596	0.422
GSum-SA	0.860	1.000	-0.050	0.300
BertExt	-0.589	0.888	-0.383	0.411

Table 5.3: Human annotation results.

Model	R-1	R-2	R-L	MRR
BART-SA	32.49	8.13	30.77	0.377
BART-A	31.92	7.70	30.27	0.357
BART	29.49	6.12	27.86	0.245
GSum-SA	31.37	7.61	29.77	0.308
GSum-A	31.13	7.32	29.51	0.292

Table 5.4: Ablation experiments on two abstractive models with structured aspects.

Factuality scores are low on all models, partially due to many sentences being unverifiable. For example, reported results, referencing the article sections, or discussions based on the results were annotated as "unsure". We observed that **GSum-SA** particularly had the tendency of generating summaries that have a similar style of writing to the source reference articles (which are not surveys).

5.6.3 Are structured aspects helpful?

To justify the utility of structured aspects, we compare the proposed models with single aspect variants, *i.e.* prompts only contain the target aspect. The evaluation results are shown in Table 5.4, where models with **-A** indicate the single aspect variants. We additionally report the results from **BART**, a baseline that does *not* take into account prompts. We observe that MRR, which is designed to capture the responsiveness of summaries to the specified aspects, captures differences between models more sensitively than ROUGE. Specifically, we observe a consistent trend that incorporating the structured aspects help the model generate an aspect-sensitive summary that is more disentangled from other summaries.

5.6.4 Qualitative Evaluation

We demonstrate the tendency and the performance of the models through metrics in Table 5.5, which is from a referring expression generation survey literature. Comparing the model outputs

BART-SA: of NLP research is based on the strong assumption that the objects in a scene can be identified through a set of commonly agreed properties. This is a strong assumption, since in reality objects may be perceived differently among people, due to a number of factors including vagueness, knowledge discrepancy, and limited perception capabilities. Taking these discrepancies into account, several researchers reinterpret concepts of algorithms generating referring expressions in view of uncertainties about the appearance of objects (Dale and Dale 1989; Linell 1998). Another important aspect of this approach is that it allows for the possibility of incremental changes in the set of objects [. . .]
GSum-SA: In this section, we present a number of techniques that have been used for the generation of referring expressions in the context of a visual context. These techniques can be broadly classified into two categories: (i) techniques that rely on a priori knowledge of the context, and (ii) those that do not. The first category of techniques is based on the assumption that the context is known in advance, that is, it is pre-defined in advance. The second category, on the other hand, relies on the fact that a context can be inferred from the context itself. [. . .]
BERTExt: It addresses the issue of combinatorial explosion inherent in the construction of relational context models by : (a) contextually defining the set of objects in the context that may function as a landmark, and (b) sequencing the order in which spatial relations are considered using a cognitively motivated hierarchy of relations, Algorithms for generating referring expressions typically assume that an object in a scenary can be identified through a set of commonly agreed properties. This is a strong assumption, since in reality properties of objects may be perceived differently among people, due to a number of factors including vagueness, knowledge discrepancies, and limited [. . .]
Gold: So far we have assumed that properties have a crisply defined meaning that is fixed, regardless of the context in which they are used. But many properties fail to fit this mold. Consider the properties young and old, for example. In Figure 1 , it is the leftmost male who looks the older of the two. But if we add an old-age pensioner to the scene then suddenly he is the most obvious target of expressions like "the older man" or "the old man." Whether a man counts as old or not, in other words, depends on what other people he is compared to: being old is a context-dependent property. [. . .]

Table 5.5: Generated examples from three supervised models on the aspect: **Context Dependency, Vagueness, and Gradability**, as well as the gold-standard summary.

with the gold-standard summary, we observe that all of them discusses the specified aspect to a different extent: **BART-SA** and **BertExt** are specific and **GSum-SA** states more regarding the paper organization. However, we see that no summaries are semantically identical to the gold-standard summary. On many occasions when the gold-standard summary is not short, we observed the tendency where the models generate summaries in their own expressions and style. Expectedly, sentences from **BertExt** are more disconnected than the other two due to its extractive nature.

5.7 Related Work

Automatic survey generation has previously been tackled at the scale of paragraphs (Huang, 2020; Jha et al., 2013, 2015a,b,c; Mohammad et al., 2009; Qazvinian et al., 2013; Xu et al., 2019). Considerations regarding content modeling for better coherence were studied using HMMs (Jha et al., 2015a) or topic models (He et al., 2016). Using key phrase detection, Yang et al. (2017) mined important aspects from citation sentences and employed integer linear programming to select relevant sentences as summaries.

Despite recent progress in neural abstractive summarization, there is little research on using abstractive models for survey generation and most work has been on simpler problem formulation: summarizing a single scientific article into an abstract. Since the introduction of large-scale scientific article summarization (Cohan et al., 2018), a number of summarization models improved the state-of-the-art performances (Lewis et al., 2020; Zhang et al., 2019b; Zhong et al., 2020). Going beyond summarizing articles with only the article content, the use of citations was considered in citation-based summarization (Abu-Jbara and Radev, 2011; Chandrasekaran et al., 2019; Cohan and Goharian, 2017; Parveen et al., 2016; Qazvinian and Radev, 2008; Yasunaga et al., 2019) Scientific documents allow for not only the aforementioned summarization tasks but other generation tasks. Cachola et al. (2020) leveraged author-written short summaries of articles as the target and proposed extreme summarization of scientific articles. Related work sections present similar information to survey articles, while focusing more on the content directly relevant to the target article (Hoang and Kan, 2010; Hu and Wan, 2014; Lu et al., 2020). Luu et al. (2021); Xing et al. (2020) studied generation of citation texts by incorporating document context and relationships between articles, respectively.

5.8 Implications and Future Work

In this work, we investigated structured aspect-based summarization, an aspect-based summarization task where the target aspects form a structure. Our data construction resulted in extracting only sequential aspect relationships in scientific documents, which do not accurately model the survey article texts. In particular, article data extraction must preserve hierarchical aspect structures. Given more complex structures such as trees or more generally graphs, more questions on how to efficiently incorporate them emerge. Methods that can leverage structure information, such as deciding the scope of context should be investigated.

Chapter 6

Learning to Generate from Fine-Grained Aspects

In this chapter, we study aspect-based NLG over fine-grained aspects regarding a topic, specifically focusing on the language modeling of Wikipedia texts conditioned on knowledge bases. Depending on different levels of specificity, different granularities of aspects can be considered in natural language. For example, restaurant owners who are interested in obtaining high ratings in review websites would primarily look for positive and negative aspects of different reviews. In the mean time, other owners who would want to further understand customers' reviews more specifically would look for the taste of one dish, presentation of another dish, and so on. Encyclopedic texts are in the latter category where highly specific texts are necessary; NLG systems should handle fine-grained aspects without missing or hallucinating the content. However, neural NLG models that generate the right content under the right aspects have been understudied. Therefore in this chapter, we propose Latent Relation Language Models (LRLMs), a variety of conditional language model capable of achieving the both control. In this chapter, we employ the local knowledge bases from Wikidata for the target entities as the conditional signal to language modeling on Wikipedia text about the entity, and relations or attributes outgoing from the entities are considered aspects (*e.g.* Barack Obama has an aspect `occupation`). Our model provides two predictors for generation: token-based and relation-based (aspect-based). By using both predictors over the texts and representing the probabilities, the model learns to switch predictors when the generation from aspects are appropriate or not. With this integration of aspect-based generation mechanism from knowledge bases, we not only outperform the previous results on the same task, but demonstrate less hallucination of aspect-related entities with our model.

The content in this chapter have been reported in the following work:

- Hayashi, Hiroaki, Zecong Hu, Chenyan Xiong, and Graham Neubig. Latent relation language models. In Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 05, pp. 7911-7918. 2020.

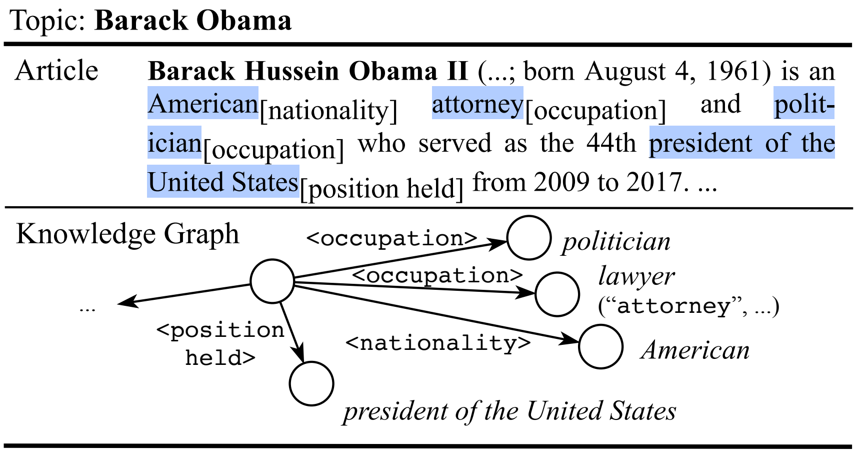
6.1 Overview

Language models (LMs) calculate the probability $P(X)$ of textual data X, and are a core model class of interest to NLP. LMs are used as testbeds for evaluation of generative models of text, and have applications such as rescoring of upstream language generation inputs (Sundermeyer et al., 2012), grammatical error correction (Felice et al., 2014), or pre-training of sentence representations (Peters et al., 2018). Neural networks are used to model this probability in state-of-the-art LMs (Bengio et al., 2003; Merity et al., 2017b; Mikolov et al., 2010).

Textual data X comprise a wide variety of words to be modeled, from closed-class function words, to common nouns or verbs, to named entities and numbers (Zipf, 1949). Notably, words on the rarer end of this spectrum are often more semantically or topically important, as evidenced by the success of heuristics such as TF-IDF (Salton and McGill, 1986), which up-weight words with low frequency. Previous work has noted that while neural LMs greatly outperform alternatives such as n-gram models on frequent words, they often under-perform on these rare words due to their limited parameter budget, which puts them at a disadvantage compared to non-parametric models like count-based n-grams (Neubig and Dyer, 2016).

Methods to mitigate this bottleneck have been proposed in the context of *conditional LMs*, which instead model the conditional probability $P(X \mid C)$, where C is some context given to the model. For instance, in sequence transduction tasks, there are mechanisms to copy from the source sequence (Gu et al., 2016) or use word or phrase dictionaries (Arthur et al., 2016) to improve modeling of low-frequency words. Perhaps more interesting from an LM perspective are methods conditioned on information from structured knowledge sources such as knowledge graphs (Ahn et al., 2016; Logan et al., 2019; Parvez et al., 2018), tables (Lebret et al., 2016), or grammars (Konstas and Lapata, 2013). These methods are analogous to human language production, where the underlying knowledge is converted into linguistic realizations.

In this chapter, we propose Latent Relation Language Models (LRLMs), a class of conditional LMs that take *relational* information between entities in a knowledge graph as context. Specifically, our model is able to generate either words from a fixed word vocabulary, or a span of words defined according to their relations with a topic entity of interest, as shown in Figure 6.1.



Topic: **Barack Obama**

Article

Barack Hussein Obama II (...; born August 4, 1961) is an American[nationality] attorney[occupation] and politician[occupation] who served as the 44th president of the United States[position held] from 2009 to 2017. ...

Figure 6.1: Overview of our task of language modeling conditioned on a knowledge graph. For a given topic, we want to learn a language model that leverages the knowledge graph through relations when modeling the text.

The choices of which method of generation to use is defined as a latent variable sequence Z. We use Latent Predictor Networks (LPNs; Ling et al. (2016)) to jointly learn $P(X, Z \mid C)$, thus tractably marginalizing over all the possible spans. Compared to other word-by-word generation methods that condition LMs on knowledge graphs (KGs; Ahn et al. (2016); Wang et al. (2018)), the span-based generation from the KGs alleviates problems of malformed or incomplete mentions. Moreover, the posterior probabilities of Z can be considered as entity links, which are of interest in their own right in the information extraction field (Ceccarelli et al., 2013; Ganea and Hofmann, 2017).

We apply the model on articles from Wikipedia (X), with the help of relational information (C) such as Wikidata (Vrandečić and Krötzsch, 2014) or Freebase (Bollacker et al., 2008) regarding each article topic. Empirical results on open vocabulary language modeling show that the proposed model outperforms previous approaches on the same task, demonstrating that LRLMs provide an effective way to condition on this context. We also demonstrate the merit of explicitly modeling latent relations by examining the posterior probabilities over the chosen relations Z, which are in concert with human intuitions about how relations are being expressed in the text.

6.2 Related Work

A variety of entity-aware LMs exist, conditioning on information sources such as coreference annotations (Ji et al., 2017), entity annotations (Logan et al., 2019), or keywords (Kiddon et al., 2016; Parvez et al., 2018). Among them, NKLM (Ahn et al., 2016) uses relational information and is the most relevant. Our proposed LRLM formulation is more successful at lowering perplexity and allows calculating posterior probabilities of relations.

Incorporating KGs for natural language generation (NLG) has a long history (Chen and Mooney, 2008; Goldberg et al., 1994; Reiter et al., 2005). With the recent advancement of neural sequence modeling, prevalent approaches for language generation from KGs employ sequence-to-sequence models with special attention mechanisms tailored for input structures such as graphs (Wang et al., 2018) or tables (Liu et al., 2018b). Unlike our focus, however, this class of research focuses on learning discriminative models that do not explicitly generate the referent entity as latent variables, like we do in Section 6.

While not directly related to our core task, there have been a number of other methods for incorporating latent variables into NLG problems. Latent structure has included predicting latent sequences of topics (Wiseman et al., 2018), chunking of word sequences into n-grams (Buckman and Neubig, 2018), deciding between input sources (Gu et al., 2016), or generating compressed summary tokens (Miao and Blunsom, 2016). Our model borrows its underlying structure from Ling et al. (2016), who focused on an entirely different task of source code generation. We use a similar method for selecting latent sources for Wikipedia article language modeling with a repository of KG triples.

6.3 Language Modeling Conditioned on Structured Knowledge

In this section, we define the task of open-vocabulary language modeling conditioned on structured data.

6.3.1 Task Definition

Knowledge graphs (KGs) can be represented as a directed labeled graph $G = (V, E)$ consisting of a set of nodes $V = \{v_1, \ldots, v_{|V|}\}$ and a set of relation edges $E = \{e_i{:}\, \langle s_i, \omega_i, o_i \rangle \mid s_i, o_i \in V,\ \omega_i \in R\}$.

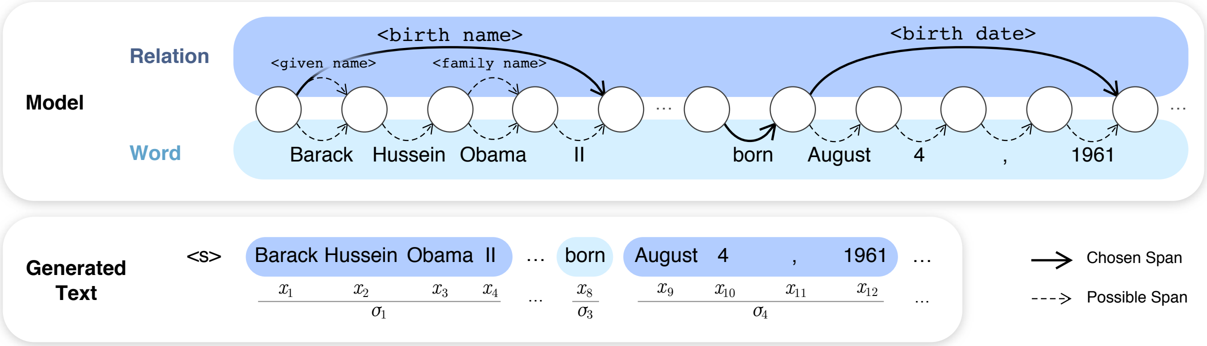


Figure 6.2: While generating, our model switches between the two *sources*: “Relation” and “Word”. Circles represent hidden states up to each token, and edges represent possible span matches. Here we show one valid derivation with solid lines, and other options as dashed lines. We also show an “annotation” of the generated tokens by the spans and sources we choose.

Relation e_i contains s_i, ω_i, and o_i as the subject, relation type, and object. R is the set of all relation types. Each node $v_i \in V$ represents either an entity or an attribute[1], and is associated with a set of surface forms (also called aliases) $\mathcal{A}(v_i) = \{a_{i,1}, \ldots, a_{i,|\mathcal{A}(v_i)|}\}$ that can be used to refer to v_i. For instance in Figure 6.1, the subject “*Barack Obama*” is connected to both “*politician*” and “*lawyer*” with the relation <occupation>, and the object entity “*politician*” has “political figure” and “polit.” as additional aliases. Notably surface forms of many objects in the KG can be multiple words, and thus it is necessary to have machinery to deal with this fact.

Given this KG, we further define a topic entity s about which we would like to generate a piece of text. Our conditional language modeling problem is then defined as the problem of modeling the conditional probability of text X: $P(X \mid G, s)$. In particular, we consider a subgraph $G' = (V', E')$ of the original KG G by extracting nodes and edges directly related to the topic entity s:

$$\begin{aligned} V' &: \{s\} \cup \{o_i \mid \langle s, *, o_i \rangle \in E\}, \\ E' &: \{e_i{:}\langle s, \omega_i, o_i \rangle \mid \langle s, \omega_i, o_i \rangle \in E \wedge o_i \in V'\}. \end{aligned}$$

We consider an *open-vocabulary* setting where all word types within X are incorporated. Perplexity under this setting provides a more realistic measure than under closed-vocabulary setting by taking into account words that rarely or never appear in the training set, which, as previously noted, are particularly important for conveying the main content of the text.

[1] A value specified with a relation from an entity (e.g., dates).

6.3.2 Why Condition on Knowledge Graphs?

KGs provide two important benefits for neural LMs. First, the high coverage of rarer words due to entities being often infrequent addresses lack of textual supervision for predicting these words. More importantly, KGs have the potential to help LMs generate *factually consistent* text by providing consistent associations between entities. Normal LMs would have to rely on supervision purely from textual data, which may not provide a learning signal strong enough to accurately generate these facts. For instance, results from Radford et al. (2019) show that even with a very large model trained on massive amounts of data, samples can be factually incorrect, although being fluent and coherent.

6.4 Latent Relation Language Models

In this section, we describe our proposed framework of Latent Relation Language Models (LRLMs).

6.4.1 Definition

Knowledge from the KG subgraph G' can be incorporated into generation by copying aliases from related entities into the generated text. For instance in Figure 6.2, to generate Obama's birth date, the model can of course pick words from its vocabulary. But it is more straightforward to copy from the `<birth date>` relation of the topic entity "*Barack Obama*", which gives the correct birth date.

However, it is insufficient to model probabilities for such choices conditioning only on G' and s, because it is unknown to us which text spans are matched to which relations. Naïve solutions like simple text matching algorithms would yield many false positives. For example, "*New York City*" has an alias "`New York`", which matches "*New York*" (state) and parts of "*New York City Council*".

To circumvent this lack of relation annotation, we treat relations corresponding to such text spans as latent variables. Formally, let $X = \{x_i\}_{i=1}^{N}$ be the sequence of N tokens, and $Z = \{(\sigma_t, \pi_t, \rho_t)\}_{t=1}^{T}$ a sequence of latent variable triplets describing text span matches:

- The *span* variable $\sigma_t := (\ell_t, r_t)$ specifies a token subsequence $x_{\sigma_t} = \{x_i\}_{i=\ell_t}^{r_t}$.
- The *source* variable $\pi_t \in \{\textsc{rel}, \textsc{word}\}$ denotes the generation source of the span x_{σ_t}.
- The *relation* variable $\rho_t := (e_t, a_t)$ describes the matching relation and surface form of the span x_{σ_t}, and is only used when $\pi_t = \textsc{rel}$.

Algorithm 1 Generative Process of LRLM

Input previous span $\sigma_{t-1} = (\ell_{t-1}, r_{t-1})$, previously generated tokens $x_{<r_{t-1}}$.
Output span $\sigma_t = (\ell_t, r_t)$, source π_t, relation $\rho_t = (e_t, a_t)$, and token subsequence x_{σ_t}.

```
$\ell_t \leftarrow r_{t-1} + 1$                                        ▷ Update the beginning of span. :1
$\widehat{\pi}_t \sim P(\pi_t \mid x_{<\ell_t})$                        ▷ Choose whether to generate a word or relation. :2
if $\widehat{\pi}_t = \textsc{word}$ then                              ▷ Generating a word. :3
    $P(\sigma_t, x_{\sigma_t}, \rho_t \mid \pi_t = \textsc{word}, x_{<\ell_t}) \coloneqq P(x_{\ell_t} \mid x_{<\ell_t})$    ▷ Simplify the probability. :4
    $\widehat{x}_{\ell_t} \sim P(x_{\ell_t} \mid x_{<\ell_t})$          ▷ Choose a word from model vocabulary. :5
    if $\widehat{x}_{\ell_t} =$ <UNK> then
        $\widehat{x}_{\ell_t} \sim P(c_1 \ldots c_{|c|}; \theta_{\text{char}})$    ▷ Generate a word using a character model. :7
    else if $\widehat{x}_{\ell_t} =$ <EOS> then
        End generation.
else if $\widehat{\pi}_t = \textsc{rel}$ then                          ▷ Generating a relation. :10
    $P(\sigma_t, x_{\sigma_t}, \rho_t \mid \pi_t = \textsc{rel}, x_{<\ell_t}) \coloneqq P(e_t \mid x_{<\ell_t}) P(a_t \mid e_t, x_{<\ell_t})$    ▷ Factor the probability. :11
    $\widehat{e}_t \sim P(e_t \mid x_{<\ell_t})$                        ▷ Choose a relation. :12
    $\widehat{a}_t \sim P(a_t \mid \widehat{e}_t, x_{<\ell_t})$         ▷ Choose a surface form from the selected relation. :13
    $\widehat{x}_{\sigma_t} \leftarrow \widehat{a}_t$                   ▷ Generate a phrase. :14
```

For Z to be a valid sequence of latent variables, the following conditions must be satisfied:

- Span variables $\{\sigma_t\}_{t=1}^{T}$ form a *segmentation* of X, *i.e.*, $\ell_t = r_{t-1} + 1$ for $t = 2, \ldots, T$. This also implies $T \leq N$.
- If $\pi_t = \textsc{word}$, then $\ell_t = r_t$.
- If $\pi_t = \textsc{rel}$, then $\rho_t = (e_t, a_t)$ where $e_t = \langle s, \omega_t, o_t \rangle$ should satisfy $e_t \in E'$, $a_t \in \mathcal{A}(o_t)$, and $x_{\sigma_t} = a_t$, *i.e.*, ρ_t must correspond to a valid surface form of an object that is related to the topic entity s and matches the text span.

Let $\mathcal{Z}$ be the set of all valid latent variable sequences. We can now model the probability by marginalizing over $\mathcal{Z}$:

$$P(X \mid G', s) = \sum_{Z \in \mathcal{Z}} P(X, Z \mid G', s). \tag{6.1}$$

For sake of brevity, unless noted otherwise, we drop G' and s from the conditions in the following sections.

6.4.2 Training

Given the latent variable sequence Z, we follow Ling et al. (2016) in factoring the joint probability:

$$\begin{aligned} P(X, Z) &= \prod_{t=1}^{T} P(\sigma_t, \pi_t, \rho_t, x_{\sigma_t} \,|\, x_{<\ell_t}) \\ &= \prod_{t=1}^{T} P(\pi_t \,|\, x_{<\ell_t}) P(\sigma_t, x_{\sigma_t}, \rho_t \,|\, \pi_t, x_{<\ell_t}), \end{aligned}$$

here $x_{<i}$ is the sequence of first $i-1$ tokens in X. Figure 6.2 shows an example of generation according to this factorization, and Algorithm 1 precisely defines the process of generating at time step t.

We marginalize over $\mathcal{Z}$ according to Eq 6.1 and optimize for the marginal likelihood. Since the probability at time step t is independent of previous latent variables, the marginalization is tractable using the forward-backward algorithm (Baum et al., 1970). The forward probability α_i is defined as the marginal probability of the sequence up to the i-th token (specifically, $\alpha_0 = 1$), computed as follows:

$$\alpha_i = \sum_{(\sigma:(\ell,r),\pi,\rho)\in\tau_i} \alpha_{\ell-1} P(\sigma, \pi, \rho, x_\sigma \,|\, x_{<\ell}),$$

where τ_i is defined as the set of valid latent variable tuples $(\sigma : (\ell, r), \pi, \rho)$ such that $r = i$, *i.e.*, all valid spans ending at the i-th token. The marginal probability we optimize for is then α_N. The backward probability β_i which is required for gradient computation can be similarly calculated.

6.4.3 Parameterization

We use neural networks to parameterize all probability distributions mentioned above. Decisions for time step t are based on a D-dimensional hidden state $\mathbf{h}_{\ell_t}$. This hidden state can be generated by any neural sequence model, and we experiment with multiple models to demonstrate the generality of our approach.

Source Selection

Source selection is done using a simple linear model followed by a softmax function applied to the latest word-level hidden state $\mathbf{h}_{\ell_t}$:

$$P(\pi_t \,|\, x_{<\ell_t}) = \mathrm{softmax}(\mathbf{W}_\pi \mathbf{h}_{\ell_t} + \mathbf{b}_\pi),$$

where $\mathbf{W}_\pi \in \mathbb{R}^{2\times D}$, $\mathbf{b}_\pi \in \mathbb{R}^2$ are trainable parameters.

Word Generation

Like conventional word-level neural language models, we have the option to generate the next token from a fixed vocabulary. This option is used to generate any word that isn't part of an object entity participating in a relation. The probability is:

$$P(x_{\ell_t} \,|\, x_{<\ell_t}) = \mathrm{softmax}(\mathrm{Linear}_w(\mathbf{h}_{\ell_t})),$$

where $\mathrm{Linear}(\mathbf{h})$ is a linear transform with a bottleneck of dimension K into a vector over vocabulary size L:

$$\mathrm{Linear}(\mathbf{h}) = \mathbf{W}_1(\mathbf{W}_2\mathbf{h} + \mathbf{b}_2) + \mathbf{b}_1,$$

where $\mathbf{W}_1 \in \mathbb{R}^{L\times K}$, $\mathbf{b}_1 \in \mathbb{R}^L$, $\mathbf{W}_2 \in \mathbb{R}^{K\times D}$, $\mathbf{b}_2 \in \mathbb{R}^D$ are trainable parameters. Empirically we found this low-rank version to outperform a full linear transform.

Unknown Word Generation

Since our task is language modeling under an open-vocabulary setting, we must be able to generate words even if they are out of vocabulary. Following Luong and Manning (2016), we do so by having a character-level LM "spell-out" any unknown words. If the unknown word is $x = c_1 \ldots c_{|c|}$ with $|c|$ characters:

$$P(x \,|\, x_{<\ell_t}) = P(\text{<UNK>} \,|\, x_{<\ell_t}) P(c_1 \ldots c_{|c|}; \theta_{\text{char}}),$$

where θ_{char} are the parameters of the character LM. We pre-train this model on the set of all unique words in the training set and fix its parameters while training LRLM.

Dataset	Doc	Vocab	Rel/Ent	Tok/Doc	Ment/Doc
WikiFacts	7856	40.0k	82.71	157.25	16.04
WikiText-S	27685	71.1k	11.38	295.75	11.20
WikiText-F	27685	264k	11.38	3559.91	73.01

Table 6.1: Training set statistics: number of training documents, vocabulary size, relations per head entity, tokens per document, and entity mentions per document.

Relation Generation

The goal of relation generation is to find the most suitable span that can be copied into the text. As Line 11 of Algorithm 1 depicts, this is factorized into two steps: relation selection and surface form selection.

- **Relation selection.** We utilize pre-trained KG embeddings from OpenKE (Han et al., 2018) for entities and relation types. For a relation $e_i : \langle s, \omega_i, o_i \rangle$, we concatenate KG embeddings for ω_i and o_i to obtain the *relation embedding* $\mathbf{e}_i$.[2] We then compute the probability of selecting each relation as:

$$P(e_i \,|\, x_{<\ell_t}) = \mathrm{softmax}(\mathbf{e}_i^\top \mathrm{Linear}_o(\mathbf{h}_{\ell_t})).$$

- **Surface form selection.** We featurize surface forms via fastText embeddings (Bojanowski et al., 2017) pre-trained on the training corpus, and calculate probability of surface form a_k as:

$$P(a_k \,|\, e_i, x_{<\ell_t}) = \mathrm{softmax}(\mathbf{f}_{a_k}^\top (\mathbf{W}_a \mathbf{h}_{\ell_t} + \mathbf{b}_a)),$$

where $\mathbf{f}_{a_k}$ is the fastText embedding for a_k and $\mathbf{W}_a$, $\mathbf{b}_a$ are trainable parameters.

6.5 Datasets

We use two datasets with different characteristics for experiments; statistics are shown in Table 6.1.

[2] We train embeddings for each relation type not covered by pre-trained embeddings, and an UNK embedding for attributes and entities not covered by pre-trained embeddings.

Base model	Dataset	Dev				Test			
		Vanilla LM	Alias LM	NKLM	LRLM	Vanilla LM	Alias LM	NKLM	LRLM
LSTM	WikiFacts	231.03	213.34	96.77	**93.55**	225.40	207.57	93.18	**88.37***
	WikiText-S	68.37	70.07	46.16	**45.84**	86.12	87.75	55.98	**55.38**
	WikiText-F	45.13	46.18	44.46	**42.18***	49.47	50.88	48.54	**45.70***
Transformer-XL	WikiFacts	172.27	158.54	99.46	**84.76****	167.91	154.27	94.36	**79.35****
	WikiText-S	42.63	39.65	43.05	**37.75****	52.96	50.60	52.51	**44.98****
	WikiText-F	30.14	31.20	32.19	**29.56****	33.01	34.37	35.27	**32.20****

Table 6.2: Perplexity values of different models on open vocabulary language modeling, lower is better. Best results are in bold. Asterisk symbols represent statistical significance according to Wilcoxon signed-rank test (Dror et al., 2018) against the best baseline model, with $p < 0.05$ (*) and $p < 0.01$ (**), respectively.

6.5.1 WikiFacts

WikiFacts (Ahn et al., 2016) is a collection of Wikipedia articles restricted to `/film/actor` domain entities in Freebase (Bollacker et al., 2008).[3] Each example consists of the *first section* of the original article. Since official splits for evaluation are not provided, we follow previous work and performed a random split of 80/10/10%.

This dataset assumes a single alias for each entity (*i.e.*, $\forall o \in V'; |\mathcal{A}(o)| = 1$). Hence, the surface form selection module acts as oracle, where it always assigns a probability of 1 to the correct surface form.

6.5.2 WikiText

While WikiFacts has been used in previous work on LMs using structured data (Ahn et al., 2016), the domain is limited. To investigate the capability of knowledge-infused LMs in an open-domain setting with a wide variety of relations, we build a large-scale open-domain dataset from the existing WikiText-103 dataset (Merity et al., 2017b) by associating articles with entities in Wikidata (Vrandečić and Krötzsch, 2014). We employ the same data splits from the original dataset. Bridging KGs and the articles from WikiText-103 involves two steps (more details in Appendix A).

- **Constructing subgraphs for articles.** As discussed in Section 2, we take the original KG and extract a relevant subgraph G' for each article. While there are many options on how to extract this subgraph, we choose the subgraph G' consisting of *direct neighbors* of

[3] The original WikiFacts also includes topic entities from other articles linked to the page to be generated. However, these (gold) entities are inaccessible when actually attempting to generate new articles. We experiment without them, but also report results with them in Appendix C.

the topic entity for each article. This forms a star-shaped subgraph, with the topic entity as the central node, connected by the related entities and attributes. We found on average 11.38 neighbors and 3.1 surface forms for each neighbor.

- **Linking mentions with the KG.** For each object entity in G', we search for occurrences of all surface forms in the article while allowing token overlaps among them. Note that, similarly to distant supervision for relation extraction (Mintz et al., 2009), this string-matching process can produce false positive mentions. We rely on our model's ability to handle such noisy mentions by learning to assign high probabilities only on the correct mentions.

We name the dataset obtained through this process as WikiText-F (Full). We also create WikiText-S (Short) by only using the first sections of WikiText-F documents.

6.6 Experimental Settings

In this section, we explain the evaluation metric, configurations, and baseline models compared against LRLM.

6.6.1 Evaluation Measure

We report token-level perplexity under the *open-vocabulary* setting. We use pre-trained character-level LMs from Section 3 for each dataset to discount the probability of out-of-vocabulary words based on its spelling.[4] This is done for all tested models, both proposed and baselines.

6.6.2 Model Configuration

For WikiFacts, we use a fixed word vocabulary size of 40,000 following Ahn et al. (2016). For WikiText-derived datasets, we include all words with frequencies no less than 3 in our dataset following Merity et al. (2017b). We use adaptive embeddings (Baevski and Auli, 2019) and adaptive softmax (Grave et al., 2017) to handle large vocabulary.

To calculate the hidden state $\mathbf{h}_{x_{<i}}$, we test two varieties of neural sequence models: standard LSTMs (Hochreiter and Schmidhuber, 1997), and the state-of-the-art Transformer-XL (Dai

[4] This contrasts to UPP (Ueberla, 1994), which adjusts likelihood of OOV words based on a uniform probability equivalent to the size of the vocabulary, which does not actually measure the ability to generate words outside of training data. Results using closed vocabulary setting or UPP can be found in Appendix C and E, respectively.

et al., 2019). We implement all models in PyTorch (Paszke et al., 2017). Training details and hyperparameters are summarized in Appendix B.

6.6.3 Baselines

We compare LRLM against three baselines that utilizes information from KGs to various degrees.

Vanilla language model (Vanilla LM)

This is a standard language model baseline that does not condition on KGs, such as LSTM (Merity et al., 2017a) or Transformer-XL (Dai et al., 2019).

Alias-prepended language model (Alias LM)

The same model as above, but prepending to the text the concatenated aliases of all entities in G' which appear in the article.[5] This gives a simple baseline LM conditioned on the KG.

Neural Knowledge Language Model (NKLM)

Similarly to LRLM, the Neural Knowledge Language Model (NKLM; Ahn et al. (2016)) also has the ability to copy from a given set of KG triples, but differs from LRLM in several ways:

1. LRLM allows generation of multi-word entities at once, while NKLM predicts one word at a time and the model needs to repeatedly predict the right relation until copying of an object is done.
2. LRLM marginalizes over all derivations of a sequence, which allows processing of overlapped tokens among spans, while NKLM makes all decisions in a hard fashion and cannot handle such overlapped tokens.[6]

The original NKLM does not differentiate between surface forms, so we incorporate the same surface form selection module as LRLM for fair comparison.

[5] This simulates the table-to-text generation setting, where all input entities should be mentioned in the generated sentence.

[6] Due to this limitation, we perform additional data preprocessing on WikiText for NKLM, detailed in Appendix D.

Warm-up: <s> = Sonic the Hedgehog (1991 video game) =	
Vanilla LM:	Sonic the Hedgehog is an action action - adventure video game published by Sonic of programmers for the 1999 Nintendo GameCube 's SNES video game Sonic the Hedgehog 2 . It was released for the Nintendo DS on September 16 , 1994 in North America and Europe in the latter part of the original Halo 2 . It was played in a post - apocalyptic fantasy fantasy universe , by Nintendo Computer Entertainment on March 6 , 1999 under the ...
Alias LM:	is the twelfth video game developed and published by EA Sports . It is the sequel to the 1992 Sonic - 6 gameSonic the Hedgehog for the Nintendo Genesis , created by Sonic system creator Pinball Ka . Doctor the Hedgehog has since gone on a hiatus in choosing an estimated global community when it quickly becomes a launch member . The game puts Princess unlock from a mansion , which once everything is devastated by a bro and I ...
NKLM:	Sonic the Hedgehog$_{\text{[TITLE]}}$ Sonic the Hedgehog$_{\text{[TITLE]}}$ Sonic the Hedgehog$_{\text{[TITLE]}}$ (Jordon VS . ストリートファイター , Picardi Sutorīto Faitā Jobs Sutorīto Faitā) is A$_{\text{[CERO rating]}}$ platform video game$_{\text{[instance of]}}$ developed by Sonic Team$_{\text{[developer]}}$ and published by Sega$_{\text{[publisher]}}$. In 2008 , the game$_{\text{[instance of]}}$ was released for Sega$_{\text{[platform]}}$ 3$_{\text{[platform]}}$ and 1320 platforms in March 2009 , replacing the original game$_{\text{[instance of]}}$ Sony$_{\text{[platform]}}$ Drive$_{\text{[platform]}}$ counterpart Sonic the Hedgehog$_{\text{[characters]}}$ for the Android$_{\text{[platform]}}$ GameCube$_{\text{[platform]}}$. It was re - released on March 12 , 2010 , in ...
LRLM:	Sonic the Hedgehog$_{\text{[TITLE]}}$ (also known as Sonic the Hedgehog 3 and Sonic$_{\text{[series]}}$ the Hedgehog 2) is a 1986 role - playing video game developed by Sonic Team$_{\text{[developer]}}$ and published by Sony Computer Entertainment (SEGA$_{\text{[publisher]}}$) for the PlayStation 3$_{\text{[platform]}}$ (Xbox 360$_{\text{[platform]}}$) . It was developed and published by Sega$_{\text{[publisher]}}$ in 1997 for the Wii , and was ported as a third installment in the Sonic the Hedgehog$_{\text{[series]}}$ series and released in Japan in 1996 . On the ...

$s =$ **Sonic the Hedgehog (1991 video game)**

$(\omega, o) = \{$
(`<TITLE>`, *Sonic the Hedgehog (1991 video game)*),
(`<instance of>`, *video game*),
(`<CERO rating>`, *A*),
(`<developer>`, *Sonic Team*),
(`<publisher>`, *Sega*),
(`<platform>`, *Sega Mega Drive*),
(`<platform>`, *Wii*),
(`<platform>`, *Nintendo GameCube*),
(`<platform>`, *Xbox 360*),
(`<platform>`, *Playstation 3*),
(`<platform>`, *Android*),
(`<characters>`, *Sonic the Hedgehog*),
(`<series>`,
Sonic the Hedgehog (video game series)),
...
$\}$

Figure 6.3: Samples from the models for the topic entity "*Sonic the Hedgehog (1991 video game)*" with the corresponding subgraph on the right. Square brackets denote the relation type of copied objects. Highlighted spans in light green are full mentions, and those in dark red are partial mentions. Underlined tokens are unknown words sampled from the character model.

6.7 Results and Analysis

In this section, we summarize the main results and perform analyses of the learned model.

6.7.1 Main Results

Perplexities over the datasets are shown in Table 6.2. We observe that for both sequence models, LRLM outperforms the baselines on all datasets and improvements are *more* significant on the stronger sequence model. Particularly on the two WikiText-derived datasets, our model outperformed the simpler Vanilla LM and Alias LM baselines, while NKLM had difficulty utilizing the KGs and in some cases results in worse perplexities than these baselines. Alias LM under-performed Vanilla LM in some cases, demonstrating that this simpler and more indirect method of conditioning on the linearized KG is not sufficient to achieve stable improvements.

6.7.2 Generated Samples

To illustrate behaviors of the learned models, we take the models using Transformer-XL trained on WikiText-S, draw 10 samples while conditioning on G' and $s =$ "*Sonic the Hedgehog*", and show the sample with lowest perplexity in Figure 6.3. We highlight tokens generated by the

	Partial	Full	Valid	Invalid
NKLM	16.9	7.81	6.37	1.44
LRLM	–	6.32	5.63	0.69
Gold	–	9.00	9.00	0.00

Table 6.3: Average number of partially generated, fully generated, and valid and invalid full mentions over 100 samples from the development set or gold human-generated article.

relation predictor and use different colors to represent full and partial mentions. A *full mention* is an identical copy of an entity surface form, while a *partial mention* is an incomplete subphrase of an entity surface form. A perfect model should not generate partial mentions as it leads to possibly corrupted phrases, and should generate the same set of full mentions as the gold article.

Although NKLM generates more mentions, it suffers from generating partial mentions because it 1) is unaware of the length of surface forms, and 2) requires making copy decisions as many times as the surface form lengths. As shown in Figure 6.3, we often observe NKLM repeating the same entity, or switching entities halfway through (*e.g.*, "*Sega 3*"). In contrast, LRLM, by design, only generates full mentions.

We quantitatively show this in Table 6.3 by counting the average number of partial and full mentions in samples. We took 10 samples from 10 random topic entities in the development set, and manually annotated "valid" full mentions, which we deemed as semantically correct based on the sentential context. NKLM generates more invalid mentions than LRLM, most of which are false positives and repetitions of the same mention. LRLM has almost no repetitions, but sometimes incorrectly predicts the "theme" of the topic entity, *e.g.*, generating an article about a TV episode for a topic entity of a song.

6.7.3 Posterior Probability of Spans

One of the advantages of our model is its capability to calculate the posterior probability of a span being generated as a relation in existing text. We calculate the joint probability of a span ($\sigma = (\ell, r)$) and the surrounding text[7] by marginalizing over the latent variable Z for both sides

[7] We consider the text segment in the batch where the span appears as the surrounding text.

Title: Sorry (Madonna Song)		
... song by American singer Madonna from her tenth ...		
Relations:	<performer>	**0.9697**
	<lyrics by>	0.0289
	word	0.0014
... written and produced by Madonna and Stuart Price , ...		
Relations:	<performer>	0.1545
	<lyrics by>	**0.7693**
	word	0.0762
... continuation from the " Hung Up " music video		
Relations:	<follows>	**1.0000**
	word	0.0000
... . However , in the United States , the song did ...		
Relations:	<origin>	0.0000
	word → <origin>	0.0003
	word	**0.9997**

Table 6.4: Posterior probability of spans (underlined) in contexts. word represents word-based generation. The second relation in the last example means generation of "*the*" using word, followed by relation-based generation of "*United States*" using the <origin> relation.

of context, and normalize over all possible spans:

$$P(X,Z) = \alpha_{\ell-1} \cdot P(Z \,|\, x_{<\ell}) \cdot \beta_{r+1},$$
$$P(Z \,|\, X) = P(X,Z) \,/\, \sum_{Z \in \mathcal{Z}} P(X,Z),$$

where α_i and β_i are the forward and backward probabilities computed following Section 3. Table 6.4 shows spans with posterior probabilities of various relation types from an article about "*Sorry (Madonna song)*". The model demonstrates the ability to relate the entity "*Madonna*" to the topic with appropriate relation types based on context. We also observe that the model tends to generate multi-word spans through relations rather than word-by-word from vocabulary. However, our model often favors word-based generation for common phrases even if related entities exist.

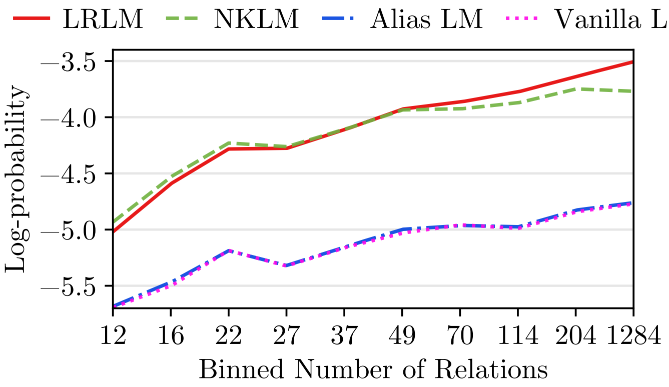


Figure 6.4: Word-average log-probabilities on development set of WikiFacts grouped by the average number of relations. LRLM shows a larger gain over the baselines as the number of relations increases.

6.7.4 Effect of Subgraph Size

Finally, we measure the performance of models with respect to the richness of resources available for conditioning. We group WikiFacts articles into 10 bins by the number of relations available, and plot binned word-average log-probabilities in Figure 6.4. While all models have slightly higher log-probabilities as the number of relations increase, LRLM achieves the largest gain.

Chapter 7

Conclusion

7.1 Summary of Contributions

In this thesis, we explored aspect-based natural language generation (NLG) from various angles.

With a lack of the definition of aspect, we first defined it based on the usage in previous studies in Chapter 2, that is, *aspect* is a semantic property of an object. In other words, aspects in NLG tasks specify the content of the text given the same underlying object such as a topic or an entity.

Having defined aspects, we first focused on aspect-based summarization, a specific instance of aspect-based NLG, and tackled the issue of lack of diversity in datasets. To address the issue that most natural datasets in aspect-based summarization are in customer reviews domain, we turned to Wikipedia and formulated the summarization task of web references into salient sections as aspect-based summarization task. We leveraged the domain diversity in Wikipedia and constructed WikiAsp dataset consisting of 20 diverse domains, each of which contains 10 salient aspects. Through the experiments and analyses, we found that domains in WikiAsp have unique characteristics that add more domain-specific complexities to the summarization task, such as chronological consistency and changes of perspective (*i.e.* first-person to third-person).

Next, we investigated the problem of transferring aspect-based summarization models to new domains in a zero-shot manner. A successful model performing reasonably well on new domains is crucial to practical use cases of aspect-based summarization models, because obtaining supervised aspect-based summarization data is costly. Unlike previous studies that regarded datasets as domains, we leveraged the aforementioned WikiAsp that includes a wide variety of domains. Moreover, WikiAsp allows for transfer experiments with respect to aspects, which allows for multi-level analysis of model transfer. We devise a two-stage classification-summarization model that (1) classifies source sentences with a NLI-based aspect classifier and (2) summarizes by taking into account domain and aspect information. Experiments showed that prompting is most effective at incorporating target domain and aspect information when transferring the summarization models. Despite this, there remained a large performance gap between supervised models and the best zero-shot transfer model. We performed error analyses and discussed challenges that needed to be addressed to improve the performance.

Third, we examined the incorporation of structures in aspects, making the outputs structure-aware, multi-aspect summaries. As multiple users likely have different interests, it is important for a model to take into account the relationships of aspects rather than treating them independently when generating multiple aspect-based summaries. To investigate this task, we regarded scientific survey articles as structurally organized multi-aspect summaries of reference articles, and construct a dataset. On this dataset, we developed a structured aspect model and show that modeling structured aspects help the summary quality of for the target aspects.

Finally, we shift our focus to more fine-grained aspect-based generation. Specifically, we considered relations outgoing from entities as the aspect and the entities as the underlying common object. Different from previous parts, models have to select fine-grained aspects at the right time in order to avoid inconsistent sentences. We proposed a model that is capable of learning to switch between generating using aspects or tokens, thereby improving the perplexity for language modeling tasks. The proposed latent relation language models can then generate text using either fine-grained aspects or tokens, depending on the likelihood of the outcome. The model can not only generate according on the specified fine-grained aspects but also serve as an aspect extractor given a sentence by modeling all possible generation steps.

7.2 Future Directions

This thesis investigated neural aspect-based generation from various angles including data, aspect structures, and aspect granularity. Through the thesis, however, we observe key challenges that are still yet to be addressed, which we list below.

Handling Complex Structural Aspects Throughout this thesis, problem settings involved *multiple* aspects for a single object, including topic, entity, and so forth. In Chapter 5, we focused on leveraging the relationships between these aspects to improve generation, specifically taking the sequential relationship in a survey article. Realistically, however, aspects form more complex relationships, such as a tree or a more general graph. An example would be tree-form aspects that specify not only the content but dependence through the structure, such as scoping the pronoun resolution. To achieve the above, one must find a dataset where such structural aspects are relatively cheap to obtain. As a next step, improved PDF parsing on scientific articles would help obtain and reconstruct more complex section structures, which we can leverage as structural aspects.

Enforcing the Aspect Controllability No matter how rich structures aspects have, NLG models have to incorporate them effectively to generate text accordingly. For example, superficial changes in the structure may affect the order or dependence of aspects. In Chapter 5, we observe that even the best summarization model often has trouble generating summaries that are most similar to the target summary (MRR).

To improve controllability, recent work has designed prompts optimized for respective tasks (Fan et al., 2018; Kikuchi et al., 2016; Yu et al., 2020). Alternatively, one could also enforce the aspect controllability via multi-objective optimization; additional objective functions to promote aspect controllability.

Improving Faithfulness of Generated Text A long standing challenge in NLG is to preserve faithfulness in the generated text (Kryscinski et al., 2020; Pagnoni et al., 2021), which also applies to aspect-based NLG. While a number of ongoing attempts have already been made, we believe the framework of aspect-based NLG could further assist faithful generation. In Chapter 6, we discussed how fine-grained aspects in the form of a knowledge base can guide the generation to be more factually consistent. Similarly, a more faithful generation model could be achieved by additional constraints in terms of what specific contents are allowed to generate. Another perspective which has recently been investigated by (Rashkin et al., 2021) is to enrich the prompt with more qualifications (*e.g.* third-person, high lexical overlap to the source), which reported more faithful generation results. Connecting to the previous challenge on controllability, enriched prompts that describe the output characteristics in this way might also be able to enforce the faithfulness.

Chapter 8

Bibliography

Amjad Abu-Jbara and Dragomir Radev. 2011. Coherent citation-based summarization of scientific papers. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 500–509, Portland, Oregon, USA. Association for Computational Linguistics. 5.7

Sungjin Ahn, Heeyoul Choi, Tanel Pärnamaa, and Yoshua Bengio. 2016. A neural knowledge language model. *CoRR*, arXiv:1608.00318. 6.1, 6.2, 6.5.1, 6.5.2, 6.6.2, 6.6.3

Reinald Kim Amplayo, Stefanos Angelidis, and Mirella Lapata. 2021. Unsupervised opinion summarization with content planning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12489–12497. 4.7

Reinald Kim Amplayo and Mirella Lapata. 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. 2.2.2

Stefanos Angelidis and Mirella Lapata. 2018. 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. (document), 1, 2.2.2, 3.3

Philip Arthur, Graham Neubig, and Satoshi Nakamura. 2016. Incorporating discrete translation lexicons into neural machine translation. In *EMNLP*, pages 1557–1567. 6.1

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735.

Springer. 3.4.2, 4.4

Alexei Baevski and Michael Auli. 2019. Adaptive input representations for neural language modeling. In *ICLR*. 6.6.2

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*. 1

Federico Barrios, Federico López, Luis Argerich, and Rosa Wachenchauzer. 2016. Variations of the similarity function of textrank for automated summarization. *CoRR*, abs/1602.03606. 3.5.2

Leonard E. Baum, Ted Petrie, George Soules, and Norman Weiss. 1970. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The Annals of Mathematical Statistics*, 41(1):164–171. 6.4.2

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*. 2.3.1

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *JMLR*, 3(Feb):1137–1155. 1, 6.1

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *TACL*, 5:135–146. 6.4.3

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *SIGMOD*, pages 1247–1250. 6.1, 6.5.1

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*. 4.1

Jacob Buckman and Graham Neubig. 2018. Neural lattice language models. *TACL*, 6:529–541. 6.2

Isabel Cachola, Kyle Lo, Arman Cohan, and Daniel Weld. 2020. TLDR: Extreme summarization of scientific documents. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4766–4777, Online. Association for Computational Linguistics. 5.1, 5.5.1, 5.7

Ziqiang Cao, Furu Wei, Li Dong, Sujian Li, and Ming Zhou. 2015. Ranking with recursive neural networks and its application to multi-document summarization. In *Twenty-ninth AAAI conference on artificial intelligence*. 3.2

Thiago Castro Ferreira, Iacer Calixto, Sander Wubben, and Emiel Krahmer. 2017. Linguistic realisation as machine translation: Comparing different MT models for AMR-to-text generation. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 1–10, Santiago de Compostela, Spain. Association for Computational Linguistics. 1

Diego Ceccarelli, Claudio Lucchese, Salvatore Orlando, Raffaele Perego, and Salvatore Trani. 2013. Learning relatedness measures for entity linking. In *CIKM*, pages 139–148. 6.1

Muthu Kumar Chandrasekaran, Michihiro Yasunaga, Dragomir Radev, Dayne Freitag, and Min-Yen Kan. 2019. Overview and results: Cl-scisumm shared task 2019. *arXiv preprint arXiv:1907.09854*. 5.7

David L. Chen and Raymond J. Mooney. 2008. Learning to sportscast: A test of grounded language acquisition. In *ICML*, pages 128–135. 6.2

Yen-Chun Chen and Mohit Bansal. 2018. Fast abstractive summarization with reinforce-selected sentence rewriting. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 675–686, Melbourne, Australia. Association for Computational Linguistics. 4.1

Yiran Chen, Pengfei Liu, Ming Zhong, Zi-Yi Dou, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2020. CDEvalSumm: An empirical study of cross-dataset evaluation for neural summarization systems. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3679–3691, Online. Association for Computational Linguistics. 5.3.3

Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 484–494, Berlin, Germany. Association for Computational Linguistics. 2.2.2

Sumit Chopra, Michael Auli, and Alexander M. Rush. 2016. Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 93–98, San Diego, California. Association for Computational Linguistics. 2.2.2

Eric Chu and Peter Liu. 2019. Meansum: a neural model for unsupervised multi-document abstractive summarization. In *International Conference on Machine Learning*, pages 1223–1232. PMLR. 4.7

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 615–621, New Orleans, Louisiana. Association for Computational Linguistics. 4.7, 5.1, 5.3, 5.7

Arman Cohan and Nazli Goharian. 2017. Scientific article summarization using citation-context and article's discourse structure. *arXiv preprint arXiv:1704.06619*. 5.7

Danish Contractor, Yufan Guo, and Anna Korhonen. 2012. Using argumentative zones for extractive summarization of scientific articles. In *Proceedings of COLING 2012*, pages 663–678. 5.1

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive language models beyond a fixed-length context. In *ACL*, pages 2978–2988, Florence, Italy. 2.3.1, 6.6.2, 6.6.3

Hoa Trang Dang. 2005. Overview of duc 2005. In *Proceedings of the document understanding conference*, volume 2005, pages 1–12. 2.2.3

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2020. Plug and play language models: A simple approach to controlled text generation. In *International Conference on Learning Representations*. 2.2.2, 4.3.2

Shrey Desai, Jiacheng Xu, and Greg Durrett. 2020. Compressive summarization with plausibility and salience modeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6259–6274, Online. Association for Computational Linguistics. 4.7

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics. 1, 2.3.1, 3.7.1

Zi-Yi Dou, Pengfei Liu, Hiroaki Hayashi, Zhengbao Jiang, and Graham Neubig. 2021. GSum: A general framework for guided neural abstractive summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4830–4842, Online. Association for Computational Linguistics. 5.4, 5.5.3, 5.5.4

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. 2018. The hitchhiker's guide to testing statistical significance in natural language processing. In *ACL*, pages 1383–1392, Melbourne, Australia. (document), 6.2

Jeffrey L Elman. 1990. Finding structure in time. *Cognitive science*, 14(2):179–211. 2.3.1

Alexander Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. 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. 3.2, 5.3.3

Alexander R Fabbri, Simeng Han, Haoyuan Li, Haoran Li, Marjan Ghazvininejad, Shafiq Joty, Dragomir Radev, and Yashar Mehdad. 2020. Improving zero and few-shot abstractive summarization with intermediate fine-tuning and data augmentation. *arXiv preprint arXiv:2010.12836*. 4.1, 4.2, 4.7

Angela Fan, Claire Gardent, Chloé Braud, and Antoine Bordes. 2019. Using local knowledge graph construction to scale Seq2Seq models to multi-document inputs. 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 4186–4196, Hong Kong, China. Association for Computational Linguistics. 3.1

Angela Fan, David Grangier, and Michael Auli. 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. 1, 2.2.2, 7.2

Mariano Felice, Zheng Yuan, Øistein E. Andersen, Helen Yannakoudakis, and Ekaterina Kochmar. 2014. Grammatical error correction using hybrid systems and type filtering. In *CoNLL*, pages 15–24. 6.1

Lea Frermann and Alexandre Klementiev. 2019. Inducing Document Structure for Aspect-based Summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6263–6273, Florence, Italy. Association for Computational Linguistics. (document), 1, 2.2.2, 3.3, 5.5.3

Michael Gamon, Anthony Aue, Simon Corston-Oliver, and Eric Ringger. 2005. Pulse: Mining customer opinions from free text. In *international symposium on intelligent data analysis*, pages 121–132. Springer. 2.2.2

Octavian-Eugen Ganea and Thomas Hofmann. 2017. Deep joint entity disambiguation with local neural attention. In *EMNLP*, pages 2619–2629, Copenhagen, Denmark. 6.1

Albert Gatt and Emiel Krahmer. 2018. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research*, 61:65–170. 1

Sebastian Gehrmann, Yuntian Deng, and Alexander Rush. 2018. Bottom-up abstractive summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4098–4109, Brussels, Belgium. Association for Computational Linguistics. 2.2.2

Eli Goldberg, Norbert Driedger, and Richard I Kittredge. 1994. Using natural-language processing to produce weather forecasts. *IEEE Expert*, 9(2):45–53. 6.2

Philip John Gorinski and Mirella Lapata. 2015. Movie script summarization as graph-based scene extraction. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1066–1076, Denver, Colorado. Association for Computational Linguistics. 3.1

Édouard Grave, Armand Joulin, Moustapha Cissé, David Grangier, and Hervé Jégou. 2017. Efficient softmax approximation for GPUs. In *ICML*, volume 70 of *Proceedings of Machine Learning Research*, pages 1302–1310, International Convention Centre, Sydney, Australia. 6.6.2

Barbara Jean Grosz. 1977. *The representation and use of focus in dialogue understanding*. University of California, Berkeley. 2.2.1

Max Grusky, Mor Naaman, and Yoav Artzi. 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. Association for Computational Linguistics. 3.1

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *ACL*, pages 1631–1640, Berlin, Germany. 6.1, 6.2

Xu Han, Shulin Cao, Xin Lv, Yankai Lin, Zhiyuan Liu, Maosong Sun, and Juanzi Li. 2018. OpenKE: An open toolkit for knowledge embedding. In *EMNLP*, pages 139–144. 6.4.3

Junxian He, Wojciech Kryściński, Bryan McCann, Nazneen Rajani, and Caiming Xiong. 2020. Ctrlsum: Towards generic controllable text summarization. *arXiv preprint arXiv:2012.04281*. 2.2.2, 4.3.2

Lei He, Wei Li, and Hai Zhuge. 2016. Exploring differential topic models for comparative summarization of scientific papers. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1028–1038, Osaka, Japan. The COLING 2016 Organizing Committee. 5.7

Cong Duy Vu Hoang and Min-Yen Kan. 2010. Towards automated related work summarization. In *Coling 2010: Posters*, pages 427–435, Beijing, China. Coling 2010 Organizing Committee. 5.7

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780. 2.3.1, 6.6.2

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177. 2.2.2

Yue Hu and Xiaojun Wan. 2014. Automatic generation of related work sections in scientific papers: An optimization approach. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1624–1633, Doha, Qatar. Association for Computational Linguistics. 5.7

Xinyu Hua and Lu Wang. 2017. A pilot study of domain adaptation effect for neural abstractive summarization. In *Proceedings of the Workshop on New Frontiers in Summarization*, pages 100–106, Copenhagen, Denmark. Association for Computational Linguistics. 4.7

Hen-Hsen Huang. 2020. Autosurvey: Automatic survey generation based on a research draft. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 5255–5257. International Joint Conferences on Artificial Intelligence Organization. 5.7

Rahul Jha, Amjad Abu-Jbara, and Dragomir Radev. 2013. A system for summarizing scientific topics starting from keywords. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 572–577, Sofia, Bulgaria. Association for Computational Linguistics. 5.7

Rahul Jha, Reed Coke, and Dragomir Radev. 2015a. Surveyor: a system for generating coherent survey articles for scientific topics. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 2167–2173. 5.1, 5.3.2, 5.5.1, 5.7

Rahul Jha, Catherine Finegan-Dollak, Ben King, Reed Coke, and Dragomir Radev. 2015b. Content models for survey generation: A factoid-based evaluation. 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 441–450, Beijing, China. Association for Computational Linguistics. 5.7

Rahul Jha, Catherine Finegan-Dollak, Ben King, Reed Coke, and Dragomir Radev. 2015c. Content models for survey generation: a factoid-based evaluation. 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 441–450. 5.7

Yangfeng Ji, Chenhao Tan, Sebastian Martschat, Yejin Choi, and Noah A. Smith. 2017. Dynamic entity representations in neural language models. In *EMNLP*, pages 1830–1839, Copenhagen, Denmark. 6.2

Dongyeop Kang, Waleed Ammar, Bhavana Dalvi, Madeleine van Zuylen, Sebastian Kohlmeier, Eduard Hovy, and Roy Schwartz. 2018. A dataset of peer reviews (PeerRead): Collection, insights and NLP applications. 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 1647–1661, New Orleans, Louisiana. Association for Computational Linguistics. 3.1

Chris Kedzie, Kathleen McKeown, and Hal Daumé III. 2018. Content selection in deep learning models of summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1818–1828, Brussels, Belgium. Association for Computational Linguistics. 3.1

Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*. 2.2.2, 4.3.2, 5.5.3

Chloé Kiddon, Luke Zettlemoyer, and Yejin Choi. 2016. Globally coherent text generation with neural checklist models. In *EMNLP*, pages 329–339. 6.2

Yuta Kikuchi, Graham Neubig, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. 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. 2.2.2, 4.6.4, 7.2

Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer. In *International Conference on Learning Representations*. 2.3.1

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages

388–395, Barcelona, Spain. Association for Computational Linguistics. 4.5.3

Ioannis Konstas and Mirella Lapata. 2013. A global model for concept-to-text generation. *Journal of Artificial Intelligence Research*, 48:305–346. 6.1

Kundan Krishna and Balaji Vasan Srinivasan. 2018. Generating Topic-Oriented Summaries Using Neural Attention. 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 1697–1705, New Orleans, Louisiana. Association for Computational Linguistics. 1, 2.2.2

Wojciech Kryscinski, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Neural text summarization: A critical evaluation. 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 540–551, Hong Kong, China. Association for Computational Linguistics. 5.5.3

Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346, Online. Association for Computational Linguistics. 5.6.2, 7.2

Logan Lebanoff, Kaiqiang Song, and Fei Liu. 2018. Adapting the Neural Encoder-Decoder Framework from Single to Multi-Document Summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4131–4141, Brussels, Belgium. Association for Computational Linguistics. 3.2

Rémi Lebret, David Grangier, and Michael Auli. 2016. Neural text generation from structured data with application to the biography domain. In *EMNLP*, pages 1203–1213. 1, 6.1

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics. 1, 2.3.1, 4.1, 4.5.2, 5.4, 5.7

Chenliang Li, Weiran Xu, Si Li, and Sheng Gao. 2018. Guiding generation for abstractive text summarization based on key information guide network. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 55–60, New Orleans, Louisiana. Association for Computational Linguistics. 2.2.2

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics. 3.6, 5.5.2

Wang Ling, Phil Blunsom, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiský, Fumin Wang, and Andrew Senior. 2016. Latent predictor networks for code generation. In *ACL*, pages 599–609. 6.1, 6.2, 6.4.2

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*. 2.3.1

Peter J. Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Łukasz Kaiser, and Noam Shazeer. 2018a. Generating Wikipedia by summarizing long sequences. In *ICLR*. 3, 3.1, 3.2, 3.3, 3.4.1, 4.1, 4.4, 4.5.3, 4.7

Tianyu Liu, Kexiang Wang, Lei Sha, Baobao Chang, and Zhifang Sui. 2018b. Table-to-text generation by structure-aware seq2seq learning. *AAAI*. 6.2

Yang Liu and Mirella Lapata. 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. 3.1

Yang Liu and Mirella Lapata. 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. 3.5.2, 5.5.3

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv:1907.11692 [cs]*. 3.5.1

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel S. Weld. 2020. S2ORC: The Semantic Scholar Open Research Corpus. In *Proceedings of ACL*. 5.3

Robert Logan, Nelson F. Liu, Matthew E. Peters, Matt Gardner, and Sameer Singh. 2019. Barack's wife Hillary: Using knowledge graphs for fact-aware language modeling. In *ACL*, pages 5962–5971, Florence, Italy. 6.1, 6.2

Yao Lu, Yue Dong, and Laurent Charlin. 2020. Multi-XScience: A large-scale dataset for extreme multi-document summarization of scientific articles. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8068–8074, Online. Association for Computational Linguistics. 5.1, 5.3.3, 5.7

Yue Lu, ChengXiang Zhai, and Neel Sundaresan. 2009. Rated aspect summarization of short comments. In *Proceedings of the 18th International Conference on World Wide Web - WWW '09*, page 131, Madrid, Spain. ACM Press. 2.2.2

Minh-Thang Luong and Christopher D. Manning. 2016. Achieving open vocabulary neural machine translation with hybrid word-character models. In *ACL*, pages 1054–1063. 6.4.3

Kelvin Luu, Xinyi Wu, Rik Koncel-Kedziorski, Kyle Lo, Isabel Cachola, and Noah A. Smith. 2021. Explaining relationships between scientific documents. 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 2130–2144, Online. Association for Computational Linguistics. 5.7

Ahmed Magooda and Diane Litman. 2020. Abstractive summarization for low resource data using domain transfer and data synthesis. In *The Thirty-Third International Flairs Conference*. 4.7

Kathleen McKeown. 1983. Focus constraints on language generation. 2.2.1

Kathleen McKeown. 1992. *Text generation*. Cambridge University Press. 2.2.1

Qiaozhu Mei, Xu Ling, Matthew Wondra, Hang Su, and ChengXiang Zhai. 2007. Topic sentiment mixture: modeling facets and opinions in weblogs. In *Proceedings of the 16th international conference on World Wide Web*, pages 171–180. 2.2.2

Stephen Merity, Nitish Shirish Keskar, and Richard Socher. 2017a. Regularizing and optimizing LSTM language models. *CoRR*, arXiv:1708.02182. 6.6.3

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017b. Pointer sentinel mixture models. In *ICLR*. 6.1, 6.5.2, 6.6.2

Yishu Miao and Phil Blunsom. 2016. Language as a latent variable: Discrete generative models for sentence compression. In *EMNLP*, pages 319–328, Austin, Texas. 4.7, 6.2

Paul Michel and Graham Neubig. 2018. Extreme adaptation for personalized neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 312–318, Melbourne, Australia. Association for Computational Linguistics. 2.2.2, 4.3.2

Rada Mihalcea and Paul Tarau. 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. 3.5.2, 5.5.3

Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *INTERSPEECH*. 6.1

George A. Miller. 1995. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41. 4.3.1

Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *ACL*, pages 1003–1011, Suntec, Singapore. 6.5.2

Saif Mohammad, Bonnie Dorr, Melissa Egan, Ahmed Hassan, Pradeep Muthukrishan, Vahed Qazvinian, Dragomir Radev, and David Zajic. 2009. Using citations to generate surveys of scientific paradigms. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 584–592, Boulder, Colorado. Association for Computational Linguistics. 5.7

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany. Association for Computational Linguistics. 2.2.2, 3.1, 4.5.2, 4.7, 5.3.3, 5.3.3

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics. 5.3.3

Ani Nenkova, Lucy Vanderwende, and Kathleen McKeown. 2006. A compositional context sensitive multi-document summarizer: exploring the factors that influence summarization. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 573–580. ACM. 3.2

Graham Neubig and Chris Dyer. 2016. Generalizing and hybridizing count-based and neural language models. In *EMNLP*, pages 1163–1172, Austin, Texas. 6.1

Artidoro Pagnoni, Vidhisha Balachandran, and Yulia Tsvetkov. 2021. Understanding factuality in abstractive summarization with FRANK: A benchmark for factuality metrics. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4812–4829, Online. Association for Computational Linguistics. 5.6.2, 7.2

Daraksha Parveen, Mohsen Mesgar, and Michael Strube. 2016. Generating coherent summaries of scientific articles using coherence patterns. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 772–783, Austin, Texas. Association for Computational Linguistics. 5.7

Md Rizwan Parvez, Saikat Chakraborty, Baishakhi Ray, and Kai-Wei Chang. 2018. Building language models for text with named entities. In *ACL*, pages 2373–2383, Melbourne, Australia. 6.1, 6.2

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in PyTorch. 6.6.2

Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A deep reinforced model for abstractive summarization. In *International Conference on Learning Representations*. 2.2.2

Jian Pei. 2020. A survey on data pricing: from economics to data science. *IEEE Transactions on Knowledge and Data Engineering*. 1

Jeffrey Pennington, Richard Socher, and Christopher Manning. 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. 2

Laura Perez-Beltrachini, Yang Liu, and Mirella Lapata. 2019. Generating Summaries with Topic Templates and Structured Convolutional Decoders. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5107–5116, Florence, Italy. Association for Computational Linguistics. 1, 2.2.2, 3.1, 3.2, 3.4.2

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *NAACL*, pages 2227–2237. 6.1

Ana-Maria Popescu and Orena Etzioni. 2007. Extracting product features and opinions from reviews. In *Natural language processing and text mining*, pages 9–28. Springer. 2.2.2

Vahed Qazvinian and Dragomir R. Radev. 2008. Scientific paper summarization using citation summary networks. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 689–696, Manchester, UK. Coling 2008 Organizing Committee.

5.7

Vahed Qazvinian, Dragomir R Radev, Saif M Mohammad, Bonnie Dorr, David Zajic, Michael Whidby, and Taesun Moon. 2013. Generating extractive summaries of scientific paradigms. *Journal of Artificial Intelligence Research*, 46:165–201. 5.7

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. 2.3.1

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *Preprint*. 6.3.2

Jack W Rae, Anna Potapenko, Siddhant M Jayakumar, Chloe Hillier, and Timothy P Lillicrap. 2020. Compressive transformers for long-range sequence modelling. In *International Conference on Learning Representations*. 2.3.1

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67. 2.2.2

Hannah Rashkin, David Reitter, Gaurav Singh Tomar, and Dipanjan Das. 2021. Increasing faithfulness in knowledge-grounded dialogue with controllable features. 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 704–718, Online. Association for Computational Linguistics. 7.2

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. CoQA: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266. 2.2.3

Ehud Reiter, Somayajulu Sripada, Jim Hunter, Jin Yu, and Ian Davy. 2005. Choosing words in computer-generated weather forecasts. *Artificial Intelligence*, 167(1-2):137–169. 6.2

Alexander M Rush, Sumit Chopra, and Jason Weston. 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. 2.2.2

Gerard Salton and Michael J. McGill. 1986. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA. 6.1

Evan Sandhaus. 2008. The new york times annotated corpus. *Linguistic Data Consortium, Philadelphia*, 6(12):e26752. 4.7

Christina Sauper and Regina Barzilay. 2009. Automatically Generating Wikipedia Articles: A Structure-Aware Approach. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 208–216, Suntec, Singapore. Association for Computational Linguistics. 3.2

Raphael Schumann. 2018. Unsupervised abstractive sentence summarization using length controlled variational autoencoder. *arXiv preprint arXiv:1809.05233*. 4.7

Abigail See, Peter J Liu, and Christopher D Manning. 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. 2.2.2

Eva Sharma, Chen Li, and Lu Wang. 2019. BIGPATENT: A large-scale dataset for abstractive and coherent summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2204–2213, Florence, Italy. Association for Computational Linguistics. 4.7

Benjamin Snyder and Regina Barzilay. 2007. Multiple aspect ranking using the good grief algorithm. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 300–307. 2.2.2

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31. 4.3.1

Dan Su, Yan Xu, Tiezheng Yu, Farhad Bin Siddique, Elham Barezi, and Pascale Fung. 2020. CAiRE-COVID: A question answering and query-focused multi-document summarization system for COVID-19 scholarly information management. In *Proceedings of the 1st Workshop on NLP for COVID-19 (Part 2) at EMNLP 2020*, Online. Association for Computational Linguistics. 2.2.3

Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. 2012. LSTM neural networks for language modeling. In *INTERSPEECH*. 6.1

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *NeurIPS*, pages 3104–3112. 1, 2.2.2

Bowen Tan, Lianhui Qin, Eric Xing, and Zhiting Hu. 2020. Summarizing text on any aspects: A knowledge-informed weakly-supervised approach. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6301–6309, Online.

Association for Computational Linguistics. 4.1, 4.2

Simone Teufel and Marc Moens. 2002. Summarizing scientific articles: experiments with relevance and rhetorical status. *Computational linguistics*, 28(4):409–445. 5.1

Ivan Titov and Ryan McDonald. 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. 2.2.2

Joerg Ueberla. 1994. Analysing a simple language model· some general conclusions for language models for speech recognition. *Computer Speech & Language*, 8(2):153–176. 4

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*, pages 5998–6008. 2.3.1, 2.3.1, 2.3.1

Ellen M Voorhees. 2001. The TREC question answering track. *Natural Language Engineering*, 7(4):361–378. 5.5.2

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: A free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85. 6.1, 6.5.2

Danqing Wang, Pengfei Liu, Ming Zhong, Jie Fu, Xipeng Qiu, and Xuanjing Huang. 2019. Exploring domain shift in extractive text summarization. *arXiv preprint arXiv:1908.11664*. 4.1, 4.2, 4.7

Lu Wang and Wang Ling. 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. (document), 1, 2.2.2, 3.3

Qingyun Wang, Xiaoman Pan, Lifu Huang, Boliang Zhang, Zhiying Jiang, Heng Ji, and Kevin Knight. 2018. Describing a knowledge base. In *INLG*, pages 10–21. 6.1, 6.2

Ziyang Wang, Shuhan Zhou, Nuo Xu, Bei Li, Yinqiao Li, Quan Du, Tong Xiao, and Jingbo Zhu. 2021. ABigSurvey: A survey of surveys (NLP & ML). 5.3.1

Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically conditioned LSTM-based natural language generation for spoken dialogue systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1711–1721, Lisbon, Portugal. Association for Computational Linguistics. 1

Sam Wiseman, Stuart Shieber, and Alexander Rush. 2018. Learning neural templates for text generation. In *EMNLP*, pages 3174–3187, Brussels, Belgium. 6.2

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771. 7

Xinyu Xing, Xiaosheng Fan, and Xiaojun Wan. 2020. Automatic generation of citation texts in scholarly papers: A pilot study. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6181–6190, Online. Association for Computational Linguistics. 5.7

Huiyan Xu, Zhijian Wang, and Xiaolan Weng. 2019. Scientific literature summarization using document structure and hierarchical attention model. *IEEE Access*, 7:185290–185300. 5.7

Yumo Xu and Mirella Lapata. 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. 2.2.3

Min Yang, Qiang Qu, Ying Shen, Qiao Liu, Wei Zhao, and Jia Zhu. 2018. Aspect and sentiment aware abstractive review summarization. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1110–1120, Santa Fe, New Mexico, USA. Association for Computational Linguistics. (document), 2.2.2, 3.3

Shansong Yang, Weiming Lu, Dezhi Yang, Xi Li, Chao Wu, and Baogang Wei. 2017. KeyphraseDS: Automatic generation of survey by exploiting keyphrase information. *Neurocomputing*, 224:58–70. 5.7

Ziyi Yang, Chenguang Zhu, Robert Gmyr, Michael Zeng, Xuedong Huang, and Eric Darve. 2020. TED: A pretrained unsupervised summarization model with theme modeling and denoising. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1865–1874, Online. Association for Computational Linguistics. 4.7

Michihiro Yasunaga, Jungo Kasai, Rui Zhang, Alexander R Fabbri, Irene Li, Dan Friedman, and Dragomir R Radev. 2019. Scisummnet: A large annotated corpus and content-impact models for scientific paper summarization with citation networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7386–7393. 5.7

Michihiro Yasunaga, Rui Zhang, Kshitijh Meelu, Ayush Pareek, Krishnan Srinivasan, and Dragomir Radev. 2017. Graph-based Neural Multi-Document Summarization. In *Proceed-*

ings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), pages 452–462, Vancouver, Canada. Association for Computational Linguistics. 3.2

Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. 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 3914–3923, Hong Kong, China. Association for Computational Linguistics. 4.3.1, 4.5.2

Adams Wei Yu, David Dohan, Quoc Le, Thang Luong, Rui Zhao, and Kai Chen. 2018. Fast and accurate reading comprehension by combining self-attention and convolution. In *International Conference on Learning Representations*. 2.3.1

Tiezheng Yu, Zihan Liu, and Pascale Fung. 2021. AdaptSum: Towards low-resource domain adaptation for abstractive summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5892–5904, Online. Association for Computational Linguistics. 4.7

Wenhao Yu, Chenguang Zhu, Zaitang Li, Zhiting Hu, Qingyun Wang, Heng Ji, and Meng Jiang. 2020. A survey of knowledge-enhanced text generation. *arXiv preprint arXiv:2010.04389*. 7.2

Jianmin Zhang, Jiwei Tan, and Xiaojun Wan. 2018. Towards a neural network approach to abstractive multi-document summarization. *arXiv preprint arXiv:1804.09010*. 3.2

Jingqing Zhang, Piyawat Lertvittayakumjorn, and Yike Guo. 2019a. Integrating semantic knowledge to tackle zero-shot text classification. 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 1031–1040, Minneapolis, Minnesota. Association for Computational Linguistics. 4.3.1

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J Liu. 2019b. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. *arXiv preprint arXiv:1912.08777*. 4.1, 4.2, 5.7

Hao Zheng and Mirella Lapata. 2019. Sentence centrality revisited for unsupervised summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6236–6247, Florence, Italy. Association for Computational Linguistics. 4.1, 4.3.1

Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2020. Extractive summarization as text matching. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online. Association for Computational Linguistics.

5.7

Jiawei Zhou and Alexander Rush. 2019. Simple unsupervised summarization by contextual matching. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5101–5106, Florence, Italy. Association for Computational Linguistics. 4.7

Haichao Zhu, Li Dong, Furu Wei, Bing Qin, and Ting Liu. 2019. Transforming wikipedia into augmented data for query-focused summarization. *arXiv preprint arXiv:1911.03324*. 3.2

George Kingsley Zipf. 1949. *Human behavior and the principle of least effort: An introduction to human eoclogy*. Addison-Wesley Press. 6.1

Appendix A

Additional Results for Chapter 3

A.1 Domain Statistics

Domain	Train	Valid	Test
Album	24434	3104	3038
Animal	16540	2005	2007
Artist	26754	3194	3329
Building	20449	2607	2482
Company	24353	2946	3029
EducationalInstitution	17634	2141	2267
Event	6475	807	828
Film	32129	4014	3981
Group	11966	1462	1444
HistoricPlace	4919	601	600
Infrastructure	17226	1984	2091
MeanOfTransportation	9277	1215	1170
OfficeHolder	18177	2218	2333
Plant	6107	786	774
Single	14217	1734	1712
SoccerPlayer	17599	2150	2280
Software	13516	1637	1638
TelevisionShow	8717	1128	1072
Town	14818	1911	1831
WrittenWork	15065	1843	1931

Table A.1: The list of domains and the number of Wikipedia articles in each domain that contain at least one salient aspect.

A.2 Additional Samples

Title: Recomposed by Max Richter: Vivaldi – The Four Seasons
Aspect: *Critical Reception*
Gold: recomposed by max richter : vivaldi - the four seasons received widespread acclaim from contemporary classical music critics . ivan hewett of the telegraph gave the album a very positive review , stating , " as you would expect of a composer who once studied with the great modernist luciano berio , richter is very self - aware .. . . **Ext.**: listen to recomposed by max richter : vivaldi , the four seasons now . i am highly impressed with ‘ recomposed ’ . the music then propels the audience into an atmosphere of isolation ; a delicate harmony that is sustained whilst hope takes centre stage . . . . **Abs.**: the allmusic review by michael g . nastos awarded the album 4 stars stating “ this is an album that generally considered for fans of the genre “ . . . .

Table A.2: Generated summaries from **Album** domain.

Title: Pride and Glory (film)
Aspect: *Plot*
Gold: assistant chief francis tierney sr . is the head of a multigenerational new york city police department (nypd) family , which includes his sons francis " franny " jr . , ray , and his son - in - law jimmy egan . deputy inspector franny is the commanding officer of the 31st precinct , where sergeant jimmy is a patrol officer , . . . **Ext.**: as we know , under the macho code , this means that after two people who love each other end up beaten and bloody , they will somehow arrive at a catharsis . the plot involves how and why the four cops were killed . a family of police officers - patriarch , two sons , and a son - in - law - deals with corruption in a precinct in washington heights . . . . **Abs.**: in the year before the events of the first film , the movie takes place in washington heights , d . c . , a . army sergeant - in - law , ray ’ s wife , and sister abby , living in washington city . they have a romantic relationship with one of their officers . while the four officers are called to “ the mental patient “ , . . .

Table A.3: Generated summaries from **Film** domain.

Title: Dimitri Soudas

Aspect: *Career*

Gold: soudas served for one term as a school trustee at the western quebec school board from 2002 to 2005 . between 2006 and 2011 , soudas was a " high profile " member of prime minister stephen harper ’ s communication team , and one of the prime minister ’ s " closest and most faithful aides . " initially serving as a press secretary and later as an associate director of communications for the prime minister ’ s office , . . .

Ext.: april 2010 – after serving as a press secretary in the prime minister ’ s office , soudas was promoted to director of communications . " to fulfil the opportunities afforded by social media , directors of communication need to be aware of this trend and engage with it , " dimitri soudas writes in his master ’ s thesis , a copy of which has been obtained by cbc news . . . .

Abs.: in 2001 , he was elected to the canadian house of commons as a member of the people ’ s action party (pc) for the riding of yorkshire . he was re - elected in 2002 and 2006 . in 2006 , he was .

Table A.4: Generated summaries from **OfficeHolder** domain.

A.3 Aspect Statistics

Table A.5 and A.6 shows aspect frequency statistics. Perf., hist., dist., ext., desc., dev., edu., nm., and intl. correspond to performance, history, distribution, extracurricular, description, development, education, naming, and international, respectively.

Album		Animal	
reception	11782	description	12729
critical reception	6682	distribution	7813
background	6202	dist. & habitat	2967
commercial perf.	2398	taxonomy	2737
release	2209	habitat	2208
chart positions	1891	behavior	2167
recording	1490	ecology	1777
promotion	1150	diet	1363
history	1045	reproduction	1291
overview	840	biology	1238
Artist		**Building**	
career	10193	history	16885
biography	8292	architecture	3223
early life	7587	desc. & hist.	1395
personal life	6775	description	1382
music career	2829	location	906
death	1607	interior	877
life and career	1512	construction	862
early life & edu.	1239	exterior	746
early years	1129	design	623
exhibitions	1030	facilities	572
Company		**EducationalInstitution**	
history	21488	history	12798
products	2921	athletics	5602
operations	1630	academics	4638
services	1019	campus	2471
controversy	920	sports	1433
overview	891	student life	1327
background	572	ext. activities	1227
subsidiaries	556	curriculum	1191
company history	504	facilities	1189
technology	471	rankings	836
Event		**Film**	
background	3453	plot	25772
aftermath	2483	reception	14003
history	1361	production	13882
battle	1228	release	7299
format	461	box office	4572
prelude	450	critical reception	4195
event	416	critical response	2802
report	323	synopsis	2626
summary	321	home media	2461
casualties	290	filming	2013

Table A.5: Aspect frequency for 8 domains.

Group		HistoricPlace	
history	8894	history	3232
biography	1206	description	1398
career	1102	desc. & hist.	1250
musical style	683	heritage listing	942
background	581	architecture	549
formation	408	location	161
early years	279	historic uses	90
legacy	272	preservation	84
style	265	geography	75
influences	204	interior	70
MeanOfTransportation		OfficeHolder	
history	2572	personal life	5119
design	2152	political career	4950
operational hist.	1989	early life	4740
design & dev.	1566	career	4115
service history	1435	biography	2801
development	1096	education	2168
construction	933	background	1578
fate	632	death	1402
background	604	legacy	889
description	602	early life & career	859
Plant		Single	
description	4684	music video	9606
dist. & habitat	1649	critical reception	3829
uses	1585	background	3459
distribution	1399	reception	2097
cultivation	1387	composition	1729
taxonomy	1121	cover versions	1594
ecology	884	content	1266
conservation	554	release	1045
etymology	389	commercial perf.	849
taxonomy & nm.	384	live performance	113
SoccerPlayer		TelevisionShow	
intl. career	8055	plot	2902
club career	8029	production	2648
career	6386	reception	2643
personal life	3621	synopsis	1304
playing career	1930	premise	944
early career	1578	history	908
early life	1191	format	842
professional	992	broadcast	779
style of play	887	overview	650
football career	550	critical reception	583
Town		WrittenWork	
geography	12667	plot	5495
demographics	10949	reception	4970
history	7298	plot summary	3900
education	2868	history	2527
government	1910	background	1218
2010 census	1363	adaptations	1173
2000 census	1284	critical reception	933
transportation	1239	manga	830
economy	1066	history and profile	803
name and history	1002	anime	714

Table A.6: Aspect frequency for 10 domains.