Abstract

Text Summarization Across
High and Low-Resource Settings

Alexander R. Fabbri

2021

Natural language processing aims to build automated systems that can both understand and generate natural language textual data. As the amount of textual data available online has increased exponentially, so has the need for intelligence systems to comprehend and present it to the world. As a result, automatic text summarization, the process by which a text's salient content is automatically distilled into a concise form, has become a necessary tool.

Automatic text summarization approaches and applications vary based on the input summarized, which may constitute single or multiple documents of different genres. Furthermore, the desired output style may consist of a sentence or sub-sentential units chosen directly from the input in extractive summarization or a fusion and paraphrase of the input document in abstractive summarization. Despite differences in the above use-cases, specific themes, such as the role of large-scale data for training these models, the application of summarization models in real-world scenarios, and the need for adequately evaluating and comparing summaries, are common across these settings.

This dissertation presents novel data and modeling techniques for deep neural network-based summarization models trained across high-resource (thousands of supervised training examples) and low-resource (zero to hundreds of supervised training examples) data settings and a comprehensive evaluation of the model and metric progress in the field. We examine both Recurrent Neural Network (RNN)-based and Transformer-based models to extract and generate summaries from the input. To facilitate the training of large-scale networks, we introduce datasets applicable for multi-document summarization (MDS) for pedagogical applications and for news summarization. While the high-resource settings allow models to

advance state-of-the-art performance, the failure of such models to adapt to settings outside of that in which it was initially trained requires smarter use of labeled data and motivates work in low-resource summarization. To this end, we propose unsupervised learning techniques for both extractive summarization in question answering, abstractive summarization on distantly-supervised data for summarization of community question answering forums, and abstractive zero and few-shot summarization across several domains. To measure the progress made along these axes, we revisit the evaluation of current summarization models.

In particular, this dissertation addresses the following research objectives:

1) High-resource Summarization. We introduce datasets for multi-document summarization, focusing on pedagogical applications for NLP, news summarization, and Wikipedia topic summarization. Large-scale datasets allow models to achieve state-of-the-art performance on these tasks compared to prior modeling techniques, and we introduce a novel model to reduce redundancy. However, we also examine how models trained on these large-scale datasets fare when applied to new settings, showing the need for more generalizable models.

2) Low-resource Summarization. While high-resource summarization improves model performance, for practical applications, data-efficient models are necessary. We propose a pipeline for creating synthetic training data for training extractive question-answering models, a form of query-based extractive summarization with short-phrase summaries. In other work, we propose an automatic pipeline for training a multi-document summarizer in answer summarization on community question-answering forums without labeled data. Finally, we push the boundaries of abstractive summarization model performance when little or no training data is available across several domains.

3) Automatic Summarization Evaluation. To understand the extent of progress made across recent modeling techniques and better understand the current evaluation protocols, we examine the current metrics used to compare summarization output quality across 12 metrics across 23 deep neural network models and propose better-motivated summarization evaluation guidelines as well as point to open problems in summarization evaluation.

Text Summarization Across

High and Low-Resource Settings

A Dissertation
Presented to the Faculty of the Graduate School
of
Yale University
in Candidacy for the Degree of
Doctor of Philosophy

by
Alexander R. Fabbri

Dissertation Director: Dragomir R. Radev

June 2021

Acknowledgments

First, I would like to thank my excellent advisor, Professor Dragomir Radev, for his support and tutelage throughout my Ph.D. I first met Drago while researching over the summer at Columbia in 2016 and became interested in the materials he was preparing for his class. I was thrilled at the possibility of beginning a Ph.D. at Yale with Drago. Drago provided the right mixture of direction based on his vast knowledge and freedom to explore and find my own paths. This thesis would not be possible without him.

I want to thank the members of my thesis committee. At the end of my freshman year at Columbia, I sent an email inquiring about research in the NLP group to Rebecca Passonneau, who directed me to Smaranda Muresan. I met with Smara at the beginning of my sophomore year and began small contributions to a project. Although I was completely inexperienced, Smara was always helpful, and I later worked more closely with her as I gained experience. I want to thank Robert Frank for his insightful conversations. His patience and experience always left me feeling more confident about my work and choice of directions following our conversations. I want to thank Nisheeth Vishnoi, whose class provided me with a deeper theoretical understanding and helped me think within a broader paradigm.

I want to thank others who were integral to my development and interest in NLP while at Columbia. I want to thank Professor Kata Gábor, with whom I did a research project while studying abroad for a semester in Paris in Spring 2016. This project was my first research experience that I led myself, under the guidance of Professor Gábor. Her willingness and patience to work through problems made a strong impression on me. I want to thank Owen

Rambow, with whom I had the opportunity to work during that summer. I always look forward to any opportunity to talk with him to hear his insight and advice. Of course, I must thank Professor Kathleen McKeown. During that same summer, meeting with Kathy and the larger NLP group increased my desire to pursue NLP work. A meeting with Kathy convinced me to apply to Ph.D. programs, and I am forever grateful for her advice.

I want to thank my internship mentors Professor Mona Diab, Xiaojian Wu, and Srini Iyer at Facebook AI; Bing Xiang, Zhiguo Wang, Patrick Ng, Ramesh Nallapati, and Xiaofei Ma at AWS AI.

I want to thank collaborators from various projects: Yijiao He, Swapnil Hingmire, Weitai Ting, Pong Trairatvorakul, Robert Tung, Caitlin Westerfield, and Wai Pan Wong from the AAN project; Griffin Adams, Garrett Bingham, Youngduck Choi, Javid Dadashkarimi, William Hu, Jungo Kasai, Suyi Li, Peter Liu, Sally Ma, Tomoe Mizutani, Yavuz Nuzumlali, Tianwei She, Sungrok Shim, Neha Verma, and Michihiro Yasunaga from various projects I did while at Yale; Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher from a collaboration with Salesforce. Simeng Han, Haoyuan Li, Haoran Li, Marjan Ghazvininejad, Shafiq Joty, and Yashar Mehdad from a collaboration with Facebook AI, Nanyang Technological University, and the Renmin University of China.

I want to thank fellow Ph.D. students in the LILY Lab Rui Zhang, Irene Li, and Tao Yu for collectively working throughout my Ph.D. years. I would like to particularly thank Tao Yu, who also graduated with me in the same year from Columbia. We began working together with Owen Rambow, and Tao introduced me to Drago and talked with me many times about applying for Ph.D. programs while at Columbia and our progress during the past few years.

Finally, I want to thank my family. Thank you to my brothers Sonny and Joey and my sister April for always supporting me and helping distract me from work, especially during the last year. Thank you, mom and dad, for your endless support and encouragement.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Automatic Text Summarization

Automatic text summarization is a necessary tool within Natural Language Processing (NLP) to make sense of the growing availability of online data found as text. Text summarization was defined by Maybury (1999) as the "process of distilling the most important information from a source (or sources) to produce an abridged version for a particular user (or users) and task (or tasks)." Text summarization thus may vary along several axes, depending on the particular input and desired output.

The input to be summarized may consist of a single source, single-document summarization, or multiple documents in multi-document summarization. These two subtasks entail different problems; multi-document summarization requires an understanding and condensing of often redundant information and modeling components designed to reduce redundancy in the output. Furthermore, summaries may differ in their output style. Extractive summaries consist of substrings of the source input, typically whole sentences. Extracting entire sentences from the input has the advantage of producing fully grammatical sentences, although sentences extracted may not fit together as a coherent summary. Abstractive summarization may also include substrings of the input text, but entire sentences

are not extracted; the output may consist of fused sentences or paraphrases. Abstractive summarization is seen as the more challenging task, and early summarization models were primarily extractive. While abstractive summaries may be more fluent, abstractive models are more prone to hallucinating or producing summaries that are not implied by the input text. This phenomenon is a large challenge in the use of abstractive summarization in production environments.

The variations in summarization settings also imply difficulties in evaluating summarization models; multiple summaries may validly represent the input text in a concise form. Typically, summarization model outputs are compared with a human-written reference summary or summaries. The model summary should cover the content described in the gold summary, and having multiple reference summaries increases the coverage of good summaries. However, datasets typically contain a single reference summary, and often the summary comparison is made on a lexical level, which does not capture paraphrases.

1.2 Objectives and Challenges

This dissertation addresses summarization along the dimension of training data size, dividing work between high-resource and low-resource settings, which refers to the amount of data used to train the summarization model. Recent work found specific data-efficient models to achieve results comparable to state-of-the-art with only supervised 1000 examples. We thus define high-resource settings as those with 1000 or more non-automatically-created, supervised training examples are available and low-resource settings as those in which zero or fewer than 1000 such examples are available. We address research objectives and challenges related to creating large-scale datasets for multi-document summarization and applying neural network models on these datasets. We aim to understand their strengths and flaws and design data-efficient neural networks for text summarization when training data is not readily available. Furthermore, we examine the current state of neural network-based

summarization models and their evaluation to propose a more uniform research setup.

High-Resource Summarization Work in neural text summarization initially focused on single-document news summarization due to the availability of large-scale datasets such as the CNN-DailyMail dataset (CNNDM) (Nallapati et al., 2016). These datasets allowed neural abstractive models to surpass the performance of previous non-neural extractive models. The first large-scale dataset for multi-document summarization introduced was the WikiSum dataset (Liu et al., 2018), which aimed to produce Wikipedia-style pages based on content collected from the web. Previously, however, such multi-document data was missing for the news domain. We are interested in the application of techniques from single-document summarization to multi-document summarization. In particular, we analyze multi-document summarization within two domains, multi-document news summarization and summarization of scientific topics. We are interested in techniques for modeling two key aspects in multi-document summarization, namely information redundancy and the length of input texts. We are interested in reducing redundancy in model output and determining the extent to which the models trained on these datasets generalize to real-life settings.

Low-Resource Summarization Creating new datasets for each new domain that arises is infeasible, so making use of existing data and creating models that generalize without needing large-scale training data is very desirable. The models which are trained on one domain may not perform well on other domains. Recently, pre-trained language models were introduced to improve the transferability of models, and we study the application of these methods in unsupervised, distantly supervised, and few-shot learning. We aim to study how the characteristics of the testing domain, or the desired summary, can be used as prior knowledge to create data and improve performance in training summarization models with limited amounts of training data.

Automatic Text Summarization Evaluation Automatic text summarization evaluation poses many challenges, as there exist many valid summaries of the same input. Enumerating all possible valid summaries for automatic comparison is infeasible, and human evaluation of model outputs is expensive and time-consuming. The standard evaluation metric is ROUGE (Lin, 2004a), a lexical overlap-based metric that requires a gold reference summary. A fairly standard protocol combines analysis from automatic metrics such as ROUGE with human judgments of the summaries' qualities. However, these protocols vary greatly from paper to paper (Hardy et al., 2019). Furthermore, despite many proposed variations of ROUGE, which claim to improve correlations with human judgments, ROUGE remains the default automatic metric despite having shown poor human judgment correlations in several settings (Kryscinski et al., 2020a). Additionally, recent work in summarization has shifted from RNN-based models to Transformer-based models. While these models demonstrate improvements in automatic metrics, less work has quantified these improvements in terms of large-scale human judgments. We aim to understand better what improvements have been made in recent summarization models by studying a wide range of extractive and abstractive model outputs on the CNNDM dataset. We also aim to analyze metrics more comprehensively for summarization evaluation to understand better which metrics correlate most strongly with human evaluation and which should be reported in future model comparisons. We also are particularly interested in the evaluation of hallucinations in summarization to understand better where summarization models are unfaithful to the input documents.

1.3 Contributions

In this dissertation, we aim to analyze text summarization evaluation metrics and propose datasets and methodologies in deep neural networks for improving text summarization. We summarize our contributions along the following axes:

High-resource Summarization We introduce novel datasets applicable for training and evaluating multi-document summarization models in Chapter 3, Chapter 4, and Chapter 5. In Chapter 3, we introduce TutorialBank, a publicly-available dataset that aims to facilitate NLP education and research. We have manually collected and categorized over 6,300 resources on NLP as well as the related fields of Artificial Intelligence (AI), Machine Learning (ML), and Information Retrieval (IR). In Chapter 4 we introduce the first large-scale multi-document summarization dataset in the news domain. We propose an end-to-end method to incorporate a classical extractive method for diverse summarization into pointer-generator networks (See et al., 2017) to reduce redundancy in model output. In both automatic and human evaluations, our model improves in terms of content selection and redundancy over a comparable baseline model, with a statistically significant difference seen in human evaluations. This chapter is the most representative of the work in this part of the dissertation, as it mirrors an important trend in summarization work of the introduction of high-quality, large-scale datasets as well as modeling components to take advantage of this large-scale data. Chapter 5, builds on the style of data introduced in Chapter 3 for topic summarization, treating it as a multi-document summarization task as in Chapter 4. We obtain state-of-the-art performance on the WikiSum (Liu et al., 2018) dataset through the novel application of state-of-the-art pretrained models in a simple two-step pipeline. We extend this Wikipedia topic introduction summarization task to generate longer scientific topic summaries and notice the failure of previous models to generalize within this setting. We point to areas of improvement for future work and provide a better understanding of current methods and their faults in a real-world application, which stressed the need for data-efficient models that generalize to new settings.

Low-Resource Summarization We design methods for training data-efficient models, achieving state-of-the-art unsupervised performance for both extractive question-answering and abstractive text summarization across a suite of tasks, and state-of-the-art performance in few-shot and distantly-supervised settings for abstractive summarization. In Chapter

6, we introduce a retrieval, template-based framework for extractive question answering, a form of query-based summarization where the summary is a short phrase. We achieve, at the time, state-of-the-art results on SQuAD (Rajpurkar et al., 2016) for unsupervised models, improving over previous unsupervised models about 14%, and 20% when the answer is a named entity. In Chapter 7, we introduce a dataset generation pipeline for multi-answer summarization when no data is available. We show that current work using entailment as a measure of summary consistency with a source document does not use entailment models at the optimal granularity. Furthermore, we introduce a novel RL reward function for answer summarization in the form of the volume of semantic space covered to increase coverage of the source answers. Furthermore, we introduce a sentence-relevance prediction loss that increases the faithfulness of the resulting summaries and allows for more interpretable answer summaries. In Chapter 8 we introduce a method, called WikiTransfer, to create pseudo-summaries with subaspects of the target dataset, which can be used as unlabeled data for intermediate fine-tuning, and show that this method improves zero-shot domain transfer over transfer from other domains. We achieve state-of-the-art unsupervised abstractive summarization performance on the CNNDM dataset and three additional, diverse datasets. We demonstrate additional improvements in transferring our WikiTransfer models in the few shot setting, achieving state-of-the-art 10-shot performance across four datasets and state-of-the-art 100-shot performance across three of the four studied datasets. Furthermore, in human evaluations, the zero-shot performance of our model on CNNDM is not distinguishable from the fully-supervised performance in a statistically significant way for both the relevance and faithfulness dimensions. This trend is also found in zero-shot consistency performance and 100-shot relevance judgments for the XSum dataset. This chapter is emblematic of recent demand in NLP for more efficient models which perform well in zero or few-shot settings.

Automatic Text Summarization Evaluation In Chapter 9, we take stock of the current status of summarization evaluation. We re-evaluate 12 automatic evaluation metrics in a comprehensive and consistent fashion using neural summarization model outputs compared to expert and crowd-sourced human annotations along four quality dimensions. We also consistently benchmark 23 recent summarization models using the aforementioned automatic evaluation metrics to understand whether current automatic evaluation comparisons of recent models also align with human evaluation. Furthermore, we assemble the largest collection of summaries generated by models trained on the CNNDM news dataset and share it in a unified format along with a toolkit consisting of an extensible and unified API for evaluating summarization models across a broad range of automatic metrics. We believe that this work will help promote a complete evaluation protocol for text summarization and advance research in developing evaluation metrics that better correlate with human judgments. Furthermore, we explore summarization evaluation throughout our analysis of large-scale multi-document summarization and few-shot summarization, such as evaluating summaries in real-world settings in Chapter 5. Additionally, in Chapter 7, we show the effectiveness of entailment as a metric for faithfulness in answer summarization on community question answering forums.

1.4 Outline

The chapters in this thesis are based on the following publications and submissions:

- **Chapter 3: TutorialBank: A Manually-Collected Corpus for Prerequisite Chains, Survey Extraction and Resource Recommendation**. In The 56th Annual Meeting of the Association for Computational Linguistics (ACL), 2018.
- **Chapter 4: Multi-News: a Large-Scale Multi-Document Summarization Dataset and Abstractive Hierarchical Model**. In The 57th Annual Meeting of the Association for Computational Linguistics (ACL), 2019.

- **Chapter 5: Generating Full Wikipedia-Style Pages of Scientific Topics: an Application**. To be submitted to The 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2021.

- **Chapter 6: Template-Based Question Generation from Retrieved Sentences for Improved Unsupervised Question Answering**. In The 58th Annual Meeting of the Association for Computational Linguistics (ACL), 2020. First author, equal contribution with Patrick Ng.

- **Chapter 7: Multi-perspective Abstractive Answer Summarization**. Under submission at The 59th Annual Meeting of the Association for Computational Linguistics (ACL), 2021.

- **Chapter 8: Improving Zero and Few-Shot Abstractive Summarization with Intermediate Fine-tuning and Data Augmentation**. In The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT), 2021.

- **Chapter 9: SummEval: Re-evaluating Summarization Evaluation**. In The Transactions of the Association for Computational Linguistics (TACL), 2021. First author, equal contribution with Wojciech Kryściński.

Chapter 2

Background and Preliminaries

In this chapter, we provide background information and preliminaries for automatic text summarization. This chapter does not present a comprehensive overview of the history of text summarization. Instead, we review a brief overview of key components of summarization work and their relation to this thesis's work and broader trends in the field.

2.1 Summarization: Pre Neural Networks

Early work in text summarization focused primarily on extractive summarization.

Extractive Summarization Work in extractive automatic text summarization goes back to the 1950s, with the work of Luhn (1958), which proposed a method for summarization based on word frequencies. Other early work set the framework for later, machine-learning-based approaches by proposing to extract summaries based on a combination of factors rather than a single representation of a document topic (Edmundson, 1969). Most work has focused on sentence-level extraction. As noted in Nenkova and McKeown (2012), extractive systems typically follow three steps: 1) An intermediate representation of the input is formed, 2) Sentence scoring by which each sentence is scored according to importance, and 3) Sentence selection, by which sentences are chosen according to their importance score and other

desired properties such as redundancy reduction.

Within the formation of intermediate representations, there are topic representations, which convert the text to an intermediate representation interpreted as the topic discussed in the text. In contrast, indicator representations represent each sentence as a list of indicators of importance. Within the topic representation paradigm, approaches range from using word probability to calculate importance (Vanderwende et al., 2007), centroid summarization based on term-frequency inverse-document frequency (Radev et al., 2000), lexical chain methods (Chen et al., 2005), latent semantic analysis (Deerwester et al., 1990) in Gong and Liu (2001) as well as Bayesian models (Daumé III and Marcu, 2006). A common approach for indicator-representation-based summarization is graph-based modeling. Two foundational works in this area are inspired by the Page-Rank algorithm, namely Lexrank (Erkan and Radev, 2004), and Textrank (Mihalcea and Tarau, 2004). These graph-based approaches are typically used as benchmarks unsupervised extractive when experimenting with novel datasets, as in Chapter 4 and Chapter 7. Machine-learning-based approaches build on Hidden Markov Models (Conroy and O'leary, 2001) and Conditional Random Fields (Shen et al., 2007), among others (Hong and Nenkova, 2014).

Within sentence selection, sentences are chosen either in a greedy or globally optimal fashion. Maximal Marginal Relevance (MMR) is an approach for combining query-relevance with information-novelty in summarization for greedy sentence selection (Carbonell and Goldstein, 1998). We incorporate this sentence-selection algorithm within a neural network framework in Chapter 4. For global summary selection, constraints may be introduced within an integer linear programming solution (McDonald, 2007). Submodular functions of informativeness may be chosen for quickly finding an optimal solution (Lin and Bilmes, 2010).

Summarization Evaluation While summarization evaluation will be the focus of Chapter 9, its importance pervades all periods of text summarization work, so we introduce the

primary evaluation metric and evaluation settings to understand better our comparisons in the chapters which follow. The ROUGE (Lin, 2004a) metric measures lexical overlap between a summary generated by a model and an ideal (also called gold, or reference), typically human-written, summary, or summaries. The recall-based version of ROUGE, which measures n-gram overlap, is shown in Equation 2.1. The F1 measure is typically reported, and other units of overlap besides n-gram units, such as the longest common subsequence.

$$\frac{\sum_{S \in \text{ReferenceSummaries}} \sum_{\text{gram}_n \in S} \text{Count}_{\text{match}}(\text{gram}_n)}{\sum_{S \in \text{ReferenceSummaries}} \sum_{\text{gram}_n \in S} \text{Count}(\text{gram}_n)} \tag{2.1}$$

Lin (2004b) examined the effectiveness of the ROUGE metric in various tasks, concluding that evaluating against multiple references results in higher correlation scores with human judgments; however, a single-reference setting is sufficient for the metric to be effective. However, problems have been found with ROUGE. On a basic level, ROUGE fails to capture paraphrases not present in the reference summaries. Furthermore, correlation with human judgments has been shown to decrease when ROUGE is used outside of its original setting (Liu and Liu, 2008; Cohan and Goharian, 2016). Typically, human evaluation is performed alongside automatic evaluation to measure desired characteristics. Dang (2005) define several characteristics of readability such as grammaticality and coherence. While several variations on ROUGE (Zhou et al., 2006; Ng and Abrecht, 2015; Ganesan, 2015; ShafieiBavani et al., 2018) have since been introduced, as well as other text generation metrics, they have not seen widespread use, which we will discuss further in Chapter 9.

2.2 Summarization: Neural Networks

We introduce several large-scale datasets and modeling techniques that allowed for the proliferation of neural network-based summarization models.

Large-scale datasets Neural network methods came to the fore-front, in part, due to the availability of large-scale datasets, a trend which followed suit in summarization work. These works take advantage of large datasets such as the Gigaword Corpus (Napoles et al., 2012), the CNNDM dataset (Hermann et al., 2015), and the Newsroom corpus (Grusky et al., 2018), which contain on the order of hundreds of thousands to millions of article-summary pairs. A benchmark dataset for training summarization models is the CNNDM corpus (Hermann et al., 2015), originally a question answering task, which was repurposed for summarization by Nallapati et al. (2016). The dataset consists of news articles and associated human-created bullet-point summaries. The majority of these datasets encompass single-document summarization of news articles. However, our work on high-resource summarization focuses on expanding to multi-document and non-news summarization.

Neural Network Models Initial summarization models, and more broadly NLP models, consisted of sparse features such as n-gram models to represent text and the interactions among textual units. For example, a graph can be constructed from input sentences by comparing the cosine similarity of term-frequency inverse document frequency vectors. Recurrent neural network models gained renewed popularity in NLP with their performance on language modeling tasks (Mikolov et al., 2010). Furthermore, dense textual representations began to dominate work in NLP with the success of Word2Vec (Mikolov et al., 2013a), which produces dense word representations in which words that appear in similar contexts have similar representations. A common testbed for neural network models in NLP was the task of machine translation. Several recurrent neural network variations were proposed for this task (Cho et al., 2014; Sutskever et al., 2014), including variations such as Long Short-term Memory Networks (LSTMs) (Hochreiter and Schmidhuber, 1997), which aimed to improve the handling of longer-length sequences of text. Neural sequence-to-sequence methods were often first tested on machine translation before being applied to additional tasks such as machine translation. Neural methods showed great promise in single-document

setting, with both extractive (Nallapati et al., 2016; Cheng and Lapata, 2016; Narayan et al., 2018b) and abstractive methods (Chopra et al., 2016; Nallapati et al., 2016; See et al., 2017; Paulus et al., 2018; Cohan et al., 2018; Celikyilmaz et al., 2018; Gehrmann et al., 2018). In the context of extractive summarization, neural network models provide novel approaches for representing and determining similarity among textual units. Below, we introduce the abstractive Pointer-generator model (See et al., 2017), which largely improved over previous attempts at neural abstractive summarization on the CNNDM dataset and was a foundational work for future abstractive summarization approaches. Furthermore, we expand upon the Pointer-generator model for multi-document summarization in Chapter 4.

Let $x = \{x_1, x_2, ..., x_t, ..., x_n\}$ be a source document with n words and N sentences, where x_t represents the t-th word in x. It could also be represented as $\{s_1, s_2, ..., s_t, ..., s_N\}$, where s_t represents the t-th sentence in x. The corresponding target summary y contains m words and M sentences, and y_t denotes the t-th token of y. We will follow this notation throughout this dissertation.

Pointer-generator Network The pointer-generator network (See et al., 2017) is a commonly-used encoder-decoder summarization model with attention (Bahdanau et al., 2015) which combines copying words from source documents and outputting words from a vocabulary. The encoder converts each token x_i in the document into the hidden state h_i. At each decoding step t, the decoder has a hidden state d_t. An attention distribution a^t is calculated as in Bahdanau et al. (2015) and is used to get the context vector h_t^*, which is a weighted sum of the encoder hidden states, representing the semantic meaning of the related document content for this decoding time step:

$$\begin{aligned} e_i^t &= v^T \tanh(W_h h_i + W_s d_t + b_{attn}) \\ a^t &= \text{softmax}(e^t) \\ h_t^* &= \sum_i a_i^t h_i^t \end{aligned} \tag{2.2}$$

The context vector h_t^* and the decoder hidden state d_t are then passed to two linear layers to produce the vocabulary distribution P_{vocab}. For each word, there is also a copy probability P_{copy}. It is the sum of the attention weights over all the word occurrences:

$$\begin{aligned} P_{vocab} &= \text{softmax}(V^{'}(V[d_t, h_t^*] + b) + b^{'}) \\ P_{copy} &= \sum_{i:w_i=w} a_i^t \end{aligned} \tag{2.3}$$

The pointer-generator network has a soft switch p_{gen}, which indicates whether to generate a word from vocabulary by sampling from P_{vocab}, or to copy a word from the source sequence by sampling from the copy probability P_{copy}.

$$p_{gen} = \sigma(W_{h^*}^T h_t^* + W_d^T d_t + W_y^T y_t + b_{ptr}) \tag{2.4}$$

where y_t is the decoder input at timestep t, where W_{h^*}, W_d, W_y, and b_{ptr} are learnable parameters.

$$P(x) = p_{gen} P_{vocab}(x) + (1 - p_{gen}) P_{copy}(x) \tag{2.5}$$

Transformer The Transformer model replaces recurrent layers with self-attention in an encoder-decoder framework and has achieved state-of-the-art results in machine translation (Vaswani et al., 2017) and language modeling (Baevski and Auli, 2019; Dai et al., 2019). The Transformer had also been successfully applied to single-document summarization (Gehrmann et al., 2018) prior to work in pretrained networks and forms the basis of current pretrained networks. For each word during encoding, the multi-head self-attention sub-layer allows the encoder to directly attend to all other words in a sentence in one step. Decoding contains the typical encoder-decoder attention mechanisms as well as self-attention to all previous generated output. The Transformer motivates the elimination of recurrence to allow more direct interaction among words in a sequence.

Training Settings Standard supervised training for sequence-to-sequence neural networks minimizes the negative log-likelihood loss using supervised teacher forcing (Williams and Zipser, 1989), which we label L_{sup}:

$$L_{sup}(x, y) = -\sum_{t=1}^{m} \log(f(y_t|y_{0:t-1}, x, \theta)) \tag{2.6}$$

where $f(\cdot|\cdot, \theta)$ represents the distribution among the vocabulary predicted by our model with parameter θ. θ will be ignored for the following equations for simplicity.

2.3 Summarization: Pretrained Neural Networks

Pretraining for NLP Word2Vec (Mikolov et al., 2013a) allowed for creating dense vector representations of words by training a model to predict relevant words based on context. However, Word2Vec is a shallow network; it consists only of a single layer, and the resulting embeddings are input to a larger, task-specific neural network. However, recently work found that training an entire network on a task such as language modeling, where the model aims to predict the next word given context words, and initializing task-specific network layers on top of the network, vastly improved performance. Training on a task before fine-tuning on a final task is called pretraining. The intuition is that different layers of the deep network capture different language phenomena, such as syntax and semantics. Furthermore, pretraining on the language modeling task teaches the model some notions of language, as predicting the next word requires some level of language understanding. When fine-tuning the model for a down-stream task, the model does not need to learn these properties from scratch. Ramachandran et al. (2017) first applied pretraining networks for NLP. However, pretraining did not gain steam until the introduction of ULMFit (Howard and Ruder, 2018), ELMo (Peters et al., 2018), and GPT (Radford et al., 2018) models, which pretrained using the task of language modeling. Currently, the most widely used pretrained model is BERT (Bi-directional Encoder Representations from Transformers)

(Devlin et al., 2019). BERT consists of a bi-directional encoder Transformer model with the base version containing 110 million parameters and the large version containing 340 million parameters. BERT is notably bi-direction compared to previous work; during pretraining, the model predicts words given context from before and after the word after masking a given percentage of the input. BERT is trained on English Wikipedia and Book Corpus data (Zhu et al., 2015). Fine-tuning BERT with task-specific neural network layers achieved state-of-the-art performance on a wide range of natural language understanding tasks.

Pretraining for Summarization Liu and Lapata (2019a) first applied BERT to summarization, introducing a novel document-level encoder on top of BERT for both extractive and abstractive summarization. While BERT is pretrained as an encoder-only model focusing on natural language understanding as opposed to generation, subsequent work explored pretraining for sequence-to-sequence models. Raffel et al. (2019) frame a suite of understanding and generation tasks as text-input to text-output generation tasks, including summarization. Other work has also aimed to unify understanding and generation pretraining, such as UniLM (Dong et al., 2019) and decoder-only pretraining for applied to summarization (Ziegler et al., 2019). Zhang et al. (2019) introduce a model pretrained with a novel pretraining objective function designed for summarization by which important sentences are removed from an input document and then generated from the remaining sentences.

BART (Bidirectional and Auto-Regressive Transformers) (Lewis et al., 2020), a denoising autoencoder for pretraining sequence to sequence tasks applicable to natural language understanding and generation tasks. During pretraining, BART corrupts text with several noising functions, such as token deletion and sentence permutation, and learns an encoder-decoder model to reconstruct the original text. It can be seen as a generalization of the encoder-only BERT and autoregressive GPT models. The pretrained BART model can then be fine-tuned on summarization by training with standard negative log-likelihood loss as

described above and achieved state-of-the-art performance on summarization tasks such as CNNDM at the time of its release.

We make use of BERT-based models for extractive summarization in Chapter 6 and the BART model for Chapters 5, 7, and 8. We provide a comprehensive comparison of neural network-based models, both pretrained and without pretraining, in Chapter 9.

Part I

High-resource Text Summarization

Chapter 3

TutorialBank: A Manually-Collected Corpus for Prerequisite Chains, Topic Summarization, and Resource Recommendation

This chapter introduces the TutorialBank Corpus, a collection of over 6,300 hand-collected tutorials and other resources on NLP and related fields. This dataset and analysis do not directly apply summarization methods. However, we tackle similar problems central to summarization, such as learning from large corpora to address information overload. The need for data to tackle this and similar problems motivates a similar collection for multi-document summarization in Chapter 4. It has been found useful in subsequent work (Li et al., 2019). Furthermore, we directly tackle the topic summarization task introduced in this chapter as a testing ground for neural network summarization methods in Chapter 5.

3.1 Introduction

NLP has seen rapid growth over recent years. A Google search of "Natural Language Processing" returns over 100 million hits with papers, tutorials, blog posts, codebases, and other related online resources. Additionally, advances in related fields such as Artificial Intelligence and Deep Learning are strongly influencing current NLP research. With these developments, an increasing number of tutorials and online references are being published daily. As a result, the task of students, educators, and researchers of tracking the changing landscape in this field has become increasingly difficult as they must continuously sift through multiple sources to find valuable, relevant information.

Recent work has studied the educational aspect of mining text for presenting scientific topics. One goal has been to develop concept maps of topics, graphs showing which topics are prerequisites for learning a given topic (Gordon et al., 2016; Liu et al., 2016; Pan et al., 2017a,b; Liang et al., 2017). Another goal has been to automatically create reading lists for a subject either by building upon concept graphs (Gordon et al., 2017) or through an unstructured approach (Jardine, 2014). Additionally, other work has aimed to summarize scientific topics automatically, either by extractive summarization of academic papers (Jha et al., 2013, 2015; Jaidka et al., 2016) or by producing Wikipedia articles on these topics from multiple sources (Sauper and Barzilay, 2009; Liu et al., 2018). Scientific articles constitute primary texts which describe an author's work on a particular subject. In contrast, Wikipedia articles can be viewed as tertiary sources that summarize both results from primary works and explanations from secondary sources. Tang and McCalla (2004, 2009) and Sheng et al. (2017) explore the pedagogical function among the types of sources.

To address the problem of the scientific education of NLP more directly, we focus on the annotation and utilization of secondary sources presented in a manner immediately useful to the NLP community. We introduce the TutorialBank corpus, a manually-collected dataset of links to over 6,300 high-quality resources on NLP, Artificial Intelligence (AI), Machine Learning (ML), and Information Retrieval (IR). The corpus's magnitude, manual

collection, and focus on annotation for education and research differentiates it from other corpora. Throughout this chapter, we use the general term "resource" to describe any tutorial, research survey, blog post, codebase, or other online sources, focusing on educating on a particular subject. We have created a search engine [1] for these resources and have annotated them according to a taxonomy to facilitate their sharing. Additionally, we have annotated for pedagogical role, prerequisite relations, and relevance of resources to hand-selected topics and provide a command-line interface for our annotations. We released the dataset and present several avenues for further research.

Our main contribution is the manual collection of good quality resources related to NLP and the annotation and presentation of these resources conducive to NLP education. Our dataset is notably the largest manually-picked corpus of resources intended for NLP education which does not include only academic papers. Additionally, we show initial work on topic modeling and resource recommendation. We present a variant of standard reading-list generation, which recommends resources based on a title and abstract pair, and demonstrate additional uses and research directions for the corpus, such as scientific topic summarization.

3.2 Related Work

Pedagogical Value of Resources Online resources are found in formats that vary in their roles in education. Sheng et al. (2017) identify seven types of pedagogical roles found in technical works: Tutorial, Survey, Software Manual, Resource, Reference Work, Empirical Results, and Other. They annotate a dataset of over 1,000 resources according to these types. Beyond these types, resources differ in their pedagogical value, which they define as "the estimate of how useful a document is to an individual who seeks to learn about specific concepts described in the document". Tang and McCalla (2004, 2009) discuss the

1. `http://aan.how`

pedagogical value of a single type, academic papers, in relation to a larger recommendation system.

Prerequisite Chains Prerequisite chains refer to edges in a graph describing which topics are dependent on the knowledge of another topic. Prerequisite chains play an important role in curriculum planning and reading list generation. Liu et al. (2016) propose "Concept Graph Learning" in order to induce a graph from which they can predict prerequisite relations among university courses. Their framework consists of two graphs: (1) a higher-level graph which consists of university courses, and (2) a lower-level graph which consists of induced concepts and pair-wise sequential preferences in learning or teaching the concept.

Liang et al. (2017) experiment with prerequisite chains on education data but focus on the recovery of a concept graph rather than on predicting unseen course relations as in (Liu et al., 2016). They introduce both a synthetic dataset and one scraped from 11 universities, which includes course prerequisites and concept-prerequisite labels. Concept graphs are also used in Gordon et al. (2016) to address the problem of developing reading lists for students. The concept graph, in this case, is a labeled graph where nodes represent both documents and concepts (determined using Latent Dirichlet Allocation (LDA) (Blei et al., 2001)), and edges represent dependencies. They propose methods based on cross-entropy and information flow for determining edges in the graph. Finally, finding prerequisite relationships has also been used in other contexts such as Massive Open Online Courses (MOOCs) (Pan et al., 2017a,b).

Reading List Generation Jardine (2014) generates recommended reading lists from a corpus of technical papers in an unstructured manner in which a topic model weighs the relevant topics, and relevant papers are chosen through the proposed ThemedPageRank approach. Conversely, Gordon et al. (2017) approach reading list generation from a structured perspective, first generating a concept graph from the corpus and then traversing the graph to select the most relevant document.

Topic Summarization Recent work on topic summarization, or creating a broad summary of a topic, for scientific topics has focused on creating summaries from academic papers (Jha et al., 2013, 2015; Jaidka et al., 2016). Jha et al. (2013) present a system that generates summaries given a topic keyword. From a base corpus of papers found by query matching, they expand the corpus via a citation network using a heuristic called Restricted Expansion. This process is repeated for seven standard NLP topics. Similarly, Jha et al. (2015) experiment with fifteen topics in computational linguistics and collect at least three surveys written by experts on each topic, also using citation networks to expand their corpus. They introduce a content model and a discourse model and perform qualitative comparisons of coherence with a standard summarization model.

The task of topic summarization has also been viewed in the multi-document summarization setting of generating Wikipedia articles (Sauper and Barzilay, 2009; Liu et al., 2018). Sauper and Barzilay (2009) induce domain-specific templates from Wikipedia and fill these templates with content from the Internet. More recently Liu et al. (2018) explore a diverse set of domains for summarization and are the first to attempt abstractive summarization of the first section of Wikipedia articles by combining extractive and abstractive summarization methods.

3.3 Resource Collection

An Overview of TutorialBank As opposed to other collections like the ACL Anthology (Bird et al., 2008; Radev et al., 2009, 2013, 2016), which contain solely academic papers, our corpus focuses mainly on resources other than academic papers. The main goal in our decision process of what to include in our corpus has been the quality-control of resources that can be used for an educational purpose. Initially, the resources collected were conference tutorials as well as surveys, books, and longer papers on broader topics, as these genres contain an inherent amount of quality-control. Later on, other online resources were added

1 - Introduction and Linguistics
2 - Language Modeling, Syntax and Parsing
3 - Semantics and Logic
4 - Pragmatics, Discourse, Dialogue and Applications
5 - Classification and Clustering
6 - Information Retrieval and Topic Modeling
7 - Neural Networks and Deep Learning
8 - Artificial Intelligence
9 - Other Topics

Table 3.1: TutorialBank Top-level Taxonomy Topics

Topic Category	Count
Introduction to Neural Networks and Deep Learning	635
Tools for Deep Learning	475
Miscellaneous Deep Learning	287
Machine Learning	225
Word Embeddings	139
Recurrent Neural Networks	134
Python Basics	133
Reinforcement learning	132
Convolutional Neural Networks	129
Introduction to AI	89

Table 3.2: TutorialBank corpus count by taxonomy topic for the most frequent topics (excluding topic "Other").

to the corpus, as explained below. Student annotators, described later on, as well as the professor, examined resources which they encountered in their studies. The resources were added to the corpus if deemed of good quality. It is important to note that not all resources found on the Internet were added to TutorialBank; one could scrape the web according to search terms, but the quality control of the results would be largely missing. The quality of a resource is a somewhat subjective measure, but we aimed to find resources that would serve a pedagogical function to either students or researchers, with a professor of NLP making the final decision. This collection of resources and meta-data annotation has been done over multiple years, while we created the search engine and added additional annotations mentioned below in preparation for this chapter.

TutorialBank Taxonomy To facilitate the sharing of resources about NLP, we developed a taxonomy of 305 topics of varying granularity. The top levels of our taxonomy tree are shown in Table 3.1. Our Taxonomy's backbone corresponds to the syllabus of a university-level NLP course and was expanded to include related topics from other courses in ML, IR, and AI. As a result, there is a bias in the corpus towards NLP resources and resources from other fields in so far as they are relevant to NLP. However, this bias is planned, as our focus remains on teaching NLP. The resource count for the most frequent taxonomy topics is shown in Table 3.2.

Data Preprocessing We downloaded the corresponding PDF, PowerPoint presentations, and other source formats for each resource in the corpus and used PDFBox to perform OCR in translating the files to textual format. For HTML pages, we downloaded both the raw HTML with all images as well as a formatted text version of the pages. We release only the metadata such as URLs and annotations and provide scripts for reproducing the dataset for copyright purposes.

3.4 TutorialBank Annotation

Annotations were performed by a group of three Ph.D. students in NLP and six undergraduate Computer Science students who have taken at least one course in AI or NLP.

Pedagogical Function When collecting resources from the Internet, each item was labeled according to the medium in which it was found, analogous to the pedagogical function of Sheng et al. (2017). We will use this term throughout this chapter to describe this categorization. The categories, along with their counts, are shown in Table 3.3:

• **Corpus:** A corpus provides access to and a description of a scientific dataset.

• **Lecture:** A lecture consists of slides/notes from a university lecture.

Resource Category	Count
corpus	131
lecture	126
library	1014
link set	1186
naclo	154
paper	1176
survey	390
tutorial	2079

Table 3.3: TutorialBank corpus count by pedagogical feature.

- **Library:** A library consists of GitHub pages and other codebases that aid in the implementation of algorithms.
- **NACLO:** NACLO problems refer to linguistics puzzles from the North American Computational Linguistics Olympiad.
- **Paper:** A paper is a short/long conference paper taken from sites such as https://arxiv.org/ and which is not included in the ACL Anthology.
- **Link set:** A link set provides a collection of helpful links in one location.
- **Survey:** A survey is a long paper or book which describes a broader subject.
- **Tutorial:** A tutorial is a slide deck from a conference tutorial or an HTML page that describes a contained topic.

Topic to Resource Collection We first identified by hand 200 potential topics for topic summarization in NLP, ML, AI, and IR. Topics were added according to the following criteria:

1. It is conceivable that someone would write a Wikipedia page on this topic (an actual page may or may not exist).
2. The topic is not overly general (e.g., "Natural Language Processing") or too obscure or narrow.

Capsule Networks
Domain Adaptation
Document Representation
Matrix factorization
Natural language generation
Q Learning
Recursive Neural Networks
Shift-Reduce Parsing
Speech Recognition
Word2Vec

Table 3.4: Random sample of the list of 200 topics used for prerequisite chains, readling lists and topic summarization.

3. To write a summary of the topic, one would need to include information from a number of sources.

While some of the topics come from our taxonomy, many of the taxonomy topics have a different granularity than we desired, which motivated our topic collection. Topics were added to the list along with their corresponding Wikipedia pages if they exist. A sample of the topics selected is shown in 3.4. Once the list of topics was compiled, annotators were assigned topics and asked to search that topic in the TutorialBank search engine and find relevant resources. In order to impose some uniformity on the dataset, we chose to only include resources, which consisted of PowerPoint slides as well as HTML pages labeled as tutorials. We divided the topics among the annotators and asked them to choose five resources per topic using our search engine. The resource need not solely focus on the given topic; the resource may be on a more general topic and include a section on the given topic. As in general searching for resources, often resources include related information, so we believe this setting is fitting. For some topics, the annotators chose fewer than five resources (partially due to the constraint we impose on the form of the resources). We noted topics for which no resources were found, and rather than replace the topics to reflect TutorialBank coverage; we leave these topics in and plan to add additional resources in a future release.

Prerequisite Chains Even with a collection of resources and a list of topics, a student may not know where to begin studying a topic of interest. For example, to understand sentiment analysis, the student should be familiar with Bayes' Theorem, the basics of ML, and other topics. For this purpose, the annotators annotated which topics are prerequisites for the given topics from their reading lists. We expanded our list of potential prerequisites to include eight additional topics that were too broad for topic summarization (e.g., Linear Algebra) but are important prerequisites to capture. Following the method of Gordon et al. (2016), we define labeling a topic Y as a prerequisite of X according to the following question:

- Would understanding Topic Y help you to understand Topic X?

As in (Gordon et al., 2016), the annotators can answer this question as "no", "somewhat" or "yes."

Reading Lists When annotators were collecting relevant resources for a particular topic, we asked them to order the resources they found in terms of the usefulness of the resource for learning that particular topic. We also include the Wikipedia pages corresponding to the topics, when available, as an additional source of information. We do not perform additional annotation of the order of the resources or experiment in automatically reproducing these ordered lists. Rather, we offer this annotation as a pedagogical tool for students and educators. We plan the expansion of these lists and analysis in future experiments.

Topic Summarization We frame the task of topic summarization as a document retrieval task. A student searching for resources to learn about a topic such as Recurrent Neural Networks (RNN's) may encounter resources 1) which solely cover RNN's as well as 2) resources that cover RNN's within the context of a larger topic (e.g., Deep Learning). Within the first type, not every piece of content (a single PowerPoint slide or section in a blog post) contributes equally well to an understanding of RNN's; the content may focus on background information or may not clearly explain the topic. Within the second type,

larger tutorials may contain valuable information on the topic but may also contain much information not immediately relevant to the query. Given a query topic and a set of parsed documents, we want to retrieve the parts most relevant to the topic.

In order to prepare the dataset for extractive topic summarization, we first divide resources into units of content, which we call "cards". PowerPoint slides inherently contain a division in the form of each individual slide, so we divide PowerPoint presentations into individual slides/cards. For HTML pages, the division is less clear. However, we convert the HTML pages to a markdown file and then automatically split the markdown file using header markers. We believe this is a reasonable heuristic as tutorials and similar content tend to be broken up into sections signaled by headers.

For each of the resources that the annotators gathered for the reading lists on a given topic, that same annotator was presented with each card from that resource and asked to rate its usefulness. The annotator could rate the card from 0-2, with 0 meaning the card is not useful for learning the specified topic, 1 meaning the card is somewhat useful, and 2 meaning the card is useful. We chose a 3-point scale as initial trials showed a 5-point scale to be too subjective. The annotators also had the option in our annotation interface to drop cards, which were parsed incorrectly or were repeated one after the other and skip cards and return to score a card.

Illustrations Whether needed to understand a subject more deeply or prepare a blog post on a subject, images play an important role in presenting concepts more concretely. Simply extracting the text from HTML pages leaves behind this valuable information, and OCR software often fails to parse complex graphs and images in a non-destructive fashion. To alleviate this problem and promote the sharing of images, we extracted all images from our collected HTML pages. Since many images were simply HTML icons and other extraneous images, we manually checked the images and selected those which are of value to the NLP student. We collected a total of 2,000 images and matched them with the taxonomy topic

name of the resource it came from as well as the URL of the resource. While we cannot outdo the countless images from Google search, we believe illustrations can be an additional feature of our search engine, and we describe an interface for this collection below.

3.5 Features and Analysis

Search Engine In order to present our corpus in a user-friendly manner, we created a search engine using Apache Lucene[2]. We allow the user to query keywords to search our resource corpus, and the results can then be sorted based on relevance, year, topic, medium, and other metadata. In addition to searching by term, users can browse the resources by topic according to our taxonomy. For each child topic from the top-level taxonomy downward, we display resources according to their pedagogical functions. In addition to searching for general resources, we also provide search functionality for a corpus of papers, where the user can search by keyword and author, and venue.

While the search engine described above provides access to our base corpus and metadata, we also provide a command-line interface tool with our release so that students and researchers can easily use our annotations for prerequisite topics, illustrations, and topic summarization for educational purposes. The tool allows the user to input a topic from the taxonomy and retrieve all images related to that topic according to our metadata. Additionally, the user can input a topic from our list of 200 topics, and our tool outputs the prerequisites of that topic according to our annotation as well as the cards labeled as relevant for that topic.

Resource Recommendation from Title and Abstract Pairs In addition to needing to search for a general term, often a researcher begins with an idea for a project which is already focused on a nuanced sub-task. An employee at an engineering company may be

2. http://lucene.apache.org/

starting a project on image captioning. Ideas about this project's potential direction may be clear, but what resources may be helpful or what papers have already been published on the subject may not be immediately obvious. To this end, we propose the task of recommending resources from title and abstract pairs. The employee will input the title and abstract of the project and obtain a list of resources that can help complete the project. This task is analogous to reproducing the reference section of a paper, however, with a focus on tutorials and other resources rather than solely on papers. As an addition to our search engine, we allow a user to input a title and an abstract of variable length. We then propose taxonomy topics based on string matches with the query and a list of resources and papers and their scores as determined by the search engine. We later explore two baseline models for recommending resources based on document and topic modeling.

Dataset and Annotation Statistics We created reading lists for 182 of the 200 topics we identified in Section 4.2. Resources were not found for 18 topics due to the granularity of the topic (e.g., Radial Basis Function Networks) as well as our intended restriction of the chosen resources to PowerPoint presentations and HTML pages. The average number of resources per reading list for the 182 topics is 3.94. As an extension to the reading lists, we collected Wikipedia pages for 184 of the topics and presented these URLs as part of the dataset.

We annotated prerequisite relations for the 200 topics described above. We present a subset of our annotations in Figure 3.1, which shows the network of topic relations (nodes without incoming edges were not annotated for their prerequisites as part of this shown inter-annotation round). Our network consists of 794 unidirectional edges and 33 bidirectional edges. The presence of bidirectional edges stems from our definition of a prerequisite, which does not preclude bidirectionality (one topic can help explain another and vice-versa) as well as the similarity of the topics. The set of bidirectional edges consists of topic pairs (BLEU - ROUGE; Word Embedding - Distributional Semantics; Backpropagation - Gradient descent),

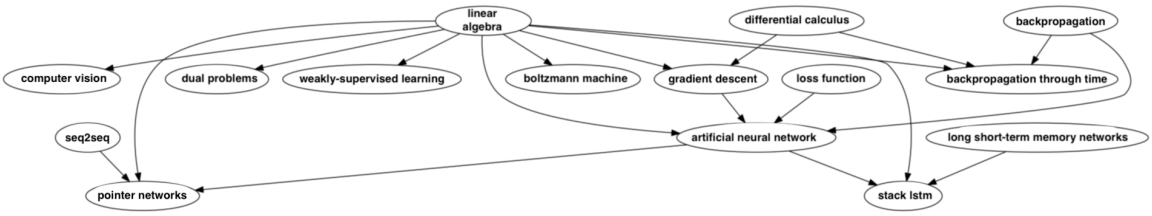

Figure 3.1: Visualization of a subset of prerequisite annotations.

which could be collapsed into one topic to create a directed acyclic graph in the future.

For topic summarization, we automatically split 313 resources into content cards, which we annotated for relevance in creating topic summaries. These resources are a subset of the reading lists limited in number due to constraints in downloading URLs and parsing to our annotation interface. The total number of cards that were not marked as repeats or misparsed totals 17,088, with 54.59 per resource. 6,099 cards were labeled as somewhat relevant or relevant for the target topic. The resources marked as non-relevant may be poorly presented or may not pertain fully to the topic. These numbers confirm the appropriateness of this corpus as a non-trivial information retrieval task.

To better understand the difficulty of our annotation tasks, we performed inter-annotator agreement experiments for each of our annotations. We randomly sampled twenty-five resources and had annotators label for pedagogical function. Additionally, we sampled twenty-five topics for prerequisite annotations and five topics with reading list lengths of five for topic relevance annotation. We used Fleiss's Kappa (Fleiss et al., 2004), a variant of Cohen's Kappa (Cohen, 1960) designed to measure annotator agreement for more than two annotators. The results are shown in Table 3.5. Using the scale as defined in Landis and Koch (1977), pedagogical function annotation exhibits *substantial agreement* while prerequisite annotation and topic summary annotation show *fair agreement*. The Kappa score for the pedagogical function is comparable to that of Sheng et al. (2017) (0.68) while the prerequisite annotation is slightly lower than the agreement metric used in Gordon et al. (2016) (0.36) although they measure agreement through Pearson correlation. We believe that the sparsity of the labels plays a role in these scores.

Annotation	Kappa
Pedagogical Function	0.69
Prerequisites	0.30
Topic Summarization Relevance	0.33

Table 3.5: Inter-annotator agreement for TutorialBank annotations.

Comparison to Similar Datasets Our corpus distinguishes itself in its magnitude, manual collection, and focus on annotation for educational purposes in addition to research tasks. We use similar categories for classifying pedagogical function as Sheng et al. (2017), but our corpus is hand-picked and over four-times larger while exhibiting similar annotation agreement.

Gordon et al. (2016) present a corpus for prerequisite relations among topics, but this corpus differs in coverage. They used LDA topic modeling to generate a list of 300 topics, while we manually create a list of 200 topics based on the criteria described above. Although their topics are generated from the ACL Anthology and related to NLP, we find less than a 40% overlap in topics. Additionally, they only annotate a subset of the topics for prerequisite annotations while we focus on broad coverage, annotating two orders of magnitude larger in terms of prerequisite edges while exhibiting fair inter-annotator agreement.

Previous work and datasets on summarizing scientific topics have focused on scientific articles (Jha et al., 2013, 2015; Jaidka et al., 2016) and Wikipedia pages (Sauper and Barzilay, 2009; Liu et al., 2018) as a summarization task. We, on the other hand, view this problem as an information retrieval task and focus on extracting content from manually-collected PowerPoint slides and online tutorials. Sauper and Barzilay (2009) differ in their domain coverage, and while the summaries of Jha et al. (2013, 2015) focus on NLP, we collect resources for an order of magnitude larger set of topics. Finally, our focus here in creating topic summaries, as well as the other annotations, is first and foremost to create a useful tool for students and researchers. Websites such as the ACL Anthology[3] and arXiv[4] provide

3. http://aclweb.org/anthology/

4. https://arxiv.org/

an abundance of resources but do not focus on the pedagogical aspect of their content. Meanwhile, websites such as Wikipedia, which aim to create a topic summary, may not reflect the latest trends in rapidly changing fields. As an example of our corpus usage, we experimented with topic modeling and its extension to resource recommendation. We restricted our corpus for this study to non-HTML files to examine the single domain of PDFs and PowerPoint presentations. This set consists of about 1,480 files with a vocabulary size of 191,446 and a token count of 9,134,452. For each file, the tokens were processed, stop tokens were stripped, and then each token was stemmed. Words with counts less than five across the entire corpus were dropped. We experimented with two models: LDA, a generative probabilistic model mentioned earlier, and Doc2Vec (Le and Mikolov, 2014), an extension of Word2Vec (Mikolov et al., 2013a) which creates representations of arbitrarily-sized documents. Figure 3.2 shows the document representations obtained with Doc2Vec as well as the topic clusters created with LDA. The grouping of related resources around a point demonstrates the clustering abilities of these models. We applied LDA in an unsupervised way, using 60 topics over 300 iterations as obtained through experimentation, and then colored each document dot with its category to observe the distribution. Our Doc2Vec model used hidden dimension 300, a window size of ten, and a constant learning rate of 0.025. Then, the model was trained for ten epochs.

3.6 Topic Modeling and Resource Recommendation

We tested these models for the task of resource recommendation from title+abstract pairs. We collected ten random papers from ACL 2017. For LDA, the document was classified into a topic, and then the top resources from that topic were chosen, while Doc2Vec computed the similarity between the query document and the training set and chose the most similar documents. We concatenated the title and abstract as input and had our models predict the top 20 documents. We then had five annotators rate the recommendations for helpfulness as

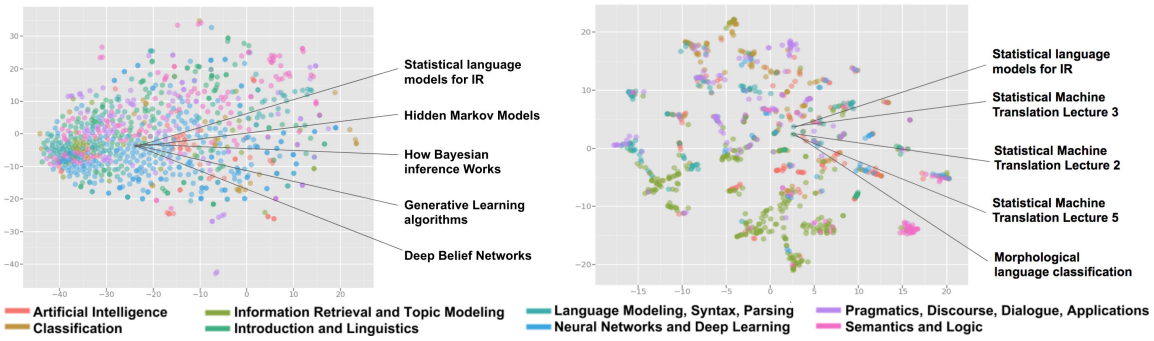

Figure 3.2: Plot showing a query document with title "Statistical language models for IR" and its neighbour document clusters as obtained through tSNE dimension reduction for Doc2Vec (left) and LDA topic modeling (right). Nearest neighbor documents titles are shown to the right of each plot.

0 (not helpful) or 1 (helpful). Recommended resources were rated according to the criterion of whether reading this resource would be useful in doing a project as described in the title and abstract. The results are found in Figure 3.3. Averaging the performance over each test case, the LDA model performed better than Doc2Vec (0.45 to 0.34), although both leave large room for improvements. LDA recommended resources notably better for cases 5 and 6, which correspond to papers with very well defined topics areas (Question Answering and Machine Translation), while Doc2Vec was able to find similar documents for cases 2 and 8, which are a mixture of topics, yet are well-represented in our corpus (Reinforcement Learning with dialog agents and emotion (sentiment) detection with classification). The low performance for both models also corresponds to differences in corpus coverage, and we plan to explore this bias in the future. We believe that this variant of reading list generation, as well as the relationship between titles and abstracts, is an unexplored and exciting area for future research.

3.7 Summary

In this chapter, we introduced the TutorialBank Corpus, a collection of over 6,300 hand-collected resources on NLP and related fields. Our corpus is notably larger than similar

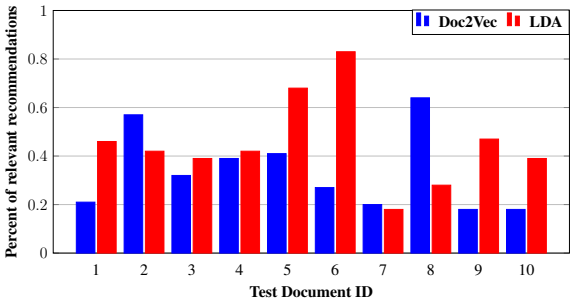

Figure 3.3: Relevance accuracy of Doc2Vec and LDA resource recommendation models.

datasets which deal with pedagogical resources and topic dependencies and unique in use as an educational tool. To this point, we believe that this dataset, with its multiple layers of annotation and usable interface, will be an invaluable tool to the students, educators, and researchers of NLP. Additionally, the corpus promotes research on tasks not limited to pedagogical function classification, topic modeling, and prerequisite relation labeling. Finally, we formulate the problem of recommending resources for a given title and abstract pair as a new way to approach reading list generation and propose two baseline models. For future work, we plan to continue the collection and annotation of resources and to separately explore each of the above research tasks.

Chapter 4

Multi-News: a Large-Scale Multi-Document Summarization Dataset and Abstractive Hierarchical Model

In the previous chapter, we introduced a large-scale dataset applicable for several tasks for information extraction and summarization for educational purposes. In this chapter, we focus on large-scale data applied to a particular task of multi-document news summarization and neural-network methods for reducing the redundancy of system outputs. We introduce Multi-News, the first large-scale MDS news dataset. Additionally, we propose an end-to-end model which incorporates a traditional extractive summarization model with a standard single-document summarization model and achieves competitive results on multi-document summarization datasets. We further benchmark several methods on Multi-News and release our data and code in hope that this work will promote advances in summarization in the multi-document setting.

Source 1
Meng Wanzhou, Huawei's chief financial officer and deputy chair, was arrested in Vancouver on 1 December. Details of the arrest have not been released...
Source 2
A Chinese foreign ministry spokesman said on Thursday that Beijing had separately called on the US and Canada to "clarify the reasons for the detention "immediately and "immediately release the detained person ". The spokesman...
Source 3
Canadian officials have arrested Meng Wanzhou, the chief financial officer and deputy chair of the board for the Chinese tech giant Huawei,...Meng was arrested in Vancouver on Saturday and is being sought for extradition by the United States. A bail hearing has been set for Friday...
Summary
...Canadian authorities say she was being sought for extradition to the US, where the company is being investigated for possible violation of sanctions against Iran. Canada's justice department said Meng was arrested in Vancouver on Dec. 1... China's embassy in Ottawa released a statement. . . .. "The Chinese side has lodged stern representations with the US and Canadian side, and urged them to immediately correct the wrongdoing "and restore Meng's freedom, the statement said...

Table 4.1: An example from our multi-document summarization dataset showing the input documents and their summary. The content found in the summary is color-coded.

4.1 Introduction

The automatic generation of summaries from multiple news articles is a valuable tool as the number of online publications grows rapidly. Single document summarization (SDS) systems have benefited from advances in neural encoder-decoder model thanks to the availability of large datasets. However, multi-document summarization, which aims to output summaries from document clusters on the same topic, had largely been performed on datasets with less than 100 document clusters such as the DUC 2004 (Paul and James, 2004) and TAC 2011 (Owczarzak and Dang, 2011) datasets and benefited less from advances in deep learning methods. Multi-document summarization (MDS) of news events offers the challenge of outputting a well-organized summary that covers an event comprehensively while simultaneously avoiding redundancy. The input documents may differ in focus and point of view for an event. We present an example of multiple input news documents and their summary in Figure 4.1. The three source documents discuss the same event and contain overlaps in content: the fact that *Meng Wanzhou was arrested* is stated explicitly in Source 1

and 3 and indirectly in Source 2. However, some sources contain information not mentioned in the others, which should be included in the summary: Source 3 states that *(Wanzhou) is being sought for extradition by the US* while only Source 2 mentioned *the attitude of the Chinese side*. Recent work in tackling this problem with neural models has attempted to exploit the graph structure among discourse relations in text clusters (Yasunaga et al., 2017) or through an auxiliary text classification task (Cao et al., 2017). Additionally, a couple of recent papers have attempted to adapt neural encoder-decoder models trained on single-document summarization datasets to MDS (Lebanoff et al., 2018; Baumel et al., 2018; Zhang et al., 2018b). However, data sparsity has largely been the bottleneck of the development of neural MDS systems. The creation of a large-scale multi-document summarization dataset for training has been restricted due to the sparsity and cost of human-written summaries. Liu et al. (2018) trains abstractive sequence-to-sequence models on a large corpus of Wikipedia text with citations and search engine results as input documents. However, no analogous dataset exists in the news domain. To bridge the gap, we introduce **Multi-News**, the first large-scale MDS news dataset, which contains 56,216 input-summary pairs. We also propose a hierarchical model for neural abstractive multi-document summarization, which consists of a pointer-generator network (See et al., 2017) and an additional Maximal Marginal Relevance (MMR) (Carbonell and Goldstein, 1998) module that calculates sentence ranking scores based on relevancy and redundancy. We integrate sentence-level MMR scores into the pointer-generator model to adapt the attention weights on a word-level. Our model performs competitively on the Multi-News datasets. We additionally perform a human evaluation on several system outputs. Our contributions are as follows: 1) We introduce the first large-scale multi-document summarization datasets in the news domain. 2) We propose an end-to-end method to incorporate MMR into pointer-generator networks. 3) Finally, we benchmark several methods on Multi-News and release our data and code to promote advances in summarization in the multi-document setting[5].

5. https://github.com/Alex-Fabbri/Multi-News

4.2 Related Work

Traditional non-neural approaches to multi-document summarization have been both extractive (Carbonell and Goldstein, 1998; Radev et al., 2000; Mihalcea and Tarau, 2004; Haghighi and Vanderwende, 2009) as well as abstractive (McKeown and Radev, 1995; Radev and McKeown, 1998; Barzilay et al., 1999; Ganesan et al., 2010).

Recent work has attempted unsupervised and weakly supervised methods in non-news domains (Chu and Liu, 2019b; Angelidis and Lapata, 2018). The methods most related to this work are SDS adapted for MDS data. Zhang et al. (2018c) adopts a hierarchical encoding framework trained on SDS data to MDS data by adding an additional document-level encoding. Baumel et al. (2018) incorporates query relevance into standard sequence-to-sequence models. Lebanoff et al. (2018) adapts encoder-decoder models trained on single-document datasets to the MDS case by introducing an external MMR module that does not require training on the MDS dataset. In our work, we incorporate the MMR module directly into our model, learning weights for the similarity functions simultaneously with the rest of the model.

4.3 Multi-News Dataset

Our dataset, which we call Multi-News, consists of news articles and human-written summaries of these articles from the site newser.com. Each summary is professionally written by editors and includes links to the original articles cited. We release stable Wayback-archived links and scripts to reproduce the dataset from these links. Our dataset is notably the first large-scale dataset for MDS on news articles. Our dataset also comes from a diverse set of news sources; over 1,500 sites appear as source documents five times or greater, as opposed to previous news datasets (for MDS, DUC comes from 2 sources while for SDS, CNNDM comes from CNN and Daily Mail respectively, and even the notably large Newsroom dataset (Grusky et al., 2018) covers only 38 news sources). A total of 20 editors contribute to 85%

# of source	Frequency	# of source	Frequency
2	23,894	7	382
3	12,707	8	209
4	5,022	9	89
5	1,873	10	33
6	763		

Table 4.2: The number of source articles per example, by frequency, in our dataset.

of the total summaries on newser.com. Thus we believe that this dataset allows for the summarization of diverse source documents and summaries.

Statistics and Analysis The number of collected Wayback links for summaries and their corresponding cited articles totals over 250,000. We only include examples with between 2 and 10 source documents per summary, as our goal is MDS, and the number of examples with more than ten sources was minimal. The number of source articles per summary present after downloading and processing the text to obtain the original article text varies across the dataset, as shown in Table 4.2. We believe this setting reflects real-world situations; often, for a new or specialized event, there may be only a few news articles. Nonetheless, we would like to summarize these events in addition to others with greater news coverage. We split our dataset into training (80%, 44,972), validation (10%, 5,622), and test (10%, 5,622) sets. Table 4.3 compares Multi-News to other news datasets used in experiments below. We choose to compare Multi-News with DUC data from 2003 and 2004 and TAC 2011 data, which are typically used in multi-document settings. Additionally, we compare to the single-document CNNDM dataset, as this has been recently used in work that adapts SDS to MDS (Lebanoff et al., 2018). The number of examples in our Multi-News dataset is two orders of magnitude larger than previous MDS news data. The total number of words in the concatenated inputs is shorter than other MDS datasets, as those consist of 10 input documents, but larger than SDS datasets, as expected. Our summaries are notably longer than in other work, about 260 words on average. While compressing information into a shorter text is the goal of summarization, our dataset tests the ability of abstractive models

Dataset	# pairs	# words (docs)	# sents (docs)	# words (summary)	# sents (summary)	vocab size
Multi-News	44,972/5,622/5,622	2,103.49	82.73	263.66	9.97	666,515
DUC03+04	320	4,636.24	173.15	109.58	2.88	19,734
TAC 2011	176	4,695.70	188.43	99.70	1.00	24,672
CNNDM	287,227/13,368/11,490	810.57	39.78	56.20	3.68	717,951

Table 4.3: Comparison of our Multi-News dataset to other MDS datasets as well as an SDS dataset used as training data for MDS (CNNDM). Training, validation and testing size splits (article(s) to summary) are provided when applicable. Statistics for multi-document inputs are calculated on the concatenation of all input sources.

% novel n-grams	Multi-News	DUC03+04	TAC11	CNNDM
uni-grams	17.76	27.74	16.65	19.50
bi-grams	57.10	72.87	61.18	56.88
tri-grams	75.71	90.61	83.34	74.41
4-grams	82.30	96.18	92.04	82.83

Table 4.4: Percentage of n-grams in summaries which do not appear in the input documents , a measure of the abstractiveness, in relevant datasets.

to generate fluent text concise in meaning while also coherent in the entirety of its generally longer output, which we consider an interesting challenge.

Diversity We report the percentage of n-grams in the gold summaries which do not appear in the input documents as a measure of how abstractive our summaries are in Table 4.4. As the table shows, the smaller MDS datasets tend to be more abstractive, but Multi-News is comparable and similar to the abstractiveness of SDS datasets. Grusky et al. (2018), in the context of SDS, additionally define three measures of the extractive nature of a dataset, which we use here for a comparison. We extend these notions to the multi-document setting by concatenating the source documents and treating them as a single input. Extractive fragment coverage is the percentage of words in the summary that are from the source article, measuring the extent to which a summary is derivative of a text:

$$\textit{COVERAGE(A,S)} = \frac{1}{|S|} \sum_{f \in F(A,S)} |f| \tag{4.1}$$

where A is the article, S the summary, and $F(A, S)$ the set of all token sequences identified as extractive in a greedy manner; if there is a sequence of source tokens that is a prefix of the remainder of the summary, that is marked as extractive. Similarly, density is defined as the average length of the extractive fragment to which each summary word belongs:

$$\textit{DENSITY(A,S)} = \frac{1}{|S|} \sum_{f \in F(A,S)} |f|^2 \tag{4.2}$$

Finally, the compression ratio is defined as the word ratio between the articles and its summary:

$$\textit{COMPRESSION(A,S)} = \frac{|A|}{|S|} \tag{4.3}$$

These numbers are plotted using kernel density estimation in Figure 4.1. As explained above, our summaries are larger on average, which corresponds to a lower compression rate. The variability along the x-axis (fragment coverage) suggests variability in the percentage of copied words, with the DUC data varying the most. In terms of the y-axis (fragment density), our dataset shows variability in the average length of the copied sequence, suggesting varying styles of word sequence arrangement. Our dataset exhibits extractive characteristics similar to the CNNDM dataset.

Other Datasets Large scale datasets for multi-document news summarization are lacking. There have been several attempts to create MDS datasets in other domains. Zopf (2018) introduce a multi-lingual MDS dataset based on English and German Wikipedia articles as summaries with about 7,000 examples. Liu et al. (2018) use Wikipedia to create a dataset of over two million examples, using Wikipedia references as input documents but largely relying on Google search to increase topic coverage. We, however, are focused on the news domain, and the source articles in our dataset are specifically cited by the corresponding summaries. Related work has also focused on opinion summarization in the multi-document setting; Angelidis and Lapata (2018) introduces a dataset of 600 Amazon product reviews.

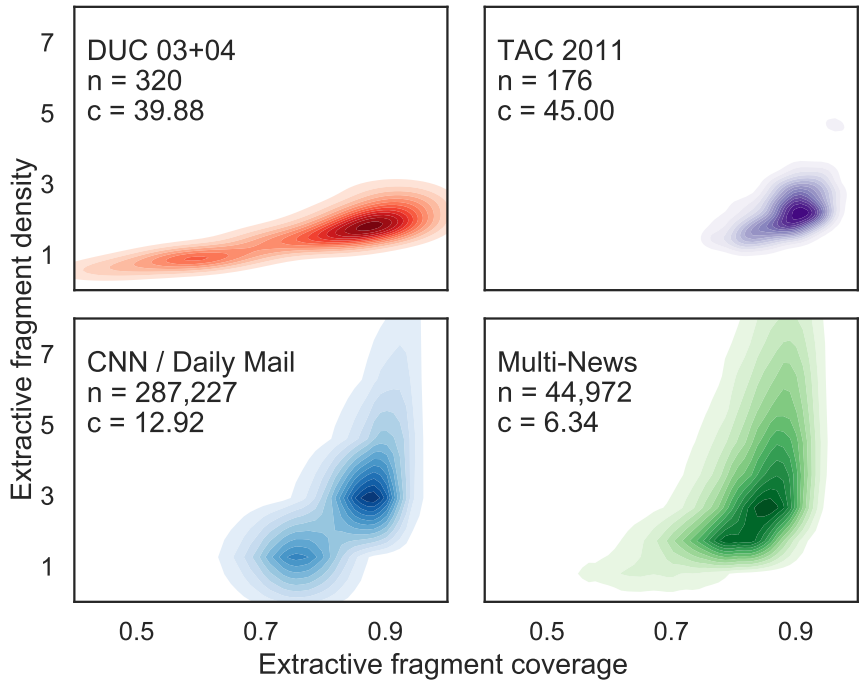

Figure 4.1: Density estimation of extractive diversity scores as explained in Section 4.3. We include scores for a standard SDS dataset (CNNDM) and MDS datasets from DUC and TAC, along with Multi-News. Large variability along the y-axis suggests variation in the average length of source sequences present in the summary, while the x axis shows variability in the average length of the extractive fragments to which summary words belong.

4.4 Hi-MAP Model

In this section, we provide the details of our Hierarchical MMR-Attention Pointer-generator (Hi-MAP) model for multi-document neural abstractive summarization.

MMR Maximal Marginal Relevance (MMR) is an approach for combining query-relevance with information-novelty in the context of summarization (Carbonell and Goldstein, 1998). MMR produces a ranked list of the candidate sentences based on the relevance and redundancy to the query, which can be used to extract sentences. The score is calculated as follows:

$$\text{MMR} = \underset{D_i \in R \setminus S}{\operatorname{argmax}} \left[\lambda \text{Sim}(D_i, Q) - (1 - \lambda) \max_{D_j \in S} \text{Sim}(D_i, D_j) \right] \tag{4.4}$$

where R is the collection of all candidate sentences, Q is the query, S is the set of sentences

that have been selected, $R \setminus S$ is set of the un-selected ones, and Sim is a similarity function such as cosine similarity. In general, each time we want to select a sentence, we have a ranking score for all the candidates that considers relevance and redundancy. A recent work (Lebanoff et al., 2018) applied MMR for multi-document summarization by creating an external module and a supervised regression model for sentence importance. Our proposed method, however, incorporates MMR with the pointer-generator network in an end-to-end manner that learns parameters for similarity and redundancy.

We expand the existing pointer-generator network model into a hierarchical network, which allows us to calculate sentence-level MMR scores. Our model consists of a pointer-generator network and an integrated MMR module, as shown in Figure 4.2.

Sentence representations To expand our model into a hierarchical one, we compute sentence representations on both the encoder and decoder. The input is a collection of sentences $D = [s_1, s_2, .., s_N]$ from all the source documents, where a given sentence $s_i = [x_{k-m}, x_{k-m+1}, ..., x_k]$ is made up of input word tokens. Word tokens from the whole document are treated as a single sequential input to a Bi-LSTM Hochreiter and Schmidhuber (1997) encoder as in the original encoder of the pointer-generator network from See et al. (2017) (see bottom of Figure 4.2). For each time step, the output of an input word token x_l is h_l^w (we use superscript w to indicate word-level LSTM cells, s for sentence-level). To obtain a representation for each sentence s_i, we take the encoder output of the last token for that sentence. If that token has an index of k in the whole document D, then the sentence representation is marked as $h_{s_i}^w = h_k^w$. The word-level sentence embeddings of the document $h_D^w = [h_{s_1}^w, h_{s_2}^w, ..h_{s_N}^w]$ will be a sequence which is fed into a sentence-level LSTM network. Thus, for each input sentence $h_{s_i}^w$, we obtain an output hidden state $h_{s_i}^s$. We then get the final sentence-level embeddings $h_D^s = [h_1^s, h_2^s, ..h_N^s]$ (we omit the subscript for sentences s). To obtain a summary representation, we simply treat the current decoded summary as a single sentence and take the output of the last step of the decoder: s_{sum}. We plan to investigate

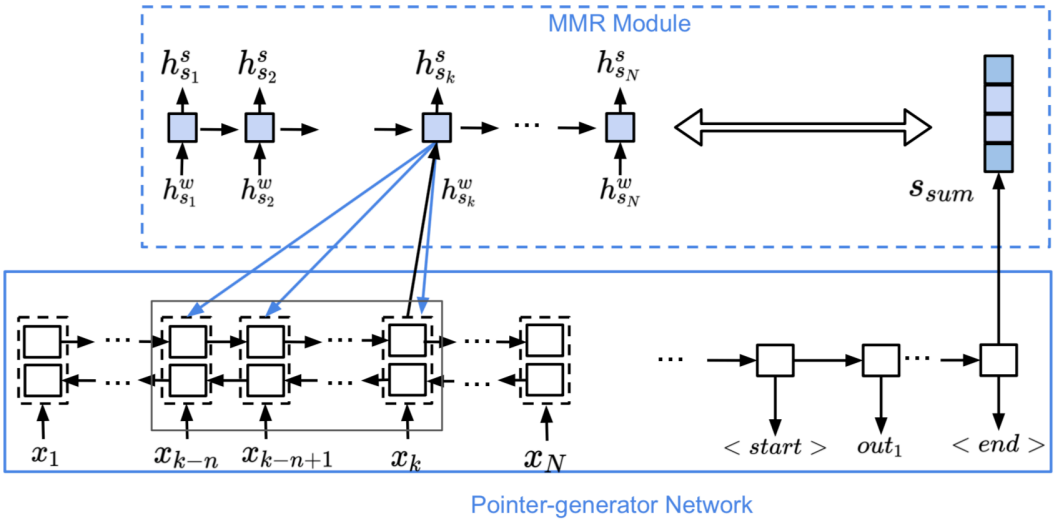

Figure 4.2: Our Hierarchical MMR-Attention Pointer-generator (Hi-MAP) model incorporates sentence-level representations and hidden-state-based MMR on top of a standard pointer-generator network.

alternative methods for input and output sentence embeddings, such as separate LSTMs for each sentence, in future work.

MMR-Attention Now, we have all the sentence-level representation from both the articles and summary and then we apply MMR to compute a ranking on the candidate sentences h^s_D. Intuitively, incorporating MMR will help determine salient sentences from the input at the current decoding step based on relevancy and redundancy. We follow Section 4.3 to compute MMR scores. Here, however, our query document is represented by the summary vector s_{sum}, and we want to rank the candidates in h^s_D. The MMR score for an input sentence i is then defined as:

$$\text{MMR}_i = \lambda \text{Sim}_1(h^s_i, s_{sum}) - (1-\lambda)\text{score}_i \tag{4.5}$$

We then add a softmax function to normalize all the MMR scores of these candidates as a probability distribution.

$$\overline{\text{MMR}_i} = \frac{\exp(\text{MMR}_i)}{\sum_i \exp(\text{MMR}_i)} \tag{4.6}$$

Now we define the similarity function between each candidate sentence h_i^s and summary sentence s_{sum} to be:

$$\text{Sim}_1 = h_i^{sT} W_{\text{Sim}} s_{sum} \tag{4.7}$$

where W_{Sim} is a learned parameter used to transform s_{sum} and h_i^s into a common feature space. For the second term of Equation 4.5, instead of choosing the maximum score from all candidates except for h_i^s, which is intended to find the candidate most similar to h_i^s, we choose to apply a self-attention model on h_i^s and all the other candidates $h_j^s \in h_D^s$. We then choose the largest weight as the final score:

$$\begin{aligned} v_{ij} &= \tanh\left(h_j^{sT} W_{self} h_i^s\right) \\ \beta_{ij} &= \frac{\exp\left(v_{ij}\right)}{\sum_j \exp\left(v_{ij}\right)} \\ \text{score}_i &= \max_j(\beta_{i,j}) \end{aligned} \tag{4.8}$$

Note that W_{self} is also a trainable parameter. Eventually, the MMR score from Equation 4.5 becomes:

MMR-attention Pointer-generator After we calculate $\overline{\text{MMR}}_i$ for each sentence representation h_i^s, we use these scores to update the word-level attention weights for the pointer-generator model shown by the blue arrows in Figure 4.2. Since $\overline{\text{MMR}_i}$ is a sentence weight for h_i^s, each token in the sentence will have the same value of $\overline{\text{MMR}_i}$. The new attention for each input token from Equation 2.2 becomes:

$$\overline{a^t} = a^t \overline{\text{MMR}_i} \tag{4.9}$$

4.5 Experiments

In this section, we describe additional methods we compare with and present our assumptions and experimental process.

Baseline and Extractive Methods

First We concatenate the first sentence of each article in a document cluster as the system summary. For our dataset, *First-k* means the first k sentences from each source article will be concatenated as the summary.

LexRank Initially proposed by (Erkan and Radev, 2004), LexRank is a graph-based method for computing relative importance in extractive summarization.

TextRank Introduced by (Mihalcea and Tarau, 2004), TextRank is a graph-based ranking model. Sentence importance scores are computed based on eigenvector centrality within a global graph from the corpus.

MMR In addition to incorporating MMR in our pointer generator network, we use this original method as an extractive summarization baseline. When testing on Multi-News data, we set these extractive methods to output 300 tokens.

Neural Abstractive Methods

PG-MMR This is the modified pointer-generator network model reported by (Lebanoff et al., 2018).

PG-BRNN The PG-BRNN model is a pointer-generator implementation from OpenNMT[6]. As in the original paper (See et al., 2017), we use a 1-layer bi-LSTM as encoder, with 128-dimensional word-embeddings and 256-dimensional hidden states for each direction. The decoder is a 512-dimensional single-layer LSTM. We include this for reference, as our Hi-MAP code builds upon this implementation.

CopyTransformer Instead of using an LSTM, the CopyTransformer model used in Gehrmann et al. (2018) uses a 4-layer Transformer of 512 dimensions for encoder and decoder. One of the attention heads is chosen randomly as the copy distribution. This model and the PG-BRNN are run without the bottom-up masked attention for inference from Gehrmann et al. (2018) as we did not find a large improvement when reproducing the model on this data.

Experimental Setting Following the setting from (Lebanoff et al., 2018), we report ROUGE (Lin, 2004a) scores, which measure the overlap of unigrams (R-1), bigrams (R-2), and longest common subsequence (R-L). For the neural abstractive models, we truncate input articles to 500 tokens in the following way: for each example with S source input documents, we take the first $500/S$ tokens from each source document. As some source documents may be shorter, we iteratively determine the number of tokens to take from each document until the 500 token quota is reached. Having determined the number of tokens per source document to use, we concatenate the truncated source documents into a single mega-document. This effectively reduces MDS to SDS on longer documents, a commonly-used assumption for recent neural MDS papers (Cao et al., 2017; Liu et al., 2018; Lebanoff et al., 2018). We chose 500 as our truncation size as related MDS work did not find a large improvement when increasing input length from 500 to 1000 tokens (Liu et al., 2018). We simply introduce a special token between source documents to aid our

6. https://github.com/OpenNMT/OpenNMT-py/blob/master/docs/source/Summarization.md

Method	R-1	R-2	R-L
First-3	40.65	12.64	36.57
LexRank (Erkan and Radev, 2004)	40.94	12.58	36.84
TextRank (Mihalcea and Tarau, 2004)	42.55	13.47	38.39
MMR (Carbonell and Goldstein, 1998)	41.62	11.58	37.80
PG-MMR (Lebanoff et al., 2018)	41.89	13.41	-
PG-BRNN (Gehrmann et al., 2018)	43.88	15.25	39.77
CopyTransformer (Gehrmann et al., 2018)	**44.57**	15.04	**40.40**
Hi-MAP (Our Model)	44.52	**16.00**	40.33

Table 4.5: ROUGE scores for models trained and tested on the Multi-News dataset.

Method	Informativeness	Fluency	Non-Redundancy
PG-MMR	51	43	27
Hi-MAP	46	46	60
CopyTransformer	53	61	64
Human	150	150	149

Table 4.6: Number of times a system was chosen as best in pairwise comparisons according to informativeness, fluency and non-redundancy.

models in detecting document-to-document relationships and leave direct modeling of this relationship, as well as modeling longer input sequences, to future work. We hope that the dataset we introduce will promote such work. For our Hi-MAP model, we applied a 1-layer bidirectional LSTM network, with the hidden state dimension 256 in each direction. The sentence representation dimension is also 256. We set the $\lambda = 0.5$ to calculate the MMR value in Equation 4.5.

4.6 Analysis and Discussion

In Table 4.5 we report ROUGE scores on the Multi-News dataset. Additionally, for Multi-News testing, we experimented with using the output of 500 tokens from extractive methods (LexRank, TextRank, and MMR) as input to the abstractive model. However, this did not improve results. We believe this is because our truncated input mirrors the First-3 baseline, which outperforms these three extractive methods and thus may provide more information as input to the abstractive model. Our model outperforms PG-MMR when trained and tested

on the Multi-News dataset. We see much-improved model performances when trained and tested on in-domain Multi-News data. The Transformer performs best in terms of R-1 and R-L while Hi-MAP outperforms it on R-2. Our PG-MMR results correspond to *PG-MMR w Cosine* reported in Lebanoff et al. (2018). We trained their sentence regression model on Multi-News data and leave the investigation of transferring regression models from SDS to Multi-News for future work. In addition to automatic evaluation, we performed a human evaluation to compare the summaries produced. We used pairwise summary comparison as in Narayan et al. (2018a). Annotators were presented with the same input that the systems saw at testing time; input documents were truncated, and we separated input documents by visible spaces in our annotator interface. We chose three native English speakers as annotators. They were presented with input documents and summaries generated by two out of four systems and were asked to determine which summary was better and which was worse in terms of *informativeness* (is the meaning in the input text preserved in the summary?), *fluency* (is the summary written in well-formed and grammatical English?) and *non-redundancy* (does the summary avoid repeating information?). We randomly selected 50 documents from the Multi-News test set and compared all possible combinations of two out of four systems. We chose to compare PG-MMR, CopyTransformer, Hi-MAP, and gold summaries. The order of summaries was randomized per example. The results of our pairwise human-annotated comparison are shown in Table 4.6. Human-written summaries were easily marked as better than other systems, which, while expected, shows that there is much room for improvement in producing readable, informative summaries. We performed a pairwise comparison of the models over the three metrics combined, using a one-way ANOVA with Tukey HSD tests and p value of 0.05. Overall, statistically significant differences were found between human summaries score and all other systems, CopyTransformer and the other two models, and our Hi-MAP model compared to PG-MMR. Our Hi-MAP model performs comparably to PG-MMR on informativeness and fluency but much better in terms of non-redundancy. We believe that the incorporation of learned

parameters for similarity and redundancy reduces redundancy in our output summaries. In future work, we would like to incorporate MMR into Transformer models to benefit from their fluent summaries.

4.7 Summary

in this chapter, we introduced Multi-News, the first large-scale multi-document news summarization dataset. We hope that this dataset will promote work in multi-document summarization similar to the progress seen in the single-document case. Additionally, we introduce an end-to-end model that incorporates MMR into a pointer-generator network, which performs competitively compared to previous multi-document summarization models. We also benchmark methods on our dataset. In the future, we plan to explore interactions among documents beyond concatenation and experiment with summarizing longer input documents.

Chapter 5

Scientific Topic Summarization: an Application

This chapter synthesizes ideas from the previous two, namely viewing the topic summarization task from Chapter 3 as a two-step multi-document summarization. We begin with a preliminary task and are the first to apply novel pretraining techniques for generating the lead paragraph of a Wikipedia article. We show that recent advances in pretrained language modeling can be combined for an improved two-stage extractive and abstractive approach for Wikipedia lead paragraph generation. However, when we extend this approach to generate longer Wikipedia-style summaries and examine, we see how such methods struggle through comparison studies with reference human-collected summaries.

5.1 Introduction

Fast-developing fields such as Artificial Intelligence (AI) often outpace the efforts of encyclopedic sources such as Wikipedia, which either do not completely cover recently-introduced topics or lack such content entirely. A pipeline for automatically creating such Wikipedia pages is thus desirable. While there has been some work on generating full Wikipedia pages, these efforts are either domain-specific (Sauper and Barzilay, 2009), make strong

assumptions about the topics being summarized (Banerjee and Mitra, 2016), or are purely extractive (Jha et al., 2015). In a related line of work, query-based summarization has been applied to specific sections of Wikipedia pages Deutsch and Roth (2019); Zhu et al. (2019), which can be viewed as a more self-contained version of Wikipedia page generation. Recent Wikipedia page generation work has focused on generating the initial leading paragraph of a Wikipedia page (Liu et al., 2018; Liu and Lapata, 2019b; Perez-Beltrachini et al., 2019). These papers consist of a two-step framework by which an extractive method selects relevant content for a specific topic, and an abstractive method generates the final summary of the topic.

In this chapter, we first examine how recently-introduced pretrained language models (Devlin et al., 2019; Liu et al., 2019; Lewis et al., 2020) improve upon both the extractive and abstractive steps of previous models for the task of lead paragraph generation. We further focus on an analysis of the extension of such methods to full Wikipedia page generation on scientific topics related to AI and NLP. We manually create summaries of 25 AI and NLP topics divided along sections, as on Wikipedia pages. We perform ablation studies on content selection and generation methods over these topics, finding that current content selection methods are not precise and fail to differentiate content well among queries for subtopics of the main topic.

Our contributions here are: 1) We demonstrate how recent advances in pretrained language models improve upon Wikipedia lead paragraph generation. 2) We extend the current Wikipedia introduction paragraph generation techniques to generate full Wikipedia-style pages of scientific topics and provide an analysis of the full summaries. 3) We study the problems encountered in this application and point to areas of improvement for future work. We provide a better understanding of current methods and their faults in a real-world application.

5.2 Pretraining Wikipedia Lead Paragraph Generation

In this section, we show how combining recent methods for a two-staged approach of content selection and generation gives improved results on the WikiSum dataset (Liu et al., 2018) as well as a newly curated set of Wikipedia articles.

Data We make use of the **WikiSum** dataset (Liu et al., 2018), a collection of over 1.5 million Wikipedia pages and their references. Applying pretraining techniques such as BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), and BART (Lewis et al., 2020), however, poses a problem with this task, as these models make use of Wikipedia during pretraining. To address this problem, we mirror the process of Liu et al. (2018) to collect an unbiased dataset of newly added Wikipedia pages[7] which did not appear in pretraining, (**NewPage WikiSum**). We collect 10,000 of the newest Wikipedia pages, scrape Wikipedia for their references and return the top 10 Google Search results. We remove non-English results and remove any articles for which we were not able to scrape a single reference. Due to the sparsity of search results on specific topics, we were left with about 1,000 articles, which we use as a test set.

Step One: Content Selection We experiment with five approaches for our initial content-selection step. **TF-IDF**: a simple approach to extract relevant content is to use term frequency–inverse document frequency (Liu et al., 2018; Fan et al., 2019). **LSTM-Rank**: Liu and Lapata (2019b) approach query-based content selection as a regression problem of predicting the ROUGE-2 recall of a given paragraph-topic pair. **WikiCite**: Deutsch and Roth (2019) approach query-based summarization via an extractive classification approach with attention (Bahdanau et al., 2015) over the topic and context.

We apply two additional methods to the task of content selection. **Semantic Search**: Reimers and Gurevych (2019a) fine-tune BERT and Roberta using siamese and triplet

7. https://en.wikipedia.org/wiki/Special:NewPages

Methods	**L=5**	**L=10**	**L=20**	**L=40**
TF-IDF	24.86	32.43	40.87	49.49
LSTM-Rank	39.38	46.74	53.84	60.42
WikiCite	**65.27**	69.77	73.54	76.51
Semantic Search	34.87	48.60	61.87	74.54
RoBERTa-Rank	64.12	**72.49**	**79.17**	**84.28**

Table 5.1: ROUGE-L-Recall scores for WikiSum content selection, varying the number of paragraphs returned.

networks to produce fixed-length vectors, which can be compared using cosine similarity to find semantically similar input. We embed the title of each Wikipedia page, and each candidate paragraph, using this method, and choose the paragraphs with the most similar vectors to the title as selected content. **RoBERTa-Rank**: we train RoBERTa similar to the approach of (Liu and Lapata, 2019b), treating the title and paragraph to be ranked as sentence pairs and use predicted relevance scores as a ranking function for determining the most relevant paragraphs. For training RoBERTa-Rank, we sampled 1,209,387 training and 10,000 validation paragraphs from the original WikiSum dataset. For training RoBERTa-Rank, we train with a polynomial decay learning rate scheduler with learning rate $2\mathrm{e}{-5}$, using the Adam optimizer (Kingma and Ba, 2015). We train with 6000 warmup steps and 10,0000 total steps. By the end of training, the validation loss is practically 0. The model has 356,461,658 parameters, building off of RoBERTa large. This model was also trained on 8 16 GB V100 GPUs for about a day.

We show the results in Table 5.1. WikiCite performs well despite not including extensive pretraining and without fine-tuning on the WikiSum data, perhaps because the model is trained for the task of fine-grained selection (for section titles within a given page). RoBERTa-Rank is the highest-scoring content selector except for the 5-paragraph case, so then we choose this as the content selection method for abstractive summarization input on WikiSum data.

Step Two: Abstractive Summarization We use the RoBERTa-Rank content selection component to select paragraphs up to 1024 total tokens as input to our abstractive summarization step. As the abstractive model in our two-step approach, we experiment with **BART** (Lewis et al., 2020), discussed in 2. We compare BART fine-tuned on the WikiSum data with the previous state-of-the-art **HierSumm** model from Liu and Lapata (2019b). For training the WikiSum component, we took a subset of the original WikiSum dataset consisting of 280,000 training instances and 10,000 validation instances. We removed paragraphs that were clones of the target summary through a threshold of .5 ROUGE-2 score. We then sort the instances according to the sum of the ROUGE scores of individual paragraphs and take the paragraphs with the highest scores for training and validation. This was done to filter out examples with poorly collected source documents and promote a stronger connection between the source documents and the target summary. The number of training examples was chosen to be close to the number found in the CNN-DailyMail dataset. For training BART on the above WikiSum data, we train with a polynomial decay learning rate scheduler with a learning rate of $3e{-}5$, using the Adam optimizer (Kingma and Ba, 2015). We train with 500 warmup steps and 20,000 total steps, ending with a validation loss of 3.492. The max-tokens per batch is 1024, and an update frequency for gradient accumulation is 8. The model is the same as the BART large model released by Facebook, without any additional parameters, for a total of 405,766,144 parameters. This model was trained on 8 16 GB V100 GPUs for about 10 hours.

We show improved results on generating the introduction paragraph on WikiSum and on our NewPage WikiSum data in Table 5.2. We use the same RoBERTa content selection algorithm for both models on NewPage WikiSum. BART generation still outperforms HierSumm. We note that the large difference in scores between that of the WikiSum data and on our collected subset is likely due to the widespread nature of topics in WikiSum; WikiSum includes many well-established topics for which finding reference documents is simple, while the newly introduced topics may not contain enough reference information for

Dataset	**Hiersumm**	**BART**
WikiSum	41.53/26.52/35.76	**46.61/26.82/43.25**
NewPage	31.64/15.06/27.13	**39.29/18.56/36.03**

Table 5.2: ROUGE-1/2/L scores for intro paragraph generation on WikiSum and NewPage WikiSum.

a higher-quality generation. So far, we have shown that applying RoBERTa-Rank and BART as a two-step pipeline gives promising results in generating lead Wikipedia sections.

5.3 Application of Pipeline to Full Wikipedia Generation

We follow Banerjee and Mitra (2016) in extending a two-step pipeline to full Wikipedia-style summaries (section by section content selection and summarization) to study the applicability of recent methods in this real-world setting.

Data Testing our models on full Wikipedia-page data would again face the problem of pretraining bias, and large-scale collection of full-size Wikipedia pages for novel topics is not infeasible. Furthermore, we focus on generating Wikipedia pages for AI-related topics. We picked a mixture of NLP and broader AI-related topics to include eight topics with existing Wikipedia pages as well as those without pages or stub articles, with 25 topics in total. We randomly chose 10 for initial ablation studies and left the remaining 15 for final analysis, which are shown in Table 5.3 and Table 5.4, respectively. We asked five students in NLP to follow the following procedure for creating summaries.

We define a template for the surveys consisting of five sections: **Introduction**, **History**, **Key Ideas**, **Variations** (similar topics or topics with similar goals) and **Applications**. We arrived at these section titles by an examination of sample Wikipedia pages in NLP. First, we searched Google for the given topic, retrieving all HTML page links for the first two search result pages. We then have the annotator read each page, extract relevant content into the corresponding section, and paraphrase and summarize the relevant content for each

Topics
AdaGrad (optimizer)
ADAM (optimizer)
Attention mechanism (deep learning)
BERT
Convolutional Neural Networks
Image captioning (deep learning)
Knowledge graphs
Recursive neural networks
RMSProp (optimizer)
Sentiment Analysis

Table 5.3: A list of the topics used for ablation studies.

Topics
Automatic Summarization
Coreference Resolution
Decision Boundary
Dialogue State Tracking
Document-term Matrix
Dropout (neural networks)
GANs
Highway Networks
HMMs
LSTMs
Machine Translation
Pretrained Language Models
Topic Models
Word2Vec
XGBoost Algorithm

Table 5.4: A list of the topics used for final analysis.

section to between 50 and 150 words per section. We will make all data public.

Content Selection We first tested the quality of the content selection methods for the generic retrieval of content relevant to a topic on our data. We choose the Semantic Search, WikiCite, and RoBERTa-Rank methods from Table 5.1 for analysis. For Semantic Search, we experiment with three types of sentence embeddings, the original sentence-transformer BERT embeddings (**Search-base**), embeddings fine-tuned with SciBERT (**Search-SciBERT**), and a version fine-tuned to differentiate whether two paragraphs

Methods	**AvgP@10 (before)**	**AvgP@10**
Search-base	0.20	4.05
Search-Scibert	0.30	5.00
Search-Wiki	0.00	4.50
WikiCite	**3.40**	**6.35**
RoBERTa	0.45	6.05

Table 5.5: Comparison of retrieved results across content selection methods before and after filtering sentences.

belong to the same Wikipedia section (**Search-Wiki**). The parsed output naturally contains some poorly parsed paragraphs, which consist of single words, short sentences, or jumbled equations. Surprisingly, we found such content was often returned during retrieval despite the poor grammaticality and relevance. We hypothesize that the tendency to return short sentences, often with odd punctuation, may relate to the extension of these methods to paragraph levels while inherently being developed for sentence-level tasks.

We then remove sentences shorter than six tokenized words, as well as apply heuristics for removing sentences based on the number of parentheses, brackets, and other tokens such as equal signs. We required that each paragraph returned consist of at least two sentences and required that the topic word (or one word within the topic, for multi-word topics) appear in the paragraph. About 85 paragraphs per topic remain after this filtering. The comparison of results before and after preprocessing and filtering is found in Table 5.5. Notably, the WikiCite method performs much better than other methods before applying any preprocessing. We believe this is because the method is trained for content selection based on a topic and not simply trained for returning content with high recall. A potential problem with current methods in this two-step approach is that content selection is trained and evaluated with recall in mind to capture as large a range of the topic, which produces models without the precision necessary in a real-world application. This aligns with previous work in extractive summarization, suggesting that optimizing for recall gives suboptimal results (Zopf et al., 2018).

Section-Specific Content Selection We investigated the ability of our content selection models to retrieve content specific for each chosen section, for example, querying "History of BERT"rather than "BERT."We observed large overlaps between the returned results, between 5 and 9 paragraph overlap between the top 10 results for each section. Among all methods, Wikicite has the least overlap. As an alternative method to select distinct content for each section, we investigate clustering methods, using out-of-the-box Agglomerative (Müllner, 2011) clustering provided by scikit-learn[8]. We cluster the embeddings obtained before the final output layer from the WikiCite and RoBERTa methods and the Search-Wiki embeddings. We annotated the coherence of each cluster. Clusters obtained using embeddings from RoBERTa, Search-Wiki and WikiCite had a corresponding average coherence of 3.07, 3.40, and 3.52 on a 1-5 scale, signaling slightly above-average coherence for each clustering. Again, the poor performance of RoBERTa in clustering may be due to the more general topic training method. As suggested by Deutsch and Roth (2019), the WikiCite method may dilute topic information in the final layer despite topic attention in previous layers and thus benefit from using embeddings before the final layer as clustering.

Abstractive Summarization

Generation Model Choice To perform an ablation study on the choice of generation model, we took the best performing WikiCite retrieval method and used the selected content for the introduction paragraph as input to BART. We experimented with two BART models, our BART model fine-tuned on WikiSum as well as one fine-tuned on the CNN-DailyMail summarization dataset (Hermann et al., 2015). We labeled for the presence of any hallucinations in the summary. Additionally, we manually rated the summaries from 1-5 for the relevance of the content to the particular topic. Results are shown in Table 5.6. As seen in the Table, we find much fewer hallucinations when using the CNN-DailyMail model

8. https://scikit-learn.org/stable/index.html

Method	**Avg. # Hallucinations**	**Avg. Relevance**
BART-WikiSum	0.5	4.35
BART-CNNDM	0.1	4.55

Table 5.6: A comparison of the number of hallucinations and the relevance of Wikipedia introduction paragraph generation on our ablation study topics.

Method	**Relevance**	**Non-redundancy**
Retrieval Search-Wiki	3.07	2.30
Cluster Search-Wiki	3.83	3.97
Cluster WikiCite	4.0	**4.07**
Cluster RoBERTa	**4.12**	3.86
GOLD	**4.73**	**4.63**

Table 5.7: A comparison of the average relevance and non-redundancy of the final generated surveys (higher is better for both).

versus the one trained on WikiSum. We hypothesize that this is due to dataset biases; the CNN-DailyMail dataset is more extractive and closely linked to the source document, so the model does not stray far from the input text. We believe that fine-tuning BART on other datasets could lead to additional improvements, especially by training on scientific text or a more focused task such as WikiCite data (Deutsch and Roth, 2019).

Analysis of Full Summaries We take the clustering output for the three embedding methods in the previous section (**Cluster Search-Wiki**,**Cluster WikiCite**, and **Cluster RoBERTa**) as well as the Search-Wiki retrieval output(**Retrieval Search-Wiki**) as input to our generation component to create full sectioned summaries. We evaluate the model outputs for relevance and redundancy on a 1-5 Likert scale. Results are shown in Table 5.7. We only see substantial differences between the retrieval and clustering methods and between the clustering and human-created summaries. This confirms our previous troubles with selecting relevant and non-redundant content for different sections of the survey and shows room for improvement. The ranking of clustering embeddings by coherence corresponds to redundancy in final surveys, suggesting the need to focus on retrieving or clustering distinct content and not relying on the abstractive summarization module, which has been the focus of recent work. We find certain stylistic features present in the surveys do not

Introduction
Text summarization is an interesting machine learning field that is increasingly gaining traction. As research in this area continues , we can expect to see breakthroughs that will assist in fluently and accurately shortening long text documents. In this article, we look at how machine learning can be used to help shorten text.
History
Summarization has been and continues to be a hot research topic in the data science arena. While text summarization algorithms have existed for a while , major advances in natural language processing and deep learning have been made in recent years. Google has reportedly worked on projects that attempt to understand novels. Summarization can help consumers quickly understand what a book is about.
Key Ideas
Automatic summarization aims to produce a shorter version of an input text, preserving only the essential information. There are two main types of summarization : extractive summarization selects important sentences from the input and abstractive summarizing generates content without explicitly re-using whole sentences. In our new paper , we constructed two novel , large-scale summarization datasets from scientific journal articles.
Variations
Multi-document summarization can be a powerful tool to quickly analyze dozens of search results. MeaningCloud 's Summarization API locates the most relevant phrases in a document and builds a synopsis with them. More specific summarization systems could be developed to analyze legal documents.
Applications
Summarization can be a crucial component in the tele-health supply chain when it comes to analyzing medical cases. The Spreading Activation approach does not allow to improve our results. Tables 8 and 9 show the high recall obtained with these methods, which may be a very interesting feature in some cases.

Table 5.8: Sample survey of the topic `Text Summarization` created using our automated pipeline, showing both the ability of our pipeline to capture important content as well as problems related to the style of presentation, such as references to input Tables.

match Wikipedia pages. For example, some content is stated in the first person: “In this paper, we...”This is an artifact of the generation model and the content extracted and can likely be remedied by fine-tuning BART in a different setting. We present two examples of the summaries in Table 5.8 and Table 5.9, respectively.

Introduction
Dropout is a technique where randomly selected neurons are ignored during training. This means that their contribution to the activation of downstream neurons is removed. Dropout alone does not have any way to prevent parameter values from becoming too large during this update phase. In the example below we add a new Dropout layer between the input (or visible layer) and the first hidden layer. The dropout rate is set to 20%, meaning one in 5 inputs will be randomly excluded from each update cycle.
History
Classical generalization theory suggests that to close the gap between train and test performance , we should aim for a simple model. Christopher Bishop formalized this idea when he proved that training with input noise is equivalent to Tikhonov regularization. In 2014, Srivastava et al. developed a clever idea for how to apply Bishop 's idea to the internal layers of the network. They proposed to inject noise into each layer of the Network before calculating the subsequent layer.
Key Ideas
Additionally , as recommended in the original paper on Dropout , a constraint is imposed on the weights for each hidden layer. This is done by setting the kernel˙constraint argument on the Dense class when constructing the layers. In the example below Dropout is applied between the two hidden layers and between the last hidden layer and the output layer.
Variations
With a Gaussian-Dropout , the expected value of the activation remains unchanged. Unlike the regular Dropout , no weight scaling is required during inferencing. Dropout is only used during the training of a model and is not used when evaluating the skill of the model. The main problem hindering dropout in NLP has been that it could not be applied to recurrent connections.
Applications
During training time , dropout randomly sets node values to zero. During inference time, dropout does not kill node values, but all the weights in the layer were multiplied. This multiplier could be placed on the input values rather than the weights. TensorFlow has its own implementation of dropout which only does work during training time.

Table 5.9: Sample survey of the topic of `Dropout`. Some stylistic problems such as references to examples described in the original document are present, although key concepts of the topic are addressed.

5.4 Summary

In this chapter, we demonstrated improvements in individual components of Wikipedia summarization through an application of recently-introduced embedding and summarization techniques but largely focus on the failures of these methods when extended in a real-world scenario of full-page Wikipedia-styled summarization. We believe that a focus on high-precision and fine-grained query-based summarization in future work will help make this pipeline viable.

In the last few chapters, we have shown that large-scale data allows for the application of neural network models, which can achieve state-of-the-art results when trained on this data. However, as shown in this chapter, the blind application of these models to similar tasks does not always give reasonable results. Thus, it is necessary to make smarter use of data available and design models to allow for their application when data is not available for the precise task at hand, which will be the focus of the next part of this dissertation.

Part II

Low-resource Text Summarization

Chapter 6

Template-Based Question Generation from Retrieved Sentences for Improved Unsupervised Question Answering

We now turn to data-efficient methods for more realistic application of neural networks. We address the task of unsupervised, extractive question answering. Extractive question answering can be viewed as a form of query-based summarization where the output summary is a short phrase or word from the input context. While recent work has achieved state-of-the-art performance in supervised question answering (QA), we tackle the more realistic problem of QA when no data is available in a domain. We propose an unsupervised approach to training QA models with generated pseudo-training data. We show that generating questions for QA training by applying a simple template on a related, retrieved sentence rather than the original context sentence improves downstream QA performance by allowing the model to learn more complex context-question relationships. Training a QA model on this data gives a relative improvement over a previous unsupervised model in F1 score on the SQuAD QA dataset (Rajpurkar et al., 2016) by about 14%, and 20% when the answer is a named entity, achieving state-of-the-art performance on SQuAD for unsupervised QA.

6.1 Introduction

Question Answering aims to answer a question based on a given knowledge source and is in increasing demand as the amount of information available online and the desire for quick access to this content grows. Recent advances have driven the performance of QA systems to above or near-human performance on QA datasets such as SQuAD (Rajpurkar et al., 2016) and Natural Questions (Kwiatkowski et al., 2019) thanks to pretrained language models such as BERT (Devlin et al., 2019), XLNet (Yang et al., 2019) and RoBERTa (Liu et al., 2019). Fine-tuning these models, however, requires large-scale data for fine-tuning. Creating a dataset for every new domain is extremely costly and practically infeasible. The ability to apply QA models on out-of-domain data in an efficient manner is thus very desirable. This problem may be approached with domain adaptation or transfer learning techniques (Chung et al., 2018) as well as data augmentation (Yang et al., 2017; Dhingra et al., 2018; Wang et al., 2018; Alberti et al., 2019). However, here we expand upon the recently introduced task of unsupervised question answering (Lewis et al., 2019) to examine the extent to which synthetic training data alone can be used to train a QA model. In particular, we focus on the machine reading comprehension setting in which the context is a given paragraph, and the QA model can only access this paragraph to answer a question. Furthermore, we work on extractive QA, where the answer is assumed to be a contiguous substring of the context. A training instance for supervised reading comprehension consists of three components: a *question*, a *context*, and an *answer*. For a given dataset domain, a collection of documents can usually be easily obtained, providing *context* in the form of paragraphs or sets of sentences. *Answers* can be gathered from keywords and phrases from the context. We focus on factoid QA; the question concerns a concise fact. In particular, we emphasize questions whose answers are named entities, the majority type of factoid questions. Entities can be extracted from text using named entity recognition (NER) techniques as the training instance's *answer*. Thus, the main challenge, and the focus of this chapter, is creating a relevant *question* from a *(context, answer)* pair in an unsupervised manner.

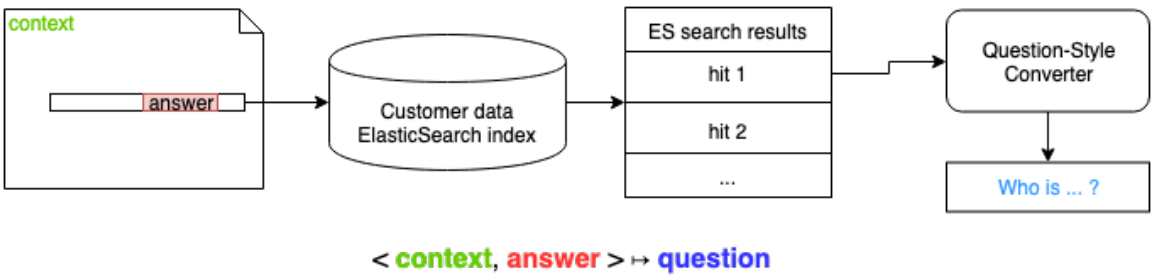

Figure 6.1: Question Generation Pipeline: the original context sentence containing a given answer is used as a query to retrieve a related sentence containing matching entities, which is input into our question-style converter to create QA training data.

Recent work of Lewis et al. (2019) uses style transfer for generating questions for *(context, answer)* pairs but shows little improvement over applying a much simpler question generator which drops, permutates, and masks words. We improve upon this paper by proposing a simple, intuitive, **retrieval and template-based question generation** approach, illustrated in Figure 6.1. The idea is to retrieve a sentence from the corpus similar to the current context and then generate a *question* based on that sentence. Having created a *question* for all *(context, answer)* pairs, we then fine-tune a pretrained BERT model on this data and evaluate on the SQuAD v1.1 dataset (Rajpurkar et al., 2016).

Our contributions are as follows: we introduce a retrieval, template-based framework which achieves state-of-the-art results on SQuAD for unsupervised models, particularly when the answer is a named entity. We perform ablation studies to determine the effect of components in template question generation. We are releasing our synthetic training data and code.[9]

6.2 Unsupervised Question Answering Data Creation

We focus on creating high-quality, non-trivial questions that will allow the model to learn to extract the proper answer from a context-question pair.

9. `https://github.com/awslabs/unsupervised-qa`

Sentence Retrieval A standard cloze question can be obtained by taking the original sentence in which the answer appears from the context and masking the answer with a chosen token. However, a model trained on this data will only learn text matching and how to fill-in-the-blank, with little generalizability. For this reason, we chose to use a retrieval-based approach to obtain a sentence similar to that which contains the answer, upon which to create a given question. For our experiments, we focused on answers which are named entities, which has proven to be a useful prior assumption for downstream QA performance (Lewis et al., 2019) confirmed by our initial experiments. First, we indexed all of the sentences from a Wikipedia dump using the ElasticSearch search engine. We also extract named entities for each sentence in both the Wikipedia corpus and the sentences used as queries. We assume access to a named-entity recognition system, and in this work, make use of the spaCy[10] NER pipeline. Then, for a given context-answer pair, we query the index, using the original context sentence as a query, to return a sentence which (1) contains the answer, (2) does not come from the *context*, and (3) has a lower than 95% F1 score with the query sentence to discard highly similar or plagiarized sentences. Besides ensuring that the retrieved sentence and query sentence share the answer entity, we require that at least one additional matching entity appears in both the query sentence and in the entire context, and we perform ablation studies on the effect of this matching below. These retrieved sentences are then fed into our question-generation module.

Template-based Question Generation We consider several question styles (1) generic cloze-style questions where the answer is replaced by the token "[MASK]", (2) templated question "Wh+B+A+?" as well as variations on the ordering of this template, as shown in Figure 6.2. Given the retrieved sentence in the form of `[Fragment A] [Answer] [Fragment B]`, the templated question "Wh+B+A+?" replaces the *answer* with a Wh-component (e.g., what, who, where), which depends on the entity type of the *answer* and

10. `https://spacy.io`

(context, answer) → question

Context: On February 10, 2007, Barack Obama, then-junior United States Senator from Illinois, announced his candidacy for the presidency of the United States in Springfield, Illinois. Obama announced his candidacy at the Old State Capitol building, where Abraham Lincoln had delivered his "House Divided" speech. Obama was the main challenger, along with John Edwards, to front-runner Hillary Clinton for much of 2007.

Query: Obama announced his candidacy at the Old State Capitol building, where Abraham Lincoln had delivered his "House Divided" speech.

Retrieved ES sentence: On February 10, 2007, Obama announced his candidacy for President of the United States in front of the Old State Capitol building in Springfield, Illinois.

Cloze-style question: On February 10, 2007, [MASK] announced his candidacy for President of the United States in front of the Old State Capitol building in Springfield, Illinois.

Templated question A + Wh + B + ?: On February 10, 2007, who announced his candidacy for President of the United States in front of the Old State Capitol building in Springfield, Illinois?

Templated question Wh + B + A + ?: Who announced his candidacy for President of the United States in front of the Old State Capitol building in Springfield, Illinois, on February 10, 2007?

Figure 6.2: Example of synthetically generated questions using generic cloze-style questions as well as a template-based approach.

places the Wh-component at the beginning of the question, followed by sentence `Fragment B` and `Fragment A`. For the choice of wh-component, we sample a bi-gram based on prior probabilities of that bi-gram being associated with the named-entity type of the answer. This prior probability is calculated based on named-entity and question bi-gram starters from the SQuAD dataset. This information does not make use of the full context-question-answer and can be viewed as prior information, not disturbing the integrity of our unsupervised approach. Additionally, the choice of wh component does not substantially affect results. For template-based approaches, we also experimented with clause-based templates but did not find substantial differences in performance.

6.3 Extractive Question Answering Experiments

Settings For all downstream question answering models, we fine-tune a pretrained BERT model using the Transformers repository (Wolf et al., 2019) and report ablation study numbers using the base-uncased version of BERT, consistent with Lewis et al. (2019). All models are *trained and validated* on *generated pairs of questions and answers* along with their contexts *tested* on the *SQuAD development set*. The training set differs for each ablation study and will be described below, while the validation dataset is a random set of 1,000 template-based generated data points, which is consistent across all ablation studies. We train all QA models for two epochs, checkpointing the models every 500 steps and choosing the checkpoint with the highest F1 score on the validation set as the best model. All ablation studies are averaged over two training runs with different seeds. Unless otherwise stated, experiments are performed using 50,000 synthetic QA training examples, as initial models performed best with this amount. We will make this generated training data public.

Model Analysis **Effect of retrieved sentences** We test the effect of retrieved vs. original sentences as input to question generation when using generic cloze questions. As shown in Table 6.1, using retrieved sentences improves over using the original sentence, reinforcing our motivation that a retrieved sentence, which may not match the current context trivially, forces the QA model to learn more complex relationships than just simple entity matching. The retrieval process may return sentences that do not match the original context. On a random sample, 15/18 retrieved sentences were judged as entirely relevant to the original sentence. This retrieval is already quite good, as we use a high-quality ElasticSearch retrieval and use the original context sentence as the query, not just the answer word. While we do not explicitly ensure that the retrieved sentence has the same meaning, we find that the search results with entity matching give largely semantically matching sentences. Additionally, we believe the sentences which have loosely related meaning may act as a regularization factor that prevents the downstream QA model from learning only string matching patterns. Along

Training procedure	EM	F1
Cloze-style original	17.36	25.90
Cloze-style retrieved	**30.53**	**39.61**

Table 6.1: Effect of original vs retrieved sentences for generic cloze-style question generation.

these lines, Lewis et al. (2019) found that a simple noise function of dropping, masking, and permuting words was a strong question generation baseline. We believe that loosely related context sentences can act as a more intuitive noise function, and investigating the role of the semantic match of the retrieved sentences is an important direction for future work. For the sections which follow, we only show results of retrieved sentences, as the trend of improved performance held across all experiments.

Effect of template components We evaluate the effect of individual template components on downstream QA performance. Results are shown in Table 6.2. Wh template methods improve largely over the simple cloze templates. "Wh + B + A + ?" performs best among the template-based methods, as having the Wh word at the beginning most resembles the target SQuAD domain and switching the order of Fragment B and Fragment A may force the model to learn more complex relationships from the question. We additionally test the effect of the wh-component and the question mark added at the end of the sentence. Using the same data as "Wh + B + A + ?" but removing the wh-component results in a large decrease in performance. We believe that this is because the wh-component signals the type of possible answer entities, which helps narrow down the space of possible answers. Removing the question mark at the end of the template also results in decreased performance, but not as large as removing the wh-component. This may be a result of BERT pretraining, which expects certain punctuation based on sentence structure. We note that these questions may not be grammatical, which may have an impact on performance. Improving the question quality makes a difference in performance, as seen from the jump from cloze-style questions to template questions. The ablation studies suggest that a combination of question

Template data	EM	F1
Cloze	30.53	39.61
A + Wh + B + ?	45.62	55.44
Wh + A + B + ?	44.08	53.90
Wh + B + A + ?	**46.09**	**56.82**
B + A + ?	37.57	46.41
Wh + B + A	44.87	54.56
Wh_simple + B + A + ?	45.60	56.07
What + B + A + ?	10.24	17.04

Table 6.2: Effect of the order of template, wh word and question mark on downstream QA performance. These results demonstrate the importance of inserting the correct wh word as well as the additional impact of the template order and question mark.

relevance, though matching entities, and question formulation, as described above, determine downstream performance. Balancing those two components is an interesting problem, and we leave improving grammaticality and fluency through means such as language model generation for future experiments.

In the last two rows of Table 6.2, we show the effect of using the wh bi-gram prior on downstream QA training. Using the most-common wh word by grouping named entities into five categories according to Lewis et al. (2019) performs very close to the best-performing wh n-gram prior method while using a single wh-word (what) results in a large decrease in performance. These results suggest that information about named entity type signaled by the wh-word does provide important information to the model, but further information beyond wh-simple does not improve results substantially.

Effect of filtering by entity matching Besides ensuring that the retrieved sentence and query sentence share the answer entity, we require that at least one additional matching entity appears in both the query sentence and the entire context. Results are shown in Table 6.3. Auxillary matching leads to improvements over no matching when using template-based data, with best results using matching with both query and context. Matching may filter some sentences whose topic is too far from the original context. We leave the further investigation

Matching procedure	EM	F1
No matching	41.02	50.81
Query matching	44.76	54.87
Context matching	44.22	55.35
Query + Context matching	**46.09**	**56.82**

Table 6.3: Effect of query and context matching for retrieved input to question generation module on downstream QA performance.

of the effect of retrieved sentence relevance to future work. Notably, Lewis et al. (2019) make use of approximately 4 million synthetic data points in order to train their model. However, we are able to train a model with better performance in much fewer examples and show that such a large subset is unnecessary for their released synthetic training data as well. Figure 6.3 shows the performance from training over random subsets of differing sizes and testing on the SQuAD development data. We sample a random question for each context from the data of Lewis et al. (2019). Even with as little as 10k data points, training from our synthetically generated template-based data with auxiliary matching outperforms the results from ablation studies in Lewis et al. (2019). Using data from our template-based data consistently outperforms that of Lewis et al. (2019). Training on either dataset shows similar trends; performance decreases after increasing the number of synthetic examples past 100,000, likely due to a distributional mismatch with the SQuAD data. We chose to use 50,000 examples for our final experiments with other ablation studies as this number gave a good performance in the initial experiments.

Effect of synthetic training dataset size

Comparison of Best-Performing Models We compare training on our best template-based data with state-of-the-art in Table 6.4. SQuAD F1 results reflect results on the hidden SQuAD test set. We report single-model numbers; Lewis et al. (2019) report an ensemble method achieving 56.40 F1 and a best single model achieving 54.7 F1. We make use of the whole-word-masking version of BERT-large, although using the original BERT-large gives

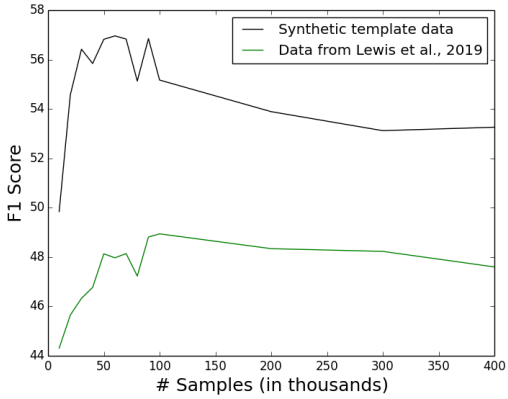

Figure 6.3: A comparison of the effect of the size of synthetic data on downstream QA performance.

Model Choice	SQuAD Test F1	SQuAD NER F1
BERT-large (ours)	**64.04**	**77.55**
BERT-large (Lewis et al., 2019)	56.40	64.50

Table 6.4: A comparison of top results using the BERT-large model.

a similar performance of 62.69 on the SQuAD dev set. We report numbers on the sample of SQuAD questions which are named entities, which we refer to as SQuAD-NER. The subset corresponding to the SQuAD development dataset has 4,338 samples, and may differ slightly from Lewis et al. (2019) due to differences in NER preprocessing. We also trained a fully-supervised model on the SQuAD training dataset with varying amounts of data and found our unsupervised performance equals the supervised performance trained on about 3,000 labeled examples.

6.4 Summary

in this chapter, we introduce a retrieval-based approach to unsupervised extractive question answering. A simple template-based approach achieves state-of-the-art results for unsupervised methods on the SQuAD dataset of 64.04 F1 and 77.55 F1 when the answer is a named entity. We analyze the effect of several components in our template-based approaches

through ablation studies. We aim to experiment with other datasets and other domains, incorporate our synthetic data in a semi-supervised setting, and test the feasibility of our framework in a multi-lingual setting.

Chapter 7

Multi-Answer Summarization

In the last chapter, we addressed a task for which data existed, although we assumed it did not exist in the desired domain. In this chapter, we find a gap in available data for the task of multi-answer abstractive answer summarization, the summarization of answers to queries in online forums, and propose a pipeline for automatic creation of such a dataset, analogous to the data created in the previous chapter. Community Question Answering (CQA) forums such as Stack Overflow and Yahoo! Answers contain a rich resource of answers to a wide range of questions. Each question thread can receive a large number of answers with different opinions. The goal of multi-answer summarization is to produce a summary that includes information from multiple source answers. One major obstacle for this task the absence of a dataset that can provide supervision for producing multi-answer summaries. This work introduces a novel dataset creation method to automatically create multi-answer, bullet-point abstractive summaries from an existing CQA forum. Supervision provided by this dataset trains models to inherently produce multi-answer summaries. Additionally, to train models to output more diverse, faithful answer summaries, we propose a multi-reward optimization technique coupled with a sentence-relevance prediction multi-task loss. Our methods demonstrate improved coverage of input answers and faithfulness as measured by automatic and human evaluations compared to a strong baseline.

7.1 Introduction

In a world of information overload and the ubiquity of discussion forums, there is a need for text summarization as a means of distilling relevant information into a concise form. The problem is even more pertinent for question answering within the context of Comunity Question Answering (CQA) forums, where a person poses a question and can get an abundance of answers to sift through. Ideally, an answer summary should cover the multiple viewpoints found in the answers, where available. For example, in Table 7.1, a person poses a question about finding a puppy and also provides context on the type of dog. We present a sample of the 14 answers to that question on Yahoo! Answers and an automatically-created summary consisting of bullet points covering the answers' main ideas. We introduce a novel pipeline to build such a *multi-answer, bullet-point summarization dataset* and introduce models to generate faithful, high-coverage summaries. Multi-answer refers to information present in the summary which is derived from multiple answers in the source.

Question: i found a puppy that is less then six weeks old an no mother around what should i feed it?
Context: it a pit puppy i think
Answer 1: Go to a vet and get some and a small feeding bottle.
Answer 2: get a baby bottle warm milk best thing is to call a pet shop
Answer 3: it needs a certain type of milk, dont feed it cows milk
Answer 4: call a vet and ask them. if you cannot do that then give them alot of water and a little balony a day, than go into dog food...
Summary Bullet Points:
1. call the vet and tell them how old you think it is and what should you feed it... **2.** the first thing you want to do if you plan to keep it is go to petsamrt or pet co and ask anyone that specializes on dogs and get the pup a baby bottle and feed it milk but not cow milk try powder milk with water. **3.** Try and find something soft to eat (as in a soft dog food). **4.** if it is not yet walking, then get a bottle

Table 7.1: An example bullet-point summary from our answer summarization dataset, illustrating the multiple viewpoints present in the summaries created through our pipeline, and a subset of the 14 user answers to which the target summary can be aligned.

To date, most CQA forums have a notion of a 'best answer,' which is either manually chosen by the person who asked the question or by a moderator or obtained via community

ratings. Work in this field typically makes use of this best answer as a proxy for summaries Tomasoni and Huang (2010); Chan et al. (2012); Pande et al. (2013); Wang et al. (2014); Song et al. (2017). However, the best answer only presents one person's viewpoint and rarely captures the variety of ideas discussed in the thread. We refer to a viewpoint when one answer contends with another answer or offers new information not found in the other answer. Datasets such as WikiHowQA (Deng et al., 2020), which consists of a question, a long answer, and an answer summary, focus on answer selection and the summarization of a single answer. While CQASumm Chowdhury and Chakraborty (2019) uses the chosen best answer as the answer summary, they also apply heuristics to ensure token overlap with the remaining answers. However, we found that the heuristics applied generally promotes only long answers instead of multiple viewpoints. To validate our hypothesis, we examine a set of 30 summaries from CQASumm and found that only 37% of the examples contained information from viewpoints in multiple answers.

Although multi-answer summarization is an important research topic with practical applications, there are no relevant datasets or techniques to address it effectively. This chapter tries to close this gap by developing a dataset together with several modeling techniques for multi-answer summarization. To generate a multi-answer summarization dataset, we devise a pipeline to produce *bullet point answer summaries*. First, we select and cluster salient answer sentences. Then, we use the cluster centroids as our summary bullet points and remove them from the input to promote a more challenging, more abstractive task. We further filter the data to improve our dataset's quality and promote desirable summary characteristics such as compression. We find that a strong baseline model trained on our data inherently outputs multi-answer summaries. We focus our modeling efforts on generating content implied by the input text and being faithful to the underlying answers by covering multiple viewpoints. To this end, we use a reinforcement learning (RL) framework with new rewards and a sentence-relevance multi-task loss, whereby the model learns to predict relevant sentences for the current decoding step to more closely align the source and

generated output. Our models improve the coverage and faithfulness of generated summaries when compared to a state-of-the-art abstractive baseline.

The main contribution of this chapter is to develop, for the first time, a method for multi-answer abstractive summarization. To achieve this, 1) We introduce a dataset generation pipeline for answer summarization that goes beyond the best-answer summary, to create multi-answer, bullet-point summaries for training and evaluation 2) We introduce and evaluate RL reward functions on answer summarization, including entailment as a measure of faithfulness and volume of semantic space as a way to increase coverage of multiple answer viewpoints 3) We introduce a sentence-relevance prediction loss to increase the faithfulness and interpretability of the generated answer summaries. We will make our code available for reproducing our dataset pipeline and model results.

7.2 Related Work

Extractive Answer Summarization Much work has focused on the extractive summarization setting as an answer-ranking problem (Chan et al., 2012; Pande et al., 2013; Wang et al., 2014). Liu et al. (2008) find that only 48% of the best answers on Yahoo! Answers are unique best answers; there are multiple correct ways to answer a question. Other recent work has focused on sentence extraction using metadata (Tomasoni and Huang, 2010) or sparse-coding frameworks Song et al. (2017). Our focus is on representing viewpoints from multiple answers in an abstractive summarization framework.

Abstractive Answer Summarization Another line of work has attempted abstractive answer summarization by treating the tagged best answer as the gold summary of all the other answers (Chowdhury and Chakraborty, 2019; Chowdhury et al., 2020). Chowdhury and Chakraborty (2019) present CQASumm, a dataset of about 100k examples consisting of the best answer as the gold summary, which, however, often only contains one viewpoint.

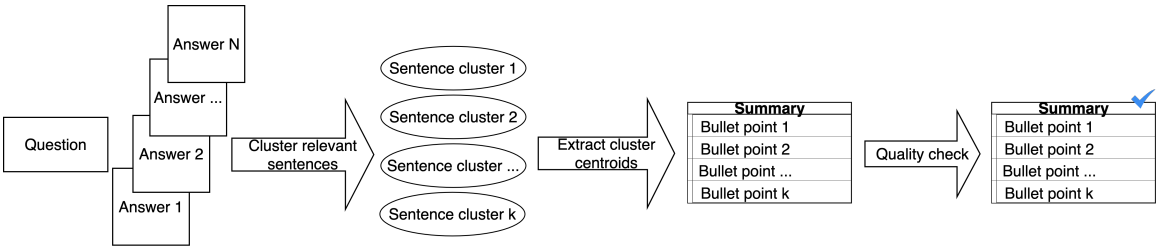

Figure 7.1: An illustration of our dataset pipeline. Given a question and answers, we cluster relevant sentences and remove the cluster centroid of non-singleton clusters from the input to use as bullet point summaries, filtering the example if it does not meet quality-control criteria.

RL and Multi-task Learning for Summarization Paulus et al. (2018) first apply the REINFORCE algorithm Williams (1992a) in the context of summarization. RL has since been applied for both extractive Narayan et al. (2018b); Dong et al. (2018), abstractive Pasunuru and Bansal (2018); Li et al. (2018); Huang et al. (2020); Laban et al. (2020) and hybrid approaches Chen and Bansal (2018). Böhm et al. (2019) stress the role of using rewards that correlate well with human judgments on downstream performance. This chapter focuses on the selection of rewards applicable for promoting faithful and diverse, abstractive answer summaries. Previous work on entailment as an RL reward has focused on document-level entailment in the news domain (Li et al., 2018; Pasunuru and Bansal, 2018). In this work, we show the effect of the choice of entailment model on downstream faithfulness prediction and the importance of using sentence-level entailment. Recent work in multi-task learning with summarization consists of sharing parameters between an abstractive generator and auxiliary tasks such as entailment and question generation (Guo et al., 2018) and text classification and syntax-labeling tasks (Lu et al., 2019).

7.3 Dataset Creation

Previous CQA work lacks multi-answer supervision. To address this research gap, we develop a system to create summaries covering multiple viewpoints of answers to a given question.

Overview of Data Generation Pipeline The input to our pipeline is a question and its answers. We use question threads from the Yahoo! Answers L6 corpus[11]. Our pipeline operates on the sentence level of these answers versus the answer level, as we believe that this granularity allows us to capture additional viewpoints. Our dataset pipeline consists of the following components: 1) a relevance model to remove irrelevant inputs, 2) a clustering model to cluster similar content, and 3) input and summary creation from centroids.

Relevance model We first aim to determine whether a given sentence is relevant to answering a question and, therefore, to be considered as a potential summary sentence. We use the ANTIQUE (Hashemi et al., 2020) relevance data for training a query-sentence relevance model. The data consists of Yahoo! answers and relevance labels on a scale from 1-4, with 1-2 not relevant and 3-4 relevant. We use a RoBERTa-based Liu et al. (2019) model fine-tuned on answer selection on the TREC-QA dataset (Wang et al., 2007) as a binary relevant/non-relevant classifier and further fine-tune it using the Tanda (Garg et al., 2020) method. We experimented with training the relevance classifier using Yahoo! Answers, treating the best answer as relevant and the other answers as not relevant, and analogously on the sentence level, although without improvements. The performance was measured using mean reciprocal rank on a held-out relevance set.

As input to the clustering stage, we remove sentences that our relevance model labels as irrelevant (our model tends to over-predict relevant sentences, as many answers contain relevant sentences, thus removing only 16% of sentences). Improving this relevance classifier to better filter irrelevant answer sentences is a very interesting research direction, although we leave this for future work.

Clustering Most methods for short-text clustering (Hadifar et al., 2019; Xu et al., 2017) require a known value of k, the number of clusters, which is dynamic from question to

11. https://webscope.sandbox.yahoo.com/catalog.php?datatype=l&did=11

question in our setting. In this work, we use the sentence-transformers library (Reimers and Gurevych, 2019a) to perform clustering. Specifically, we start with a RoBERTa-based model fine-tuned for sentence embeddings on an entailment dataset, which is further fine-tuned for semantic similarity. Clustering parameters were chosen based on a StackOverflow clustering dataset containing labeled clusters commonly used in short-text clustering. We used Agglomerative clustering with average linkage, cosine distance, and a maximum distance of .65.

To create the final summaries, we locate the centroid of clusters with at least two sentences and use these centroids as bullet-point summaries. Further, we remove the centroid sentences from the sentence-tokenized input answers to create a challenging abstractive summarization dataset analogous to the XSum dataset (Narayan et al., 2018a). Since each cluster contains at least two sentences, we assume that given a perfect clustering algorithm, a related sentence can help generate the removed centroid sentence. While removing sentences naturally decreases coherence, we believe that this introduces a tolerable level of noise, considering the existing presence of noise through ungrammatical and occasionally incoherent answers. To further account for imperfections in the pipeline, we apply additional filtering techniques, described below.

Postprocessing and Quantitative Analysis We obtained question threads from Yahoo! Answers and applied heuristics detailed in Tomasoni and Huang (2010) to find threads suitable for summarization. Threads were removed if 1) there were less than five answers, 2) the longest answer was over 400 words, 3) the sum of the length of all answers was outside of (100, 1000) words, and 4) the average length of answers was outside of the (50, 300) words interval. This filtering left us with about 350k of the approximately 4.4 million threads and included both factoid and non-factoid questions. Questions include the subject of the post as well as the content of the post when available.

Dataset	% Novel unigrams	Oracle Extractive	Length
AnswerSumm (ours)	32.2	40.02/11.16/33.70	67
XSUM	35.8	29.79/8.81/22.65	23
CNN	16.8	50.38/28.55/46.58	46
DailyMail	17.0	55.23/30.55/51.24	55

Table 7.2: Comparison between AnwerSumm and the XSum Narayan et al. (2018a) and CNNDM Nallapati et al. (2016) datasets. Oracle Extractive and Length refer to the maximum ROUGE Lin (2004a) score achievable by an extractive model, and the average length of the summaries, respectively.

Example Filtering We remove examples from the dataset based on desired summarization characteristics. A desirable trait in summarization datasets is compression, i.e., the ratio of the input size to the summary size (Grusky et al., 2018). We remove examples with a compression ratio under 4, examples for which the input length exceeded 1,100 tokens and for which the summary length exceeded 250 tokens, leaving us with 284,295 examples. We further remove target summaries labeled as contradictions from a RoBERTa-based entailment model following Matsumaru et al. (2020). Furthermore, we remove examples with more than 10 "+" or "=" signs (math queries), those with very long (>50 characters) tokens, and those with a link in the target or more than one link in the source. Finally, we filter to ensure that we have examples where the named entities found in the target are also found in the source document.

Quality Analysis The filtering process yielded 96,701 examples, which we split into 88,512/4,032/4,157 training, validation, and testing examples. We annotated a subset of 400 summaries created by our pipeline to conduct quality checks. For each summary, the annotator reads the question, and if the answer coverage of the summary was determined as reasonable, the summary was marked as 1, otherwise 0. 370 of the 400 summaries were labeled as 1, showing that the pipeline creates largely relevant content. Additionally, on examining 30 summaries, we found that 80% contained multiple viewpoints versus the 37% we found in CQASumm, showing the benefit of our dataset pipeline in encoding multiple viewpoints. To further analyze the types of questions present in our dataset, we trained a

factoid/non-factoid question classifier using SQuAD Rajpurkar et al. (2016) data as factoid examples and non-factoid Yahoo! Questions dataset[12] as non-factoid examples. 8% of threads were labeled as factoid questions; the filtering steps based on answer size likely filter out examples with short, factoid answers.

Relation to Existing Datasets CQASumm is the closest dataset with our desired answer summarization qualities, although it simply promotes answers as summaries rather than truly summarizing answers. As discussed above, this dataset lacks our desired multi-answer summaries. A similar approach to dataset creation was taken by Shapira and Levy (2020) for review summarization by clustering reviews using pivot clustering, adding reviews to a cluster based on lexical overlap until a max length and min number of review requirements are met. There are notable differences to our approach in terms of granularity (reviews vs. sentence clustering), type of clustering (lexical vs. embedding-based), as well as the ultimate use of these clusters (they train a cluster summarizer while we combine cluster centroids for creating an abstractive bullet point combined with other cluster centroids). We present a comparison of dataset statistics between our dataset, which we call **AnswerSumm**, and the standard XSum and CNNDM Nallapati et al. (2016) summarization datasets in Table 7.2. In general, we find our dataset to be more abstractive than CNNDM and less so than XSum. We also note that our generated dataset is similar to CNNDM in that it consists of bullet points. While this may create summaries with less coherence, or potentially contradictory answers, we focus on producing multi-answer summaries in this work and leave improved summary coherence for future work.

12. `https://ciir.cs.umass.edu/downloads/nfL6/`

7.4 Modeling Multi-Answer Summarization

We build upon a standard sequence-to-sequence framework, making use of the pretrained BART (Lewis et al., 2020) model. The input to the model is the question concatenated with input answers. Fine-tuning such a model with cross-entropy loss alone, however, suffers from exposure bias and also does not directly optimize the evaluation metrics such as NLI and ROUGE-L Ranzato et al. (2016). The REINFORCE algorithm Williams (1992a), on the other hand, allows for optimizing the evaluation metrics using non-differentiable rewards. Therefore, we use an RL multi-reward objective in addition to standard cross-entropy loss to promote summaries with both high coverage of the input answers and faithfulness. Additionally, we also introduce an auxiliary loss function for more interpretable and faithful summaries.

Multi-Reward Optimization We follow the settings of Pasunuru and Bansal (2018) for optimizing multiple rewards. Recall the settings from Chapter 2, where $x = \{x_1, x_2, \ldots, x_{n'}\}$ refers to the input source tokens (e.g. a question and its answers), and $y = \{y_1, y_2, \ldots, y_M\}$ refers to the gold target summary which consists of $\{y_{1_s}, y_{s_s}, \ldots, y_{M_s}\}$ sentences. Standard training minimizes the negative log-likelihood (NLL) loss using teacher forcing Williams and Zipser (1989):

$$L_{sup}(x, y) = -\sum_{t=1}^{m} \log(f(y_t|y_{0:t-1}, x)) \tag{7.1}$$

For our RL optimization, we use self-critical policy gradient training as in Paulus et al. (2018); Rennie et al. (2017). At each time-step, we produce an output y^s by sampling from the current decoding probability, $p(y_t^s|y_1^s, ..., y_{t-1}^s, x)$, as well as an output $\hat{y}$ obtained by greedily decoding from the current probability distribution. We define a reward function $r(y^o, x, y) \in [0, 1]$, i.e., the reward function compares y^o (i.e., either $\hat{y}$ or y^s) with x and y.

The RL loss function $L_{rl}(x, y) =$:

$$(r(\hat{y}, x, y) - r(y^s, x, y)) \sum_{t=1}^{m} \log p(y_t^s | y_1^s, ..., y_{t-1}^s, x) \tag{7.2}$$

As in Paulus et al. (2018) and Pasunuru and Bansal (2018), we use a mixture of the above two losses:

$$L_{mixed} = \gamma_{rl} L_{rl} + \gamma_{sup} L_{sup}, \tag{7.3}$$

where γ_{rl} and γ_{sup} are tunable hyperparameters used as scaling factors. Rather than applying weights to each reward, we follow Pasunuru and Bansal (2018) and optimize L_{mixed} by alternating rewards in each minibatch.

Rewards We now describe the three RL reward functions used: (1) ROUGE Lin (2004a) as a proxy for content coverage, (2) entailment (NLI) for faithfulness, and (3) semantic area to measure the coverage of a summary in a semantic space.

ROUGE Lin (2004a): Similar to Paulus et al. (2018) and Pasunuru and Bansal (2018), we use ROUGE-L as a reward to additionally promote important content beyond the cross-entropy loss.

Natural Language Inference (NLI) for Faithful Summarization : We use the degree of entailment of summaries given input answers as a reward to promote faithfulness of answer summarization. While entailment has been used as a reward as well as a summarization metric, we find several gaps in the current literature. Firstly, a discussion of the effect of the quality of the NLI evaluation model on downstream faithfulness metrics is incomplete. Also, summarization work typically uses NLI models with document-level input, while NLI models are generally trained on sentence-level data.

Method	% Correct
BERT NLI (Falke et al. (2019))	64.1%
FactCC Kryscinski et al. (2020b)	70.0%
QAGs Wang et al. (2020)	72.1%
BART MNLI (sentence)	71.9%
RoBERTa MNLI (sentence)	89.8%
RoBERTa MNLI (article)	47.7%
RoBERTa MNLI (max article sentence)	85.0%

Table 7.3: Results from faithfulness ranking evaluation from Falke et al. (2019), showing the importance, both of the strength of the NLI model on downstream faithfulness performance,and the effect of input granularity on performance. Sentence and article in parentheses indicate the granularity of the source input to the NLI model; max sentence calculates the max score over all article sentences as the score of a given target sentence.

Falke et al. (2019) analyze NLI models for ranking summaries; given an input sentence and two summary sentences, one faithful and one unfaithful to the input, a model should rank the faithful summary higher than an unfaithful summary. They introduce a dataset of 377 examples and measure the rank accuracy of NLI models. They define NLI as a measure of faithfulness for ranking summaries in the following way: Let $\mathcal{N}$ be an NLI model which, given a claim c and a premise p, computes $\mathcal{N}(p, c)$, the probability that the claim is entailed by the premise. We use this to calculate the NLI score for a summary y^s consisting of N_s sentences:

$$\text{NLI}(y^s, x) = \frac{1}{M_s} \sum_{i=1}^{M_S} \max_{s \in x} \mathcal{N}(s, y_{i_s}^s) \tag{7.4}$$

For the original task introduced in Falke et al. (2019), x consists of a single source sentence from the CNNDM corpus. We present our findings on this task in Table 7.3. We examine how the quality of the NLI model affects performance by comparing BART Lewis et al. (2020) and RoBERTa fine-tuned on the MNLI corpus Williams et al. (2018). Although the performance gap of these two models is very small on MNLI (90.2% for RoBERTa and 89.9% for BART), the performance gap is very large on ranking these sentences (89.8% for RoBERTa and 71.9% for BART), perhaps due to more subtle model differences not detected in the MNLI dataset.

We also address the effect of the granularity of the NLI model input. As discussed above, Falke et al. (2019) perform ranking based on sentence-level input and output. Recent work in entailment as a summarization metric, however, uses the entire input document as input to the NLI model for faithfulness calculations (Maynez et al., 2020), rather than computing the max over all the input sentences as in Equation (7.4). We locate the full source articles for the 377 examples and perform two experiments, one using Equation (7.4), and the other which uses the entire article to score the target sentence, $\mathcal{N}(x, y_{i_s})$. Performance drops when using the entire article as the input versus using Equation (7.4). To ensure that the performance drop was not caused by content truncation due to the 512 input size limitation, we also experimented with using the article starting from the relevant source sentence, without improvements.

Furthermore, we find that the use of NLI is particularly suitable for AnswerSumm. We sampled six threads from our dataset. Then for each thread, we wrote sentences entailed by the source as well as sentences based on similar themes but not stated in the source, totaling 50 faithful and 50 hallucinated examples. We find that the RoBERTa MNLI model can correctly identify these examples with 96% accuracy. We believe that NLI is intuitively more suitable for our data, which is less entity-heavy when compared to the news domain.

Semantic Area for Multi-Answer Summarization We aim to reward summaries that include information from more of the answers found in the input answers. To achieve diverse extractive summarization, Yogatama et al. (2015) embed sentences in semantic space and select those whose convex hull maximizes the volume in that space. This idea of semantic volume is also used to measure the semantic overlap between summaries and references in Jung et al. (2019). We use semantic volume as a proxy for covering multiple viewpoints; the summary with the larger semantic volume covers a wider range of views discussed in the input. We make use of sentence-transformers Reimers and Gurevych (2019b) to obtain sentence embeddings for each sentence. We project each embedding onto two dimensions

using PCA, and thus, our volume calculation reduces to an area calculation, which we call **Semantic Area**. We use min-max normalization to keep the reward in the range of 0 to 1.

Relevant Sentence Prediction We want to more closely align the decoded summary with the source text, as hallucinations may be caused by the decoder acting more as a language model rather than attending to the source text Maynez et al. (2020). Aligning the source and generated output offers a potential interpretable output during inference, which goes beyond using attention for interpretation Wiegreffe and Pinter (2019). We introduce an auxiliary loss by which the model predicts, based on the decoder representation, a span of source text relevant to the current time-step, analogous to finding evidence to support a claim of factuality Kryscinski et al. (2020b).

Let $h_{e_i} \in R^{dim_e}$ be the representation of token x_i from the last layer of the encoder. Let $h_{d_i} \in R^{dim_d}$ be the representation of token y_i^* from the last layer of the decoder right before the softmax layer. Here, $dim_e = dim_d = 1024$. Let h_e be the concatenation of all h_{e_i} and h_d be the concatenation of all h_{d_i}. We then pass these representations through separate layers L_e and L_d which correspond to the typical layer used in BART classification tasks except that it outputs a representation of size 2048:

$$h_e^* = L_e(h_e),\ h_d^* = L_d(h_d) \tag{7.5}$$

We split the resulting representations in half along the hidden dimension, resulting in encoder representations $h^*_{e\text{-}start}, h^*_{e\text{-}end}$ and decoder representations $h^*_{d\text{-}start}, h^*_{d\text{-}end}$ which will be used for start and end relevant source span prediction. We then compute an inner product between these representations, resulting in logits over the input corresponding to potential start and end spans:

$$\begin{aligned} \text{logit}_{start} &= h^*_{e\text{-}start} \bullet h^{*T}_{d\text{-}start} \\ \text{logit}_{end} &= h^*_{e\text{-}end} \bullet h^{*T}_{d\text{-}end} \end{aligned} \tag{7.6}$$

Cross entropy loss can then be calculated over the start and end logits with reference to gold spans as in SQuAD question answering training. We call this loss L_{span}. Our final loss function becomes:

$$L_{mixed} = \gamma_{rl} L_{rl} + \gamma_{ml} L_{ml} + \gamma_{span} L_{span}, \tag{7.7}$$

where L_{span} is the cross-entropy loss over start and end span predictions. Specifically, we separate input sentences with special tokens and predict sentence-level spans, which amounts to predicting a start and end token corresponding to a relevant sentence, so we call this model variation **Sent Relevance**. For each sentence in the gold target training data, we calculate the BM25 scores of the sentences in the source to pick the gold relevant source sentence for that target sentence. All the timesteps corresponding to a target sentence use the same relevant input sentence. We also experimented with just predicting relevant source sentences at the end of each target, using a binary sentence classification loss and a regression loss over the BM25 scores, without large improvements.

7.5 Experimental Settings

We use the fairseq codebase Ott et al. (2019) for our experiments. Our base abstractive text summarization model is BART (Lewis et al., 2020), a pretrained denoising autoencoder that builds off of the sequence-to-sequence transformer of Vaswani et al. (2017). Input to the model is the question concatenated with input answers. We fine-tune BART using a polynomial decay learning rate scheduler with learning rate $3\mathrm{e}{-5}$, using the Adam optimizer (Kingma and Ba, 2015). We train with 500 warmup steps and 20,000 total steps and pick the model with the best label-smoothed cross-entropy Szegedy et al. (2016) validation loss. Cross-entropy loss is our main loss, while the RL rewards and sentence-relevance prediction can be viewed as auxiliary losses. In RL experiments, we train using BART from scratch, as opposed to using a model already fine-tuned on answer summarization, as we found that this model better learned to follow the given rewards. Following similar ratios as in Lu

Method	ROUGE-1/2/L
LexRank	26.86/5.17/22.68
TextRank	27.44/5.05/22.13
BertSum Liu and Lapata (2019c)	30.01/5.76/24.83

Table 7.4: ROUGE scores for baseline extractive models.

et al. (2019), we set (γ_{rl},γ_{ml},γ_{span}) = (0.9, 0.1, 0.0) when experimenting without sentence-relevance loss, (0.00, 1.0, 1.0) for experiments with just relevant sentence prediction and cross-entropy loss, and (0.9, 0.5, 0.01) for experiments with all losses. Hyperparameters were tuned on the validation set; we found a larger γ_{ml} necessary when combining rewards with sentence relevance prediction to ensure that the main negative log-likelihood loss was not drowned out by the auxiliary losses.

7.6 Results

Extractive Baselines We use standard extractive summarization baselines such as Lexrank Erkan and Radev (2004) and TextRank Mihalcea and Tarau (2004), and a BERT-based extractive model, BertSum Liu and Lapata (2019c). Results are presented in Table 7.4. We observe a large gap between these baselines and the extractive oracle, which is the upper bound for extractive model performance, showing potential for improvement. Since we focus on abstractive summarization, we leave improving extractive models for future work.

Abstractive Models We present the results of the abstractive models in Table 7.5. We note that while the model with ROUGE reward outperforms the baseline in ROUGE-L (the ROUGE variant optimized), we do not see larger gains in ROUGE due to the similarity between the ROUGE optimization and NLL on our datasets. For bullet-point summaries, minimizing the NLL is analogous to rephrasing relevant bullet-points from the source and increasing the ROUGE-L. The model that combines all the RL rewards achieves the highest ROUGE performance, while the model with all RL rewards and sentence-relevance loss

Method	ROUGE-1/2/L	NLI
BART baseline	**33.37/8.39**/29.41	48.13
BART + RL (ROUGE)	33.26/8.30/29.46	49.29
BART + RL (NLI)	33.05/8.36/29.23	**56.68**
BART + RL (Semantic Area)	33.33/8.28/**29.60**	51.14
BART + RL (ALL)	**33.54/8.41/29.65**	51.18
BART + Sent Relevance	32.95/8.33/29.38	52.31
BART + Sent Relevance + RL (ALL)	33.21/8.29/29.46	**56.99**

Table 7.5: ROUGE and NLI scores for proposed models, with the two highest scores for each metric highlighted

achieves the highest NLI score. The faithfulness of the model with only sentence relevance loss is further improved by adding the RL rewards. In general, we see that the model with a single RL reward achieves the highest score of the target summaries for that metric, i.e., the highest NLI score is achieved using only the NLI-based reward. Additionally, we calculate the average semantic area of the resulting summaries. The baseline model, the model with just semantic reward, and the final model with all rewards have semantic areas of 39.7, 46.5, and 42. To further show the effect of our dataset on multi-answer summarization, we train a BART model on the most related answer summarization dataset CQASumm and find that the semantic area of that model's summaries is 31.54. This result shows the importance of supervision from our dataset pipeline for ensuring coverage of multiple viewpoints in answer summarization.

As automatic metrics may not correlate perfectly with human judgments, we perform a human evaluation to determine the differences in model output qualities. We presented two annotators who are fluent in English with the question, answers, and summaries from three models and asked them to annotate the summaries along the following dimensions: 1) On a Likert scale from 1-5, label the ability to capture viewpoints from multiple answers, with points deducted for repetition 2) On a Likert scale from 1-5, label the extent of faithfulness of the summary, with 5 being a completely faithful summary and 1 being an entirely inaccurate summary.

Method	Multi-Answer	Faithful
BART	4.45	3.72
BART + RL (ALL)	4.57	4.13
BART + Sent Relevance + RL (ALL)	4.55	4.24

Table 7.6: Human evaluations of model outputs measuring the ability to capture information from multiple answers and faithfulness. Higher is better.

We present results in Table 7.6. Annotations are averaged between each annotator and then across examples for 50 questions threads from three models. We choose to compare the BART baseline, the BART model with all RL rewards, and the BART model with span prediction to determine the effect of our rewards and the multi-task loss on output quality. Pearson correlations for faithfulness and multi-answer scores among the annotators were 0.41 and 0.31, displaying moderate correlation. We find that most models can generate multiple viewpoint summaries. The baseline already generates multi-answer output, likely because the dataset pipeline produces summaries that contain multiple viewpoints, so the baseline learns to produce such output. Using a student's t-test with a p-value of 0.05, we find that the improvement in faithfulness between the RL models and the baseline is statistically significant while the other differences are not. With the span-based model, this improvement comes at the cost of some level of abstraction, as the percentage of novel unigrams found in the summary is 10% vs. 13% found in the baseline and RL-only models. This reduction in abstraction likely results because the span loss more closely binds the decoder representation with the encoder representation, encouraging the model to copy more from the source. We demonstrate the added advantage of our span prediction model's interpretability by using it to provide explanations for the generated summaries in the Appendix.

Sample Output We show the model-generated summaries for the model "BART + Sent Relevance + RL (ALL)" as well as the source sentences the model predicts as relevant at the end of each sentence generated during decoding. In the example below, the model can correctly abstract meaning from the source sentences and formulate summary bullet points. Occasionally the model will output a point that is not coherent by itself (e.g. 'It's a great

Question: What is the secret to work/life balance?

Summary Sentences:	**Associated Source Sentences:**
You have to find the right balance between work and life.	I mean you keep looking outside of work for happiness, and you want a balance, so why not be happy everywhere
If you don't try something new, you'll never know what you're doing.	If what you're doing now isn't working, why not try something new
You have to make them both equal.	Only then will they matter equally
It's a great book, and you can get it at any book store.	It's absolutely possible, and in my sources is a book that you can get as cheap as $1.62
I think the trick is to go to work with the right attitude.	It seems to me that people just go to work with the wrong attitude actually

Table 7.7: An example of the predicted sentences from our span-based model with all rewards. On the left side are the generated summary sentences and on the right side are the sentences predicted to be relevant at the end of sentence timestep during generation.

book') or may output related but not supported text. We believe this is due to the BM25 relevance function used for determining relevant sentences for training. Examining this mechanism sentence relevance prediction as a model probing task as well as improving coherence in summaries, to go beyond bullet point summaries via methods such as sentence fusion, are valuable areas of future work.

We show model outputs from the three models examined in human evaluation in Table 7.8. We see that the baseline hallucinates several times. We also notice how the hallucinations, as opposed to typical hallucinations in the news domain which may replace entities, are often plausible responses. For example, although the baseline generates an output saying that it is not a good idea to lose weight, which is not directly stated in the source, such an answer is very plausible. We also found that there was occasionally a fine line between what was a hallucination and what was a plausible generated text which is not entirely implied in the source. For example, the text stating it is not a good idea to lose weight echoes the sentiment that the user asking the question should make the choice for themselves, although this is not stated in this fashion. We believe that more precisely defining the degrees of hallucination and plausibility is an important direction for future work.

Question: average 14 year old girls weight?
Context: im 14 years old, 145 pounds, 5'5" and 1/2, need to know if i need to lose weight. my brother and his friend(who is also my friend) have told me i do, but i dont know. is this a normal weight?
Answer 1: be comfortable in your own body, don't worry what others think of you! If you feel like you need to lose weight then exercise and make that choice for yourself.
Answer 2: your bro and his friend are retards who cares what they say??? and if you think you weigh alot it probaly doesn't show
Answer 3: First off, that is not very nice for your brother and his friends to be telling you to loose weight ... It is actually more healthy to be about 10% over what is normal for your age. Remember that muscle weights more than fat but it takes up less room than fat does. If you eat well and exercise daily you have nothing to worry about.
Answer 4: i'm 5'8" and wrestled at 126 lbs when i was a freshman and sophomore.
Answer 5: Your BMI (Body Mass Index) is 24.1 Based on this number and your age you are not overweight, but are considered "at risk for overweight."
BART Baseline Summary: You are not overweight if you are eating healthy and exercising. $\langle S \rangle$ Your weight will change as you get older. $\langle S \rangle$ If you are overweight, it is because you have too much muscle and not enough fat. $\langle S \rangle$ I'm not sure how old you are, but I'm assuming you are 14. $\langle S \rangle$ It is not a good idea to lose weight.
BART + RL (ALL) Summary: If you feel like you need to lose weight then exercise and make that choice for yourself. $\langle S \rangle$ You are not overweight, and you are not at risk for overweight. $\langle S \rangle$ It is normal for a fourteen year old to be a little over weight, but not overweight. $\langle S \rangle$ If you are a wrestler, you will know how much muscle you have.
BART + Sent Relevance + RL (ALL) Summary: If you feel like you need to lose weight then do so, but don't listen to your brother and his friend. $\langle S \rangle$ You are not overweight, but you are at risk for being overweight. $\langle S \rangle$ You should be comfortable with your weight. $\langle S \rangle$ If you have muscle, you will be able to lose more weight than if you had fat.

Table 7.8: Example question and answers along with bullet-point answer summaries from three models. Possible hallucinations are shown in red.

7.7 Summary

In this chapter, we propose multi-answer summarization by introducing a pipeline for creating a suitable dataset for the task and by introducing models to promote high-coverage, faithful answer summaries, as seen in automatic and human evaluations. We aim to refine this pipeline for future work by improving the relevance and clustering components and applying them to new data sources. We plan to study the abstractiveness-faithfulness tradeoff further, explore additional rewards for improved summary coherence, and move beyond bullet point summaries by building on work in sentence fusion.

Chapter 8

Improving Zero and Few-Shot Abstractive Summarization with Intermediate Fine-tuning and Data Augmentation

As seen in the previous chapter, models trained on data with given characteristics produce summaries with similar characteristics, such as the inclusion of multiple perspectives. In this chapter, we make use of this characteristic for improving zero and few-shot application of models, a realistic setting when applying summarization to new, niche domains. In this chapter, we introduce a novel and generalizable method, called WikiTransfer, for fine-tuning pretrained models for summarization in an unsupervised, dataset-specific manner. WikiTransfer fine-tunes pretrained models on pseudo-summaries, produced from generic Wikipedia data, which contain characteristics of the target dataset, such as the length and level of abstraction of the desired summaries. WikiTransfer models achieve state-of-the-art, zero-shot abstractive summarization performance on the CNNDM dataset and demonstrate the effectiveness of our approach on three additional diverse datasets. These models are

more robust to noisy data and also achieve better or comparable few-shot performance using 10 and 100 training examples when compared to few-shot transfer from other summarization datasets. Additionally, to understand the role of dataset aspects in transfer performance and the quality of the resulting output summaries, we further study the effect of the components of our unsupervised fine-tuning data and analyze few-shot performance along with data augmentation techniques using both automatic and human evaluation.

8.1 Introduction

Creating data for every new domain, however, is infeasible and highly costly. Thus, the ability to transfer large pretrained models to new domains with little or no in-domain data is necessary, especially as such models make their way into real-world applications.

Unsupervised summarization approaches include autoencoders to mirror the information compression inherent in summarization (Baziotis et al., 2019; Chu and Liu, 2019a; Bražinskas et al., 2020b) as well as large-scale pretraining for domain-specific adaptation (Yang et al., 2020). In domain adaptation for summarization, Wang et al. (2019) examine domain adaptation for extractive summarization and Hua and Wang (2017) showed that summarization models have difficulty generating text in the style of the target domain, while more recently, Zhang et al. (2019) report strong performance of pretrained models when trained in few-shot settings. Bražinskas et al. (2020a) fine-tune dataset-specific components of a model for few-shot learning. We aim to build on recent work in pretrained models and improve zero-shot and few-shot summarization by encoding characteristics of the target summarization dataset in unsupervised, intermediate fine-tuning data.

In one view, summarization can be seen as a function of several sub-functions of the input document, called subaspects, which determine the output form. Jung et al. (2019) define three subaspects for summarization: position, importance, and diversity, and study how these subaspects manifest themselves in summarization corpora and model outputs. For

example, a common subaspect for the CNNDM dataset is position; earlier sentences tend to constitute a good summary. Inspired by this view of summarization as subaspects, we aim to encode subaspects of a target dataset into unlabeled data to allow a model fine-tuned on this data to learn characteristics of the target dataset to improve zero-shot and few-shot transfer of the model. In our work, we focus on the subaspects of *extractive diversity*, as determined by extractive the summaries of the target dataset are with respect to the source input, *compression ratio* between the source document and summary, and, in the case of CNNDM, the *lead bias*. We assume knowledge of the target dataset such as the size of input documents, the size of the desired summaries, and the extent to which the summary is abstractive, all of that can be treated as prior knowledge if the task is to be well-defined (Kryscinski et al., 2020a). We encode this knowledge into Wikipedia article data by extracting summaries of the desired output length and filtering examples based on the desired level of abstraction.

Our contributions are the following: 1) We introduce a novel method, called WikiTransfer, which creates pseudo-summaries with subaspects of the target dataset, which can be used as unlabeled data for intermediate fine-tuning. We show that this method improves zero-shot domain transfer over transfer from other domains, achieving state-of-the-art unsupervised abstractive summarization performance on the CNNDM dataset while generalizing to other domains, and we perform extensive hyperparameter studies on the factors influencing zero-shot performance 2) We show robustness and additional improvements in transferring WikiTransfer models in the few-shot settings and analyze differences in performance when using data augmentation techniques across datasets.

8.2 Related Work

While advances have been made in neural techniques for summarization due in part to large datasets, less work has focused on domain adaptation of such methods in the zero

and few-shot settings. Wang et al. (2019) examine domain adaptation, but in extractive summarization. Hua and Wang (2017) examine domain adaptation between opinion and news summarization, observing that models trained on one domain and applied to another domain can capture relevant content but differ in style in generating the summary.

Bražinskas et al. (2020a) introduce plug-in networks, small finetune-able layers that aim to reproduce characteristics of the target dataset as seen in a small set of labeled examples. In contrast, we aim to encode the characteristics of our target dataset, such as level of extraction and compression, a priori in the intermediate training phase for better adaptation. In other work, Lebanoff et al. (2018) adapt a single-document summarization model to multi-document settings, while Zhu et al. (2019) use Wikipedia reference data for downstream query-based summarization

Several approaches for unsupervised summarization have made use of variational autoencoders (Baziotis et al., 2019; Chu and Liu, 2019a; Bražinskas et al., 2020b). Zhou and Rush (2019) makes use of pretrained language models for unsupervised text summarization by aligning the coverage of the generated summary to the source document. Laban et al. (2020) train an unsupervised summarization model with reinforcement learning rewards. In another line of work, extractive models such as TextRank, (Mihalcea and Tarau, 2004), LexRank (Erkan and Radev, 2004), and more recently PacSum (Zheng and Lapata, 2019), make use of graph centrality for modeling salience.

The power of pretrained models for few-shot transfer was shown for abstractive summarization in Zhang et al. (2019) and extractive summarization in Desai et al. (2020). Our work focuses on the zero-shot abstractive summarization setting as well as the transferability of models fine-tuned on task-specific data from a generic corpus, rather than just the transferability of a single pretrained model. The closest work to ours for zero-shot transfer is Yang et al. (2020), which makes use of the lead bias in news to pretrain an unsupervised model on a large dataset of news articles. Our approach, however, focuses on fine-tuning an already-pretrained model specifically for summarization on a downstream dataset by

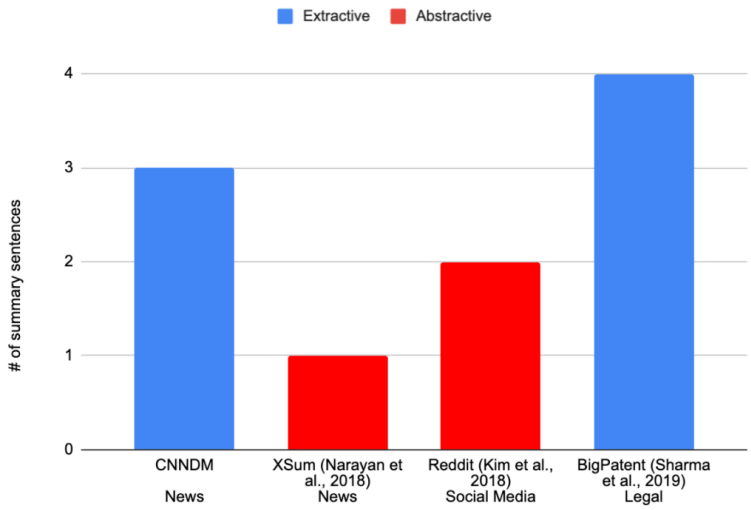

Figure 8.1: Characteristics of common summarization datasets, motivating our predefined specification of summary characteristics such as compression ratio and level of extraction.

leveraging a generic text corpus (Wikipedia) to create auxiliary fine-tuning data that transfer across domains, allowing for more fine-grained control over the transfer process. We show the generalizability of such fine-tuning across domains. BART (Lewis et al., 2020) is a pretrained denoising autoencoder and achieved state-of-the-art performance when fine-tuned on summarization tasks at the time. In this work, we use BART as our base pretrained model but in future work will experiment with other pretrained models.

8.3 WikiTransfer Zero and Few-shot Summarization

Dataset Characteristics Analysis We examine characteristics of four commonly-used summarization datasets in Figure 8.1. As seen in that figure, datasets differ in terms of the number of summary sentences, the level of abstraction, and the domain. Furthermore, such differences exist even within the same domain, as seen in the CNNDM and XSum datasets, both of which consist of news articles. Since characteristics such as domain do not determine the output form of the summary, these characteristics are better specified a priori so that the summarization problem is not underconstrained (Kryscinski et al., 2019).

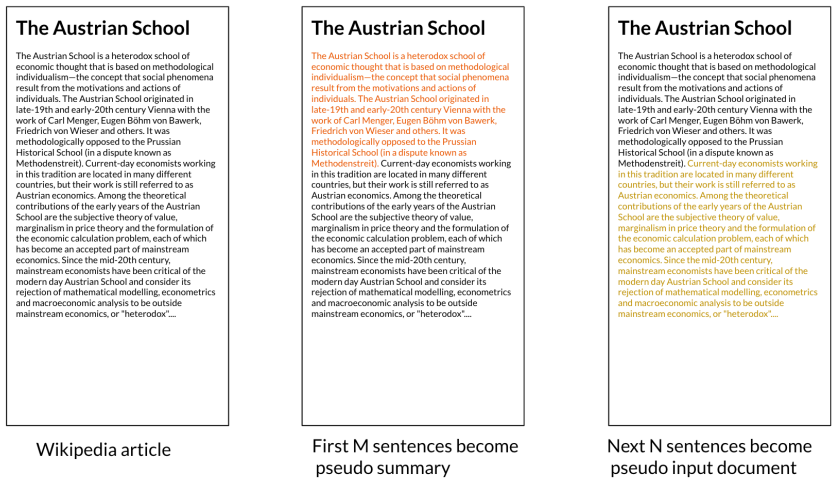

Figure 8.2: Dataset-specific WikiTransfer data is created by selecting the first M sentences from a Wikipedia article as the summary and the next N sentences as the source, where M and N are specified by the target dataset.

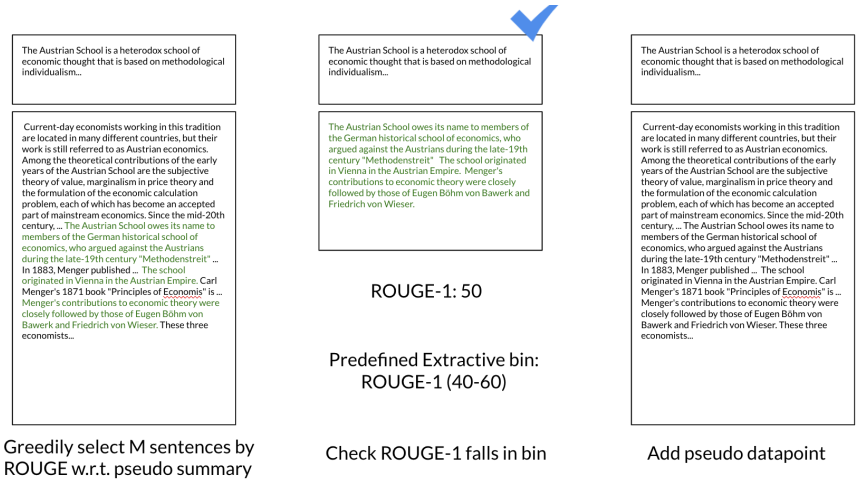

Figure 8.3: In order to filter datapoints based on the level of abstraction of the target dataset, a greedy extractive ROUGE score is calculated between the WikiTransfer source and summary and then compared to the pre-defined target dataset level of extraction. The predefined ROUGE-1 (40-60) bin corresponds to the very extractive CNNDM dataset.

WikiTransfer Intermediate Fine-tuning We propose a method for fine-tuning pretrained models using unsupervised Wikipedia data. We create dataset-specific unsupervised data for this intermediate fine-tuning, by making use of characteristics of the target dataset such as the average length of input documents, the average summary length, and the general bin of whether the summaries desired are very abstractive or very extractive, as discussed

above. Assume that we want a summary of M sentences from source documents of N sentences on average, and that we know approximately how extractive the summaries are in the target dataset, as defined as the upper bound ROUGE (Lin, 2004a) performance of an extractive model, the extractive oracle, on that dataset. We bin the level of extraction of the target summaries into extremely abstractive (ROUGE oracle 10-30), more abstractive (ROUGE oracle 20-30), more extractive (ROUGE oracle 30-50), and extremely extractive (ROUGE oracle 40-60). We then iterate the following procedure on all Wikipedia articles available in a Wikipedia dump: We remove the first M sentences from the Wikipedia article for use as a summary and the following N sentences for use as a source document, as seen in Figure 8.2. Then, we want to check whether this pseudo data point matches the level of extraction of the target dataset, as seen in Figure 8.3. We select the M sentences in the pseudo source document with the highest individual ROUGE scores against the pseudo summary and calculate the ROUGE score between those M sentences concatenated and the pseudo summary, which amounts to a greedy upper bound of the performance of an extractive model on this example. The example will be kept if this ROUGE score falls into the general range of the extractive oracle of the target dataset defined previously and otherwise discarded. We use knowledge of how abstractive a dataset is as a type of summary style which an end-user would know ahead of time. We filter the data points from Wikipedia so that only those which fall into the bin for a given dataset are used for fine-tuning. For datasets that are extremely abstractive, such examples may be hard to find, so we remove high-ROUGE sentences from the input until the desired ROUGE oracle score is reached. From here on we refer to data created through this process as **WikiTransfer**. We then fine-tune a pretrained model on this dataset-specific WikiTransfer data to transfer to a target domain.

Data Augmentation via Round-Trip Translation In addition to fine-tuning on WikiTransfer data for zero-shot domain transfer, we test the ability of our model to transfer when

we have few examples and whether data augmentation further improves these results. In few-shot fine-tuning, we conduct data augmentation to reduce brute-force memorization and introduce a regularization effect. Specifically, we perform round-trip translation (Yu et al., 2018) to generate paraphrases of both the source documents and summaries, as previous work has found this approach creates diverse paraphrases for augmentation while preserving semantic meaning (Yu et al., 2018; Xie et al., 2019). Our examination found that round-trip translation increased the number of novel n-grams while preserving semantic meaning. Given a dataset of N data points, we translate the source and target sentence-wise into a non-English language and keep the top k beam hypotheses from beam search as output. We then do likewise for the backtranslation to English. This results in $N * k^2$ augmented data points in addition to the N original supervised data points. We align a single beam from the translation to non-English text to a single beam in the backtranslation to English; using all combinations of beams for augmented data did not result in an improvement in initial experiments.

8.4 Experimental Settings

Datasets We experiment with four datasets, CNNDM, XSum (Narayan et al., 2018a), Reddit_tifu (Reddit) (Kim et al., 2019), and BigPatent (Sharma et al., 2019b). The datasets were chosen as they all differ in their abstractiveness, output length (from one sentence in XSum to on average four in BigPatent), and cover multiple domains from news (CNNDM and XSum) to social media (Reddit) to patent documents (BigPatent), to show the generalizability of our results. Each of the datasets falls into a different extractive bin, from the most extractive CNNDM to the more abstractive XSum.

We use the statistics from the original papers to determine the extractive bin of the dataset except for the case of Reddit; upon seeing the strong zero-shot performance of the CNNDM, we investigated the extractive oracle of the Reddit dataset and found it to be much

higher (about 31 ROUGE-1) than that stated in the original paper. We select the first M sentences for the pseudo-summaries from Wikipedia except in the case of Reddit, where we choose the IND-ORIG setting from Zhang et al. (2019). In this formulation, sentences are scored independently and the original implementation of ROUGE is calculated. This did not result in a difference in zero-shot performance, but upon a qualitative inspection of the output, we found the IND-ORIG to be less biased towards Wikipedia style with the coherence of the summaries not being an issue.

Model Selection and Metric For the experiments which follow, we first choose the model with the best zero-shot performance on a given domain. We test the zero-shot performance from all four domains onto every other domain. For models from our WikiTransfer subset, we choose the best model based on performance on an unsupervised WikiTransfer validation subset. We find that fine-tuning the model longer does not result in performance gains in few-shot transfer, and the checkpoints chosen were typically fine-tuned from 2 to 5 epochs. Results from hyperparameter studies for zero-shot transfer from WikiTransfer data are shown on the validation set of that given target dataset. Unless otherwise stated, all results reported are ROUGE-1/2/L. We run all few-shot transfer experiments on five subsets of supervised data, and the reported numbers, unless zero-shot, are the average of the top three results of the five runs. The 10 data point sets are subsets of the 100 data point sets.

We believe that the approximate level of extraction of desired summaries should be treated as prior knowledge. We also examine, however, how many data points are needed to accurately find the extractive oracle bin from target datasets. We found that using 10 data points sufficed to accurately estimate the bin of the extractive oracle.

Using the first M sentences does not produce ideal summaries of the remaining Wikipedia article, but experiments comparing the WikiTransfer approach on Wikipedia data as opposed to using in-domain data, as well as manual inspection of the data showed the validity of using Wikipedia data for proxy summaries. While the extractive oracle provides some

measure of overlap, this heuristic does not ensure deeper semantic overlap or faithfulness between the pseudo summary and the rest of the article. We believe a valuable direction for future work is improving the target-specific data as well as encoding additional semantics and style-based subaspects into the pseudo summaries.

Data Augmentation Parameters For data augmentation via round-trip translation, we use a beam size of 10 and k of 10 on German and Russian translation models; fairseq provides bidirectional pretrained translation models (Edunov et al., 2018) from WMT19 (Ng et al., 2019) for these language pairs. For both 10 and 100 data points, this resulted in 2010 and 20100 total data points. We call the model fine-tuned on these settings 10-aug and 100-aug. For consistency loss, we use the same augmented data.

Model Hyperparameters We use the fairseq codebase (Ott et al., 2019) for our experiments. Our base abstractive text summarization model is BART-large (Lewis et al., 2020), a pretrained denoising autoencoder with 405M parameters that builds off of the sequence-to-sequence transformer of Vaswani et al. (2017). We fine-tune BART using a polynomial decay learning rate scheduler using the Adam optimizer (Kingma and Ba, 2015). We mainly vary the learning-rate scheduler, warm-up updates, and total updates. As in the previous few-shot summarization work (Zhang et al., 2019) and work in unsupervised machine translation (Conneau and Lample, 2019), we use a subset of the target-domain validation set for early stopping based on the validation loss. We used the following learning rates, warmup updates and total parameters based on an examination of the validation curves in initial experiments: 10: (25, 100, 3e-5) 10-aug: (20, 200, 3e-5), 100 (20, 200, 3e-5), 100-aug: (200, 1000, 1e-5). For consistency loss experiments, we use the λ value of .5 for experiments with 100 data points and λ of 0.1 for experiments with 10 data points. Higher λ values with more data points follows intuition that with more data points the model naturally learns to distinguish between noisy and original output and is thus less sensitive to instabilities introduced in the auxiliary loss.

We use the standard training and testing splits of each dataset (for Reddit, we use the same 80-10-10% split as in Zhang et al. (2019)), and thus refer the reader to the original papers for detailed statistics. For validation, we used a subset of the target-dataset validation set consisting of 4k examples. While this matches previous unsupervised and transfer settings, we understand that the use of a large validation set is not ideal. We experimented with smaller validation sets on Reddit transfer and found that the results did not change using a validation set of only 10 data points, although we leave a further examination of the effect of validation set size for future work.

We provide the range of the label-smoothed cross-entropy validation loss by taking the average validation loss (over five subsets) from the best-performing and worst-performing transfer models on a given dataset. The range of validation losses for CNNDM is (4.49, 5.05), for XSum (4.63, 5.45), for Reddit (5.98, 6.65), and for BigPatent (4.88, 6.40).

We found that full-precision floating-point gave slightly better, and more stable, results, so we report full-precision floating-point numbers. We set a maximum tokens-per-batch of 1024 and use gradient accumulation with an update frequency of 8 for all experiments with 10 data points, and 32 for 10-aug as well as all experiments with 100 (+ augmented) data points. For CNNDM 10 examples, we found it necessary to use a smaller learning rate (3e-6) to avoid immediate overfitting. We perform validation after each model update, as the models typically converge in under 50 iterations. For the 100-aug setting, we begin validation checking after 50 iterations as the models typically converged around 100 iterations. We train with label-smoothed cross-entropy (Szegedy et al., 2016) loss for few-shot transfer. We found that models can be sensitive to the choice of hyperparameters in the few-shot settings, hence the averaging over 5 subsets to reduce variation.

For zero and few-shot transfer, we compare transfer from BART trained on WikiTransfer data to the best-transferring BART model trained on the datasets. The following numbers are ROUGE-1. Our application of BART on fully-supervised data achieves state-of-the-art performance on Reddit (32.74). We perform slightly worse on CNNDM (44.16 vs 45.94

from Dou et al. (2020)). Lower performance when compared to Pegasus-large (Zhang et al., 2019) on XSum (45.14 vs 47.21) and BigPatent (43.34 vs 53.63) is likely due to differences in capacity and training batch size, as our performance is comparable to Pegasus-base. Our approach is not model-specific to BART, so we leave the application of other models such as Pegasus to future work and do not focus on achieving state-of-the-art on the fully-supervised individual datasets.

We limit our primary few-shot experiments to 10 and 100 data points, as we are primarily interested in real-world few-shot applications where we likely do not have 1k data points. Initial experiments using 1k and 10k data points on CNNDM showed that WikiTransfer still outperforms transfer from other domains, although both remain below state-of-the-art performance. We leave a further examination of fine-tuning on larger training sets for future work.

8.5 Zero-shot Transfer Results

We compare the zero-shot performance of BART fine-tuned on WikiTransfer data to that of one transferred from other summarization datasets. We also show the effect of different choices for WikiTransfer fine-tuning data on CNNDM and XSum.

Zero-shot Transfer Comparison We aim to show that a model fine-tuned on WikiTransfer data has better zero-shot performance than models transferred from other summarization datasets. We fine-tune BART on WikiTransfer data for each of the four target datasets described above and also fine-tune a model on each of the fully-supervised datasets. We compare the zero-shot performance of transferring from WikiTransfer against the best zero-shot transfer performance from another dataset in Table 8.1. Zero-shot transfer from WikiTransfer notably outperforms transferring from other datasets on CNNDM, XSum, and BigPatent. On Reddit, we perform better on ROUGE-1 and comparably on ROUGE-2/L, which may be due to distinct writing style on Reddit data, as noted in Zhang et al. (2019).

Target Dataset	WikiTransfer	Other Transfer
CNNDM	**39.11/17.25/35.73**	36.81/14.18/32.62 (Reddit)
XSum	**31.85/10.44/23.75**	24.04/6.43/18.99 (Reddit)
Reddit	**21.47**/4.10/17.62	21.37/**4.14/17.76** (CNNDM)
BigPatent	**35.58/10.91/31.53**	33.57/9.34/25.76 (CNNDM)

Table 8.1: Comparison of ROUGE-1/2/L zero-shot transfer performance from dataset-specific WikiTransfer vs. transfer from another dataset. The dataset from which zero-shot transfer performed the best is in parentheses.

Model	ROUGE-1/2/L
WikiTransfer	**39.11/17.25/35.73**
TED (Yang et al., 2020)	38.73/16.84/35.40

Table 8.2: A comparison of our approach to the unsupervised pretraining of TED (Yang et al., 2020), showing the superior performance and generalizability of our approach versus the TED model, which focused specifically on the news domain.

We also experimented with training a model on data combined from multiple datasets for zero-shot transfer, but this does not report improved results, so for the experiments which follow we use the best performing single-domain transfer model.

We compare our model to the state-of-the-art unsupervised abstractive model on CNNDM in Table 8.2. We outperform the recently-introduced TED model (Yang et al., 2020) which was specifically motivated for the news domain. We believe the creation of task-specific data from a generic corpus such as Wikipedia allows for more control over the transfer process than relying on the autoencoder objective of TED, and more generalizable cross-domain results.

Effect of WikiTransfer Hyperparameters We study the effect the characteristics of our intermediate fine-tuning data have on downstream zero-shot performance on CNNDM and XSum to compare highly extractive and abstractive datasets.

Effect of learning rate in intermediate fine-tuning We examine the extent to which overfitting to the unsupervised WikiTransfer data occurs by examining the effect of the learning rate in intermediate fine-tuning on zero-shot transfer performance. We finetune the

models on the CNNDM and XSum WikiTransfer data respectively each with a maximum learning rate of 3e-6 and 3e-5. Results are shown in Table 8.3. Using a smaller learning rate in intermediate fine-tuning improves results on CNNDM, but not on XSum, likely due to the simple extractive and lead bias objectives which can easily overfit during fine-tuning. We see a similar trend with the effect of dataset size. For datasets other than CNNDM, we use a learning rate of 3e-5 in intermediate fine-tuning.

Effect of extractive oracle bin use and the choice of M We tested whether using the extractive bin to filter examples in the unsupervised data affected zero-shot transfer. For this experiment, we used the first M sentences from the Wikipedia article as the summary and the remaining N as the source, but do not filter examples according to how extractive they are. From Table 8.3, we see that the extractive bin has a very noticeable effect on transfer results for XSum and a moderate effect on CNNDM. This is to be expected, as the model otherwise is missing information about XSum's distinctive output style.

We examine how the choice of M affected performance. We set $M = 1$ for CNNDM and $M = 3$ for XSum and filtered examples in a similar way based on the extractive bin of the target dataset. We see that the choice of M has a large impact on CNNDM performance but no decrease on XSum. This result, combined with the effect of filtering examples based on the extractive bin, gives insight into the importance of the subaspect of abstractiveness over compression for XSum performance.

Effect of intermediate pretraining dataset size We examined the effect of the size of the WikiTransfer data on downstream performance. Results are shown in Table 8.4. We see a general increase with the addition of more data, although smaller increases after 100k data points and even a decrease in 250k on XSum, likely due to noise variation. The performance with 10k data points on CNNDM is already much closer to the best performance than the XSum case. We believe that this is due to the highly extractive nature of CNNDM, which is especially easy for a model such as BART to learn, as it is pretrained

Ablation	CNNDM	XSum
LR=3e-6	40.14/17.71/36.66	27.60/8.62/20.93
LR=3e-5	39.73/16.94/36.24	31.80/10.46/23.66
LR=3e-6, No-bin	39.11/16.98/35.66	22.78/5.66/17.16
LR=3e-6, bin, M=1	37.45/14.72/32.52	27.60/8.62/20.93
LR=3e-6, bin, M=3	40.14/17.71/36.66	27.98/9.59/23.11

Table 8.3: Hyperparameter studies on the effect of learning rate, the use of extractive bin for data filtering and the choice of M in intermediate fine-tuning on ROUGE-1/2/L performance on CNNDM and XSum validation sets.

Intermediate Dataset Size	CNNDM	XSum
10k	39.48/17.79/36.3	21.59/4.85/16.28
100k	39.92/17.65/36.5	31.52/10.86/23.94
250k	40.10/17.70/36.62	31.39/10.27/23.43
400k	40.14/17.71/36.66	31.80/10.46/23.66

Table 8.4: A comparison of the effect of dataset size of the unsupervised intermediate fine-tuning data on the zero-shot transfer ROUGE-1/2/L performance.

as a denoising autoencoder. For XSum, we see a noticeable improvement from 10k to 100k examples. We suspect that the abstractive objective is harder for the model to learn with small datasets. As we add more examples, we do not see a noticeable improvement. Such observations agree with our observation of the effect of learning rate and overfitting to the easier CNNDM objective. For the remaining experiments, we use 400k data points based on initial experiments.

Effect of summary sentence choice The first M sentences of a given Wikipedia article were chosen as this introduction intuitively form a coherent summary of the article. We examine the effect of choosing the first sentences compared to choosing based on other criteria. As an alternative, we pick the sentences with the highest self-ROUGE (ROUGE score of a sentence when using all other sentences as the reference summary) in a greedy fashion (the equivalent of the **IND-ORIG** settings in Zhang et al. (2019)). As in Zhang et al. (2019), we use ROUGE-1 F1. The sentences chosen under this heuristic consistently corresponded to those which were longest, and the resulting summaries were hence longer. Thus, we also experimented with choosing important sentences by using ROUGE-1 Preci-

sion, **IND-ORIG-P**. The comparison of these methods is shown in Table 8.5. The choice of the summary sentence has a noticeable impact on performance. We hypothesize that the coherence lost in the summaries is especially important for the longer CNNDM summaries. Using important sentences other than the first sentence likely adds more diversity in the data, and finding a balance between coherence and output style is an interesting direction for additional work (Christensen et al., 2013).

Effect of lead bias on CNNDM fine-tuning We examined the effect of selecting the M sentences greedily chosen for calculating the extractive oracle and inserting them at the beginning of the unsupervised source document versus leaving them in place for CNNDM intermediate fine-tuning. This is meant to mirror the lead bias present in the dataset. This had a slight impact on performance (40.14 vs 39.74 without this bias), and thus we keep the lead bias for CNNDM experiments.

Wikipedia vs target domain unlabeled data While Wikipedia is a natural source of unlabeled data, we tested whether creating unsupervised data from unlabeled in-domain data improved results. We performed the same dataset creation treating the source data of the target domain as we did the Wikipedia data. This resulted in about 60k examples for CNNDM and 200k examples for XSum. Fine-tuning on this data, however, resulted in a performance of 38.08/25.83 ROUGE-1 for CNNDM and XSum (vs 39.11/31.85 on WikiTransfer data). The removal of the first sentences may remove too much information in the case of CNNDM, while for XSum, which already has an initial sentence headline removed as the summary, the first sentence may not constitute a very good summary of the remaining document. Wikipedia data often contains multi-paragraph introductions; thus the removal of the first few sentences may still leave a pyramid-structured document with coherent informative content placed at the front. This result supports the emphasis on learning the subaspects of the target domain over simply in-domain training. An analysis of the output of intermediate fine-tuning on CNNDM reveals that the output was more

Target Dataset	First M Sents	IND-ORIG	IND-ORIG-P
CNNDM	40.14/17.71/36.66	37.62/15.15/34.21	37.85/15.32/34.39
XSum	31.80/10.46/23.66	29.95/9.37/21.78	30.22/9.79/23.23

Table 8.5: A comparison of the effect of summary sentence choice for WikiTransfer on zero-shot transfer ROUGE-1/2/L performance.

abstractive, due to information present in the summary not being directly stated in the source, than fine-tuning on Wikipedia. We also experiment with further in-domain pretraining of BART before zero-shot transfer, but this does not result in consistent improvements across datasets.

8.6 Few-Shot Transfer Results

We examine whether zero-shot transfer improvements also carry over to the few-shot setting. Also, we explore the effect of data augmentation and consistency regularization techniques. The results of our experiments with varying training data sizes and augmentation methods for all 4 datasets are shown in Figure 8.4 and, with complete numbers, in Table 8.6.

10 and 100-shot performance with round-trip translation augmentation We see that in few-shot settings, without data augmentation or consistency training, our model outperforms transferring from another domain or vanilla BART. In the case of transfer to Reddit, we observe that despite similar zero-shot performance with transfer from CNNDM, there is a more sizeable gap with 10-shot transfer. This suggests that our intermediate fine-tuning does more closely align the BART model with the target domain. Furthermore, when training on augmented data from round-trip translation, we see the best performance in transfer from WikiTransfer in all cases except BART transfer to CNNDM on 10-aug, which is likely due to the autoencoder pretraining objective of BART which biases it towards copying and lead bias, allowing it to perform well in applications to CNNDM. We see improvements when training with augmented data in 10-example cases and most 100-example cases for WikiTransfer. Less improvement is seen in the 100-aug setting when transferring from

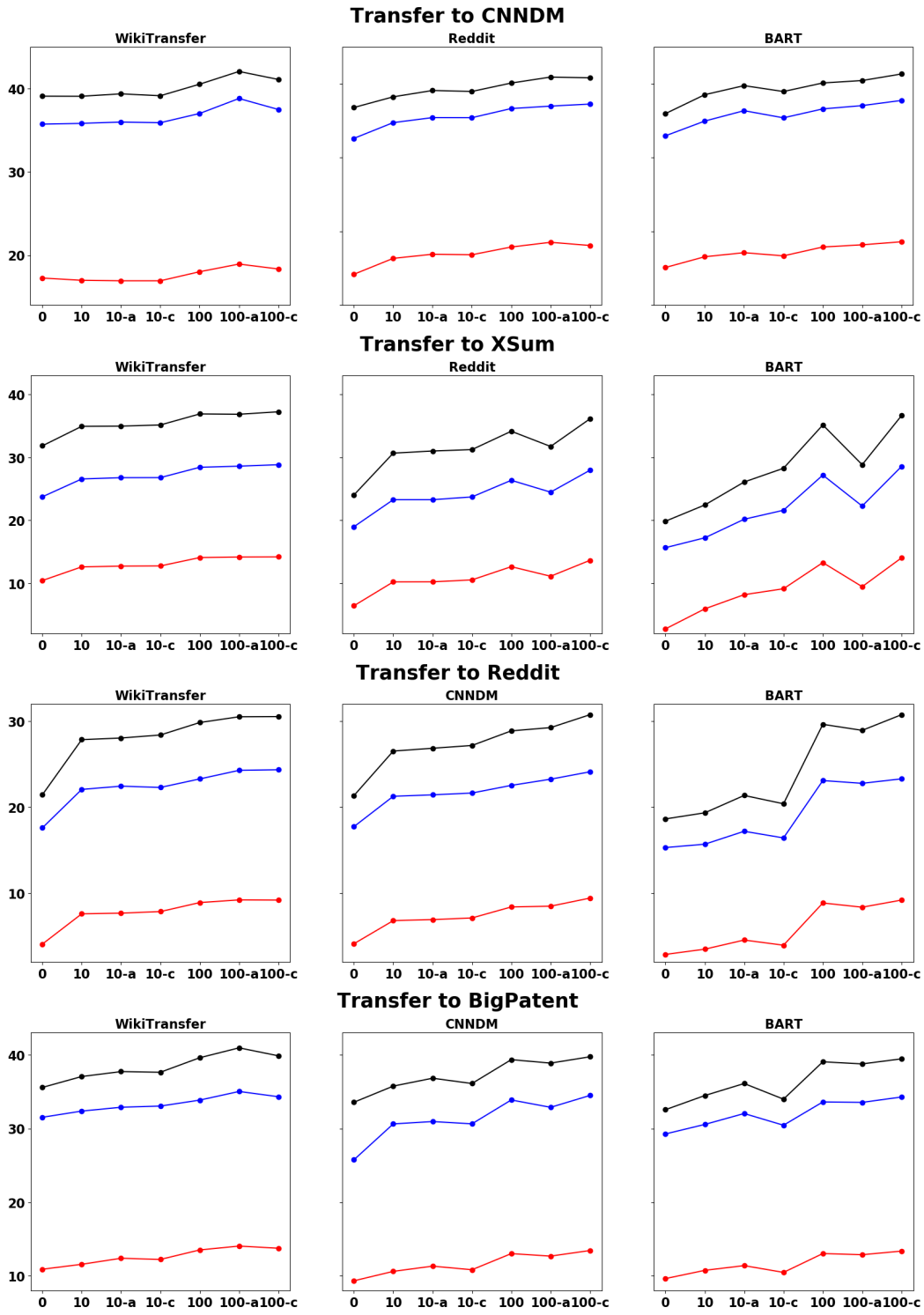

Figure 8.4: ROUGE-1/2/L scores across datasets, training dataset size, data augmentation (***-a**), and consistency loss (***-c**) showing the generalizable and robust performance of models transferred from WikiTransfer.

BART or another domain. We hypothesize that the noise present in the larger augmented dataset causes this occasional performance drop, while the WikiTransfer models appear more robust to potential noise. We also found model robustness as the standard deviation of top-performing WikiTransfer models was least among all models in the majority of cases. Interestingly, for transfer from BART and another domain 100-aug only improves on CNNDM, the most extractive dataset, while the largest drop in performance from augmented data occurs on XSum. This XSum performance drop may be caused by the high compression in the XSum summaries which leaves less room for noisy output when compared to the longer CNNDM and BigPatent summaries which may still preserve the main meaning of the original summary better despite backtranslation noise. In most cases, 100-aug with WikiTransfer results in the best performance, only several points from the state-of-the-art supervised performance.

Transfer with Consistency Training We find contrasting trends with the added consistency loss compared to data augmentation via round-trip translation. We note the most sizeable improvements in the more abstractive cases of XSum and Reddit. We hypothesize that the consistency loss promotes better abstraction as the model learns to be invariant to noise which does not change the meaning of the text, and is thus equipped with a better notion of paraphrasing. The consistency loss allows for better training of vanilla BART as well as in general better transfer from other domains than without consistency loss. The loss likely provides a regularization factor which prevents the models from overfitting to the supervised examples. As the WikiTransfer model is already more closely tuned to the target domain, this regularization may not make as large of a difference. This aligns with our observation of WikiTransfer models being more robust to noisy backtranslated data on XSum and Reddit. Transfer to Reddit shows similar results across models for consistency loss with 100 examples (better ROUGE-L for WikiTransfer, better ROUGE-1/2 for Reddit); vanilla BART's strong performance at 100 examples suggests that the information provided

Target Dataset	CNNDM		
Transfer from	WikiTransfer	Reddit	BART
0	**39.11/17.25/35.73**	36.81/14.18/32.62	35.98/15.10/32.97
10	**39.10/16.98/35.84**	38.26/16.34/34.76	38.55/16.56/34.97
10-aug	39.39/16.92/36.00	39.12/16.90/35.44	**39.78/17.11/36.38**
10-cons	**39.16/16.96/35.92**	38.99/16.83/35.43	38.98/16.68/35.41
100	**40.55/18.01/37.03**	40.13/17.88/36.67	40.14/17.88/36.62
100-aug	**42.08/18.93/38.83**	40.94/18.52/37.00	40.47/18.18/37.07
100-cons	41.12/18.34/37.51	40.84/18.09/37.28	**41.36/18.59/37.77**

Target Dataset	XSum		
Transfer from	WikiTransfer	Reddit	BART
0	**31.85/10.44/23.75**	24.04/6.43/18.99	19.87/2.75/15.66
10	**34.95/12.61/26.58**	30.69/10.22/23.29	22.45/5.94/17.23
10-aug	**34.98/12.73/26.79**	31.03/10.23/23.29	26.10/8.19/20.18
10-cons	**35.17/12.76/26.80**	31.25/10.54/23.73	28.28/9.13/21.61
100	**36.92/14.09/28.44**	34.17/12.64/26.37	35.17/13.29/27.20
100-aug	**36.87/14.18/28.62**	31.75/11.12/24.49	28.85/9.46/22.28
100-cons	**37.26/14.20/28.85**	36.14/13.65/27.97	36.65/14.05/28.57

Target Dataset	Reddit		
Transfer from	WikiTransfer	CNNDM	BART
0	**21.47**/4.10/17.62	21.37/**4.14/17.76**	18.66/2.90/15.33
10	**27.88/7.62/22.09**	26.55/6.83/21.29	19.37/3.51/15.72
10-aug	**28.07/7.70/22.47**	26.88/6.95/21.46	21.39/4.57/17.22
10-cons	**28.42/7.88/22.32**	27.20/7.14/21.67	20.42/3.97/16.45
100	**29.87/8.93/23.31**	28.90/8.42/22.56	29.66/8.88/23.12
100-aug	**30.54/9.24/24.31**	29.28/8.51/23.28	28.96/8.39/22.80
100-cons	30.56/9.22/**24.38**	**30.78/9.45**/24.14	**30.78**/9.22/23.32

Target Dataset	BigPatent		
Transfer from	WikiTransfer	CNNDM	BART
0	**35.58/10.91/31.53**	33.57/9.34/25.76	32.56/9.64/29.27
10	**37.06/11.58/32.37**	35.76/10.62/30.63	34.48/10.76/30.56
10-aug	**37.73/12.40/32.89**	36.83/11.33/30.95	36.11/11.40/32.04
10-cons	**37.64/12.24/33.05**	36.11/10.84/30.64	33.99/10.48/30.45
100	**39.61/13.53**/33.86	39.35/13.03/**33.88**	39.06/13.04/33.61
100-aug	**40.95/14.05/35.03**	38.88/12.69/32.88	38.77/12.88/33.55
100-cons	**39.87/13.76**/34.32	39.74/13.45/**34.49**	39.46/13.37/34.28

Table 8.6: A comparison of transfer results across datasets, training dataset size, data augmentation techniques, showing the generalizable and robust performance of our models transferred from WikiTransfer.

in this subset is sufficient for good performance, thus diminishing the gains from the head-start the WikiTransfer model provides in zero and 10-shot transfer. We leave aspects of the consistency training such as the role of the quality of the round-trip translation data and its relation to the transfer domain to future work.

Target Dataset	WikiTransfer			Pegasus (Zhang et al., 2019)		
# training samples	0	10	100	0	10	100
CNNDM	**39.11/17.25/35.73**	**39.39/16.92/36.00**	**42.08/18.93/38.83**	32.90/13.28/29.38	37.25/15.84/33.49	40.28/18.21/37.03
XSum	**31.85/10.44/23.75**	**35.17/12.76/26.80**	37.26/14.20/28.85	19.27/3.00/12.72	19.39/3.45/14.02/	**39.07/16.44/31.27**
Reddit	**21.47/4.10/17.62**	**28.42/7.88/22.32**	**30.56/9.22/24.38**	14.66/3.06/10.17	15.36/2.91/10.76	16.64/4.09/12.92
BigPatent	**35.58/10.91/31.53**	**37.73/12.40/32.89**	**40.95/14.05/35.03**	25.61/6.56/17.42	28.87/8.30/19.71	33.52/10.82/22.87

Table 8.7: A comparison of zero and few-shot performance between our best-performing WikiTransfer model (-aug in the case of CNNDM and BigPatent and -cons for XSum and Reddit) and the zero and few-shot results reported in Zhang et al. (2019).

Target Dataset	CNNDM		XSum	
	Relevance	Consistency	Relevance	Consistency
0	**4.37**	**4.71**	3.75*	**3.75**
10-aug	4.31	4.76	3.77*	4.10
100-aug	4.25	4.86	**4.00**	4.04
Full supervision	4.31	4.86	4.11	3.98

Table 8.8: Summary relevance and factual consistency across CNNDM and XSum datasets with varying amounts of training data. All results except those with an asterisks do not differ in a statistically significant way (p-value of 0.05) from the full supervision score. Bold results emphasize the least amount of data to achieve statistically indistinguishable results from the fully-supervised results.

Comparison to Previous Work We show a comparison of our best-performing WikiTransfer few-shot results with those from Zhang et al. (2019) in Table 8.7. The Pegasus numbers were obtained by a single run as opposed to our average of the best three over 5 subsets. We show large improvements with our few-shot approach compared to previous numbers, except for the 100-shot experiment on XSum. The XSum dataset has the highest overlap with the Pegasus pretraining dataset of all datasets explored in Zhang et al. (2019), although that work states that the effect of removing this overlap does not affect the full-dataset performance. We hope that this comparison promotes future benchmarking of few-shot results.

Human Quality Assessment We examine how the improved performance from WikiTransfer manifests itself in qualitative annotations when varying the amount of training data. We collect human judgment annotations for two of the four quality dimensions studied in Kryscinski et al. (2019); Fabbri et al. (2020), namely consistency and relevance. Consistency is defined as the factual alignment between the summary and the summarized source text,

while relevance is defined as the selection of important content; only relevant information should be included in the summary. We did not include fluency as a dimension as an initial inspection of the data found fluency to be of very high quality, and we did not include coherence due to our inclusion of single-sentence XSum summaries where coherence is not a factor. We randomly select 50 examples per dataset and collect the model output from the best-performing zero-shot, 10-aug, 100-aug, and fully supervised models on CNNDM and XSum. The annotator sees the source article and randomly-ordered output from the four models rates the summaries for relevance and consistency on a Likert from 1-5, with 5 being the best score. We averaged the score of two native English-speaking annotators on each example and then across examples, and found moderate and strong annotator correlations for relevance and consistency, respectively. Results are shown in Table 8.8. For CNNDM, we see an increase in consistency as more training data is added but not a statistically significant difference (using a Student's t-test with a p-value of 0.05) between 100 and full supervision for any of the relevance or consistency results. The relevance of the full model does not outperform the others, likely because the model output was more concise and was judged as not including source information, while the zero-shot output more closely resembles the lead-three bias, so was judged as more informative. For XSum, we see that relevance improves noticeably as more training data is used. We see varied results for consistency, although without statistically significant differences. This fluctuation in scores may be due to the transition of the model from using knowledge from pretraining in its output versus knowledge from the target dataset obtained during fine-tuning, as seen in a qualitative examination of the model outputs.

Sample Summary Outputs We include an example of model output summaries on the XSum dataset in Table 8.9. The example serves to demonstrate how output style varies as the amount of training data is increased and how the source of pretraining or fine-tuning data affects this style and model hallucinations. The source document does not state the

Source Document: Ms Jones told BBC Radio Wales she did not want to give up being an AM to go to Brussels to replace Nathan Gill, UKIP Wales leader. Mr Gill has been told by the UKIP assembly group and the UKIP party chairman Steve Crowther to stop ”double-jobbing” as an AM and MEP. Mr Gill said those making such calls were doing it out of ”malice”. ”We’ve got Brexit now and I think that, possibly, it may be best to leave that role unfilled,” Ms Jones told the Good Morning Wales programme. ”I’m surprised I’ve not been formally asked what I’d like to do.” Ms Jones, the South Wales West AM, is one of two people who could take up the role of UKIP Wales MEP if Mr Gill made it vacant - the other being South Wales East AM David Rowlands...
0: Lorraine Jones is a Welsh Labour Party Member of the Welsh Assembly for South Wales West.
10-aug: Lorraine Jones is a Welsh Labour member of the Welsh Assembly for South Wales West.
100-aug: Wales Assembly Member for South Wales West Rachel Jones says she has not been formally asked to become a UKIP MEP.
Full supervision: First Minister Carwyn Jones has said she is ”surprised” she has not been asked to become a UKIP MEP.
Gold Summary: UKIP’s Welsh MEP post may be better left unfilled as a result of Brexit , party AM Caroline Jones has said .

Table 8.9: An example of WikiTransfer model output across dataset size used in fine-tuning, illustrating how model output style and hallucinated entities differ as the model moves from Wikipedia pretraining as a source of knowledge to the target dataset. Text not stated in the source document is highlighted in red.

first name of Ms. Jones, yet every model output, and the gold target, give her one. For zero and 10-aug, the model outputs Lorraine Jones, likely still under the influence of BART Wikipedia pretraining, as there is a Wikipedia article on the Welsh politician Ruth Lorraine Jones (although it does not appear in our intermediate fine-tuning subset). The zero and 10-aug also most resemble Wikipedia introduction sentences; although the output is compact and abstractive like an XSum target sentence, the ”X is Y” format of Wikipedia appears. We see at 100-aug examples that the model output is stylistically already much like that of the fully-supervised output and gold summary. This stylistic change is also reflected in the change in hallucination; the use of Rachel Jones is likely caused by the appearance of the name of a minister Rachel Haves in an article on Welsh politics found in the 100-aug subset. The model at this point is already fitting strongly to the target domain. For the fully supervised output, we see the use of Carwyn Jones, which does not match the gender of Ms. Jones but which is found 1090 times in the training source documents. Caroline Jones, the actual person in question, only appears 21 times in the training set. This phenomenon

points to two interesting research directions for future work, how to properly preserve world knowledge from pretraining and improvement faithfulness to the source text in knowing when to insert world knowledge.

8.7 Summary

We introduced WikiTransfer, a novel and generalizable method for fine-tuning pretrained models on dataset-specific unsupervised data obtained from generic Wikipedia data. WikiTransfer models achieve state-of-the-art zero-shot abstractive summarization performance on the CNN-DailyMail dataset and generalize across three additional datasets. In few-shot settings, WikiTransfer models are robust to noise from data augmentation and benefit from consistency loss on more abstractive datasets. Furthermore, human assessments of the resulting summaries do not show significant differences between the WikiTransfer few-shot summaries and fully-supervised summaries, demonstrating the efficiency of our approach.

Part III

Taking Stock of Text Summarization Advances

Chapter 9

SummEval: Re-evaluating Summarization Evaluation

Throughout the previous chapters, we have seen the application of neural network models across both high-resource and low-resource settings. We have seen the remarkable abilities of these models on large-scale datasets, as well as some of their downfalls in Chapter 5, and, more recently, the ability to make smarter use of few data points. However, questions remain as to the extent of the progress made in the field due to variability in the evaluation protocol, which we address in this chapter. The scarcity of comprehensive, up-to-date studies on evaluation metrics for text summarization and the lack of consensus regarding evaluation protocols continue to inhibit progress. We address the existing shortcomings of summarization evaluation methods along five dimensions: 1) we re-evaluate 14 automatic evaluation metrics in a comprehensive and consistent fashion using neural summarization model outputs along with expert and crowd-sourced human annotations, 2) we consistently benchmark 23 recent summarization models using the aforementioned automatic evaluation metrics, 3) we assemble the largest collection of summaries generated by models trained on the CNNDM news dataset and share it in a unified format, 4) we implement and share a toolkit that provides an extensible and unified API for evaluating summarization models

across a broad range of automatic metrics, 5) we assemble and share the largest and most diverse, in terms of model types, collection of human judgments of model-generated summaries on the CNNDM dataset annotated by both expert judges and crowd-source workers. We hope that this work will help promote a more complete evaluation protocol for text summarization as well as advance research in developing evaluation metrics that better correlate with human judgments.

9.1 Introduction

While the current setup has become standardized, we believe several factors prevent a more complete comparison of models, thus negatively impacting the progress of the field.

As noted by Hardy et al. (2019), recent papers vastly differ in their evaluation protocol. Existing work often limits model comparisons to only a few baselines and offers human evaluations which are largely inconsistent with prior work. Additionally, despite problems associated with ROUGE when used outside of its original setting (Liu and Liu, 2008; Cohan and Goharian, 2016) as well as the introduction of many variations on ROUGE (Zhou et al., 2006; Ng and Abrecht, 2015; Ganesan, 2015; ShafieiBavani et al., 2018) and other text generation metrics (Peyrard, 2019; Zhao et al., 2019; Zhang et al., 2020; Scialom et al., 2019; Clark et al., 2019), ROUGE has remained the default automatic evaluation metric. We believe that the shortcomings of the current evaluation protocol are partially caused by the lack of easy-to-use resources for evaluation, both in the form of simplified evaluation toolkits and large collections of model outputs.

In parallel, there is an issue with how evaluation metrics are evaluated themselves. Many of the currently used metrics were developed and assessed using the Document Understanding Conference (DUC) and Text Analysis Conference (TAC) shared-tasks datasets (Dang and Owczarzak, 2008, 2009). However, it has recently been shown that the mentioned datasets contain human judgments for model outputs scoring on a lower scale compared to

current summarization systems putting into question the true performance of those metrics in the new setting (Peyrard, 2019).

We address these gaps in complementary ways: 1) We re-evaluate 14 automatic evaluation metrics in a comprehensive and consistent fashion using outputs from recent neural summarization models along with expert and crowd-sourced human annotations, 2) We consistently benchmark 23 recent summarization models using the aforementioned automatic evaluation metrics, 3) We release aligned summarization model outputs from 23 papers (44 model outputs) published between 2017 and 2019 trained on the CNNDM dataset to allow for large-scale comparisons of recent summarization models, 4) We release a toolkit of 14 evaluation metrics with an extensible and unified API to promote the reporting of additional metrics in papers, 5) We collect and release expert, as well as crowd-sourced, human judgments for 16 model outputs on 100 articles over 4 dimensions to further research into human-correlated evaluation metrics. Code and data associated with this work is available at `https://github.com/Yale-LILY/SummEval`.

9.2 Related Work

Previous work examining the research setup of text summarization can be broadly categorized into three groups, based on the subject of analysis: evaluation metrics, datasets, and models.

Dealing with evaluation methods, Lin (2004b) examined the effectiveness of the ROUGE metric in various DUC tasks. The authors concluded that evaluating against multiple references results in higher correlation scores with human judgments, however, a single-reference setting is sufficient for the metric to be effective. Owczarzak et al. (2012) studied the effects of inconsistencies in human annotations on the rankings of evaluated summarization systems. Results showed that system-level rankings were robust against annotation inconsistencies, however, summary-level rankings were not stable in such settings and largely benefit from

improving annotator consistency. Rankel et al. (2013) analyzed the performance of different variants of the ROUGE metric using TAC datasets. The authors found that higher-order and less commonly reported ROUGE settings showed a higher correlation with human judgments. In a similar line of work, Graham (2015) conducted a large-scale study of the effectiveness of different ROUGE metric variants and compared it against the BLEU metric on the DUC datasets. Its results highlighted several superior, non-standard ROUGE settings that achieved strong correlations with human judgments on model-generated summaries. In (Chaganty et al., 2018) the authors investigated using an automatic metric to reduce the cost of human evaluation without introducing bias. Together with the study, the authors released a set of human judgments over several model outputs, limited to a small set of model types. Peyrard (2019) showed that standard metrics are in agreement when dealing with summaries in the scoring range found in TAC summaries, but vastly differ in the higher-scoring range found in current models. The authors reported that additional human annotations on modern model outputs are necessary to conduct a conclusive study of evaluation metrics. Hardy et al. (2019) underscore the differences in approaches to human summary evaluation while proposing a highlight-based reference-less evaluation metric. Other work has examined the problems with applying ROUGE in settings such as meeting summarization (Liu and Liu, 2008) and summarization of scientific articles (Cohan and Goharian, 2016). We build upon this line of research by examining the performance of several automatic evaluation methods, including ROUGE and its variants, against the performance of expert human annotators.

In relation to datasets, Dernoncourt et al. (2018) presented a detailed taxonomy of existing summarization datasets. The authors highlighted the differences in formats of available corpora and called for creating a unified data standard. In a similar line of research, Grusky et al. (2018) offered a thorough analysis of existing corpora, focusing their efforts on news summarization datasets. The authors also introduced several metrics for evaluating the extractiveness of summaries which are included in the toolkit implemented as part of this work. Kryscinski et al. (2020a) showed that news-related summarization datasets, such

as CNNDM, contain strong layout biases. The authors revealed that datasets in the current format, where each news article is associated with a single reference summary, leave the task of summarization underconstrained. The paper also highlighted the problem of noisy, low-quality data in automatically-collected news datasets.

Looking into models, Zhang et al. (2018a) analyzed the level of abstraction of several recent abstractive summarization models. The authors showed that word-level extractive models achieved a similar level of abstraction to fully abstractive models. In (Kedzie et al., 2018) the authors examined the influence of various model components on the quality of content selection. The study revealed that in the current setting the training signal is dominated by biases present in summarization datasets preventing models from learning accurate content selection. Kryscinski et al. (2020a) investigate the problem of factual correctness of text summarization models. The authors concluded that the issue of hallucinating facts touches up to 30% of generated summaries and list common types of errors made by generative models. Closely related to that work, Maynez et al. (2020) conducted a large-scale study of abstractive summarizers from the perspective of faithfulness. The authors reached similar conclusions, stating that improving factual faithfulness is a critical issue in summarization. The results also showed that currently available evaluation methods, such as ROUGE and BertScore, are not sufficient to study the problem at hand. Durmus et al. (2020) and Wang et al. (2020) similarly examine faithfulness evaluation, both proposing question answering frameworks as a means of evaluating factual consistency.

Insights and contributions coming from our work are complementary to the conclusions of previous efforts described in this section. To the best of our knowledge, this is the first work in neural text summarization to offer a large-scale, consistent, side-by-side re-evaluation of summarization model outputs and evaluation methods. We also share resources that we hope will prove useful for future work in analyzing and improving summarization models and metrics.

Shortly before publishing this manuscript a library for developing summarization metrics

was released by Deutsch and Roth (2020). Our toolkit is complementary to their work as their toolkit includes only 3 of our 12 evaluation metrics.

9.3 Evaluation Metrics and Summarization Models

We briefly introduce metrics included in our evaluation toolkit as well as the summarization models for which outputs were collected at the time of releasing this manuscript.

Evaluation Metrics Our selection of evaluation methods includes several recently introduced metrics that have been applied to both text generation and summarization, standard machine translation metrics, and other miscellaneous performance statistics.

ROUGE (Lin, 2004a), (Recall-Oriented Understudy for Gisting Evaluation), measures the number of overlapping textual units (n-grams, word sequences) between the generated summary and a set of gold reference summaries.

ROUGE-WE (Ng and Abrecht, 2015) extends ROUGE by using soft lexical matching based on the cosine similarity of Word2Vec (Mikolov et al., 2013b) embeddings.

$\boldsymbol{S^3}$ (Peyrard et al., 2017) is a model-based metric that uses previously proposed evaluation metrics, such as ROUGE, JS-divergence, and ROUGE-WE, as input features for predicting the evaluation score. The model is trained on human judgment datasets from TAC conferences.

BertScore (Zhang et al., 2020) computes similarity scores by aligning generated and reference summaries on a token-level. Token alignments are computed greedily to maximize the cosine similarity between contextualized token embeddings from BERT.

MoverScore (Zhao et al., 2019) measures the semantic distance between a summary and reference text by making use of the Word Mover's Distance (Kusner et al., 2015) operating over n-gram embeddings pooled from BERT representations.

Sentence Mover's Similarity (SMS) (Clark et al., 2019) extends Word Mover's Distance to view documents as a bag of sentence embeddings as well as a variation which represents

documents as both a bag of sentences and a bag of words.

SummaQA (Scialom et al., 2019) applies a BERT-based question-answering model to answer cloze-style questions using generated summaries. Questions are generated by masking named entities in source documents associated with evaluated summaries. The metric reports both the F1 overlap score and QA-model confidence.

BLANC (Vasilyev et al., 2020) is a reference-less metric which measures the performance gains of a pre-trained language model given access to a document summary while carrying out language understanding tasks on the source document's text.

SUPERT (Gao et al., 2020) is a reference-less metric, originally designed for multi-document summarization, which measures the semantic similarity of model outputs with pseudo-reference summaries created by extracting salient sentences from the source documents, using soft token alignment techniques.

BLEU (Papineni et al., 2002) is a corpus-level precision-focused metric which calculates n-gram overlap between a candidate and reference utterance and includes a brevity penalty. It is the primary evaluation metric for machine translation.

CHRF (Popović, 2015) calculates character-based n-gram overlap between model outputs and reference documents.

METEOR (Lavie and Agarwal, 2007) computes an alignment between candidate and reference sentences by mapping unigrams in the generated summary to 0 or 1 unigrams in the reference, based on stemming, synonyms, and paraphrastic matches. Precision and recall are computed and reported as a harmonic mean.

CIDEr (Vedantam et al., 2015) computes {1-4}-gram co-occurrences between the candidate and reference texts, down-weighting common n-grams and calculating cosine similarity between the n-grams of the candidate and reference texts.

Data Statistics: Grusky et al. (2018) define three measures of the extractiveness of a dataset, which were discussed in greater detail in Chapter 4. *Extractive fragment coverage* is the percentage of words in the summary that are from the source article, measuring the extent

to which a summary is a derivative of a text. *Density* is defined as the average length of the extractive fragment to which each summary word belongs. *Compression ratio* is defined as the word ratio between the articles and its summaries: In addition to these measures, we also include the percentage of n-grams in the summary not found in the input document as a *novelty* score and the percentage of n-grams in the summary which repeat as a score of *redundancy*. For a comprehensive explanation of each metric, please refer to the corresponding paper.

Summarization models We broadly categorize the models included in this study into extractive and abstractive approaches. For each model, we provide a model code (M*) as well as a descriptive model name which will allow for easy matching with the released data.

Extractive Methods

M1 - **NEUSUM** (Zhou et al., 2018) jointly scores and selects sentences by first building a hierarchical representation of a document and considering the partially outputted summary at each time step.

M2 - **BanditSum** (Dong et al., 2018) treats extractive summarization as a contextual bandit problem where the document is the context and the sequence of sentences to include in the summary is the action.

M3 - **LATENT** (Zhang et al., 2018d) propose a latent variable extractive model which views rele-vance labels of sentences in a document as binarylatent variables

M4 - **REFRESH** (Narayan et al., 2018b) propose using REINFORCE (Williams, 1992b) to extract summaries, approximating the search space during training by limiting to combinations of individually high-scoring sentences.

M5 - **RNES** (Wu and Hu, 2018) propose a coherence model to capture cross-sentence coherence, combining output from the coherence model and ROUGE scores as a reward in a REINFORCE framework.

M6 - **JECS** (Xu and Durrett, 2019) first extracts sentences from a document and then scores possible constituency-based compressed units to produce the final compressed summary.
M7 - **STRASS** (Bouscarrat et al., 2019) extracts a summary by selecting the sentences with the closest embeddings to the document embedding, learning a transformation to maximize the similarity between the summary and the ground truth reference.

Abstractive Methods

M8 - **Pointer Generator** (See et al., 2017) propose a variation of encoder-decoder models, the Pointer Generator Network, where the decoder can choose to generate a word from the vocabulary or copy a word from the input. A coverage mechanism is also proposed to prevent repeatedly attending to the same part of the source document.
M9 - **Fast-abs-rl** (Chen and Bansal, 2018) propose a model which first extracts salient sentences with a Pointer Network and rewrites these sentences with a Pointer Generator Network. In addition to maximum likelihood training, a ROUGE-L reward is used to update the extractor via REINFORCE (Williams, 1992b).
M10 - **Bottom-Up** (Gehrmann et al., 2018) introduce a bottom-up approach whereby a content selection model restricts the copy attention distribution of a pretrained Pointer Generator Network during inference.
M11 - **Improve-abs** (Kryściński et al., 2018) extend the model of Paulus et al. (2018) by augmenting the decoder with an external LSTM language model and add a novelty RL-based objective during training.
M12 - **Unified-ext-abs** (Hsu et al., 2018) propose to use the probability output of an extractive model as sentence-level attention to modify word-level attention scores of an abstractive model, introducing an inconsistency loss to encourage consistency between these two levels of attention.
M13 - **ROUGESal** (Pasunuru and Bansal, 2018) propose a keyphrase-based salience reward as well as an entailment-based reward in addition to using a ROUGE-based reward in a

REINFORCE setting, optimizing rewards simultaneously in alternate mini-batches.

M14 - **Multi-task (Ent + QG)** (Guo et al., 2018) propose question generation and entailment generation as auxiliary tasks in a multi-task framework along with a corresponding multi-task architecture.

M15 - **Closed book decoder** (Jiang and Bansal, 2018) build upon a Pointer Generator Network by adding copy-less and attention-less decoder during training time to force the encoder to be more selective in encoding salient content.

M16 - **SENECA** (Sharma et al., 2019a) propose to use entity-aware content selection module and an abstractive generation module to generate the final summary.

M17 - **T5** (Raffel et al., 2019) perform a systematic study of transfer learning techniques and apply their insights to a set of tasks all framed as text-input to text-output generation tasks, including summarization.

M18 - **NeuralTD** (Böhm et al., 2019) learn a reward function from 2,500 human judgments which is used in a reinforcement learning setting.

M19 - **BertSum-abs** (Liu and Lapata, 2019a) introduce a novel document-level encoder on top of BERT (Devlin et al., 2019), over which they introduce both an extractive and an abstractive model.

M20 - **GPT-2** (Ziegler et al., 2019) build off of GPT-2 (Radford et al., 2019) and fine-tune the model by using human labels of which of four sampled summaries is the best to direct fine-tuning in a reinforcement learning framework.

M21 - **UniLM** (Dong et al., 2019) introduce a model pretrained on three language modeling tasks: unidirectional, bidirectional, and sequence-to-sequence prediction. It is thus applicable to natural language understanding tasks and generation tasks such as abstractive summarization.

M22 - **BART** (Lewis et al., 2020) introduce a denoising autoencoder for pretraining sequence to sequence tasks which is applicable to both natural language understanding and generation tasks.

M23 - **Pegasus** (Zhang et al., 2019) introduce a model pretrained with a novel objective function designed for summarization by which important sentences are removed from an input document and then generated from the remaining sentences.

9.4 Evaluation Resources

We now describe the resources collected and released together with this manuscript.

Model Outputs The model output collection contains summaries associated with 23 recent papers on neural text summarization described in Section 9.3. We obtained a total of 44 model outputs, as many papers include variations of the main model. All models were trained on the CNNDM news corpus and the collected summaries were generated using the test split of the dataset without constraints limiting the output length. Outputs were solicited from the authors of papers to ensure comparability between results presented in this paper with those in the original works. They are shared publicly with the consent of the authors.

Model outputs were transformed into a unified format and are shared with IDs of the original CNNDM examples so that generated summaries can be matched with corresponding source articles. Pairing model outputs with original articles was done using a heuristic approach that relied on aligning reference summaries. The pairing process revealed that 38 examples in the CNNDM test split contained duplicate reference summaries preventing those examples to be correctly aligned. However, this problem involves only 0.3% of the available data and should not have a notable impact on downstream results. IDs of duplicate examples are provided together with the data.

Evaluation Toolkit The evaluation toolkit contains 14 automatic evaluation metrics described in Section 9.3 consolidated into a Python package. The package provides a high-level, easy-to-use interface unifying all of the underlying metrics. For each metric, we implement both `evaluate_example` and `evaluate_batch` functions that return the metric's

score on example- and corpus-levels accordingly. Function inputs and outputs are also unified across all metrics to streamline multi-metric evaluation and result processing. The toolkit comes with a standard configuration resembling the most popular settings for each of the metrics to enable easy, out-of-the-box use. However, each metric can be further configured using external `gin` configuration files. We also provide a command-line tool to evaluate a summarization model with several metrics in parallel.

Human Annotations The collection of human annotations contains summary evaluations of 16 recent neural summarization models solicited from crowd-sourced and expert judges. Annotations were collected for 100 articles randomly picked from the CNNDM test set. To ensure high quality of annotations, each summary was scored by 5 crowd-sourced and 3 expert workers, amounting to 12800 summary-level annotations. Model outputs were evaluated along the following four dimensions, as in Kryscinski et al. (2019):

Coherence - the collective quality of all sentences. We align this dimension with the DUC quality question (Dang, 2005) of structure and coherence whereby "the summary should be well-structured and well-organized. The summary should not just be a heap of related information, but should build from sentence to sentence to a coherent body of information about a topic."

Consistency - the factual alignment between the summary and the summarized source. A factually consistent summary contains only statements that are entailed by the source document. Annotators were also asked to penalize summaries that contained hallucinated facts.

Fluency - the quality of individual sentences. Drawing again from the DUC quality guidelines, sentences in the summary "should have no formatting problems, capitalization errors or obviously ungrammatical sentences (e.g., fragments, missing components) that make the text difficult to read."

Relevance - selection of important content from the source. The summary should include

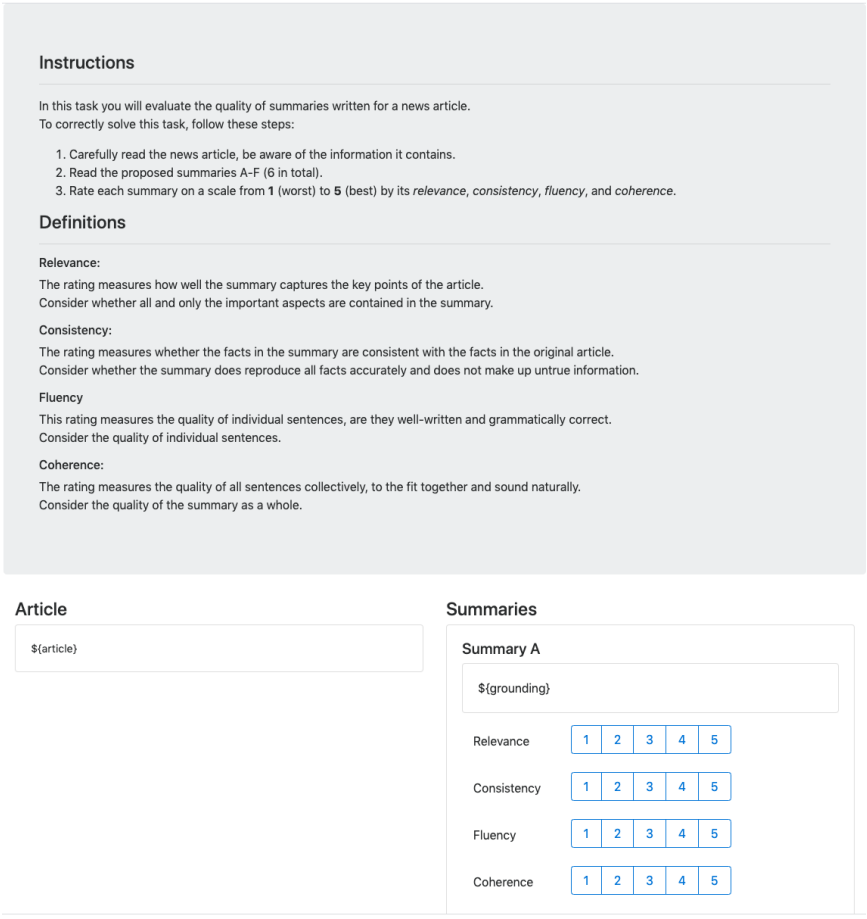

Figure 9.1: Example of the data collection interface used by crowd-source and expert annotators.

only important information from the source document. Annotators were instructed to penalize summaries which contained redundancies and excess information.

The data collection interface provided judges with the source article and associated summaries grouped in sets of 5. Each group of summaries contained the reference summary associated with the source article to establish a common point of reference between groups. Summary grouping and order within groups were randomized for each annotator. Judges were asked to rate the summaries on a Likert scale from 1 to 5 (higher better) along the four mentioned dimensions. The data collection interface used by both crowd-source and expert annotators is presented in Figure 9.1. In the annotation process, judges were first asked to carefully read the content of the source article and next proceed to evaluating the associated

summaries along four axes: *relevance*, *consistency*, *fluency*, and *coherence*.

Crowd-sourced annotators were hired through the Amazon Mechanical Turk platform. The hiring criteria were set to a minimum of 10000 approved HITs and an approval rate of 97% or higher. Geographic constraints for workers were set to United States, United Kingdom, and Australia to ensure that summaries were evaluated by native English speakers. Compensation was carefully calculated to ensure an average wage of 12 USD per hour.

Gillick and Liu (2010) showed that summary judgments obtained through non-experts may differ greatly from expert annotations and could exhibit worse inter-annotator agreement. As a result, in addition to the hired crowd-sourced workers, we enlisted three expert annotators who have written papers on summarization either for academic conferences (2) or as part of a senior thesis (1). The expert annotators were asked to evaluate the same set of summaries under the same instructions as the hired crowd-sourced workers. For expert judgments, we proceeded with two rounds of annotation to correct any obvious mistakes as well as to confirm judgments and ensure a higher quality of annotations. In the second round, annotators were asked to check all examples for which their score of a dimension differed from another annotator by more than 2 points and where the other annotators were within 1 point of each other. In cases where a score differed by more than 2 points for which such a pattern did not exist, all annotators examined the annotation. When re-evaluating examples, judges were allowed to see scores assigned by other expert annotators in the first round of annotations. While such a setting could undermine the wisdom of the crowd and shift the re-assigned scores towards the average judgment from the first round, we encouraged experts to remain critical and discuss contested examples when necessary.

9.5 Metric Re-evaluation

Human Annotations Considering the concerns raised in previous work (Gillick and Liu, 2010) about the quality differences between crowd-sourced and expert annotations we study

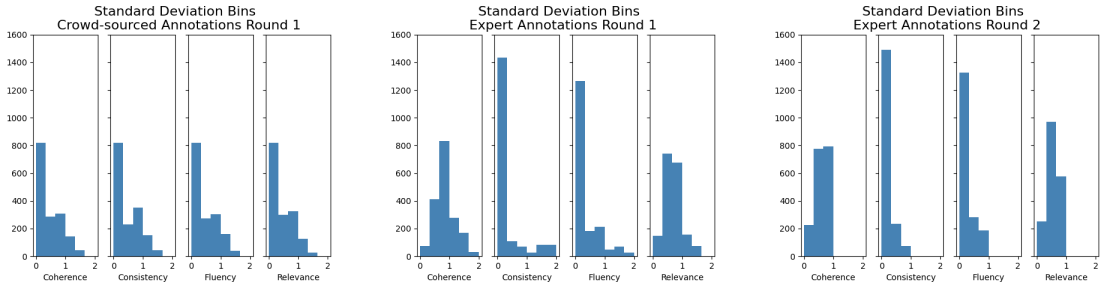

Figure 9.2: Histogram of standard deviations of inter-annotator scores between: crowd-sourced annotations, first round expert annotations, second round expert annotations, respectively.

this issue using the human annotations collected as part of this work.

To evaluate the inter-annotator agreement of collected crowd-sourced and expert annotations we computed the Krippendorff's alpha coefficient (Krippendorff, 2011). We found the inter-annotator interval kappa to be below an acceptable range - 0.4920 and 0.4132 for the crowd-sourced workers and the first round of expert annotations accordingly. However, the second round of expert annotations improved the inter-annotator agreement achieving a kappa coefficient of 0.7127. For further insights, we computed standard deviations of annotator scores within the respective groups and present histograms of those statistics in Figure 9.2. Plots of crowd-sourced annotations show strong similarities across all evaluated dimensions. Such an effect could be caused by an insufficient distinction made by the annotators between the 4 scored axes, where the overall quality of a summary biased scores of the individual dimensions. The histograms also show that while the second round of expert annotations lowered the standard deviation of scores and substantially increased inter-annotator agreement, relevance and coherence remained the most disagreed on dimensions between experts. This could be attributed to the subjective nature of relevance and coherence as an evaluation dimensions (Kryscinski et al., 2020a).

To assess the similarity of annotations between the crowd-sourced and expert annotators we averaged the assigned scores per example within the respective annotator groups and computed Pearson's correlation coefficient. The statistic returned a value close to 0, indicating no correlation between expert and crowd-sourced judges.

Generated Summaries	Expert scores (avg.)	Crowd-worker scores (avg.)
the queen's guard was left red-faced after he slipped on a manhole cover he lost his footing and slid sideways, knocking his bearskin on the side . the embarrassed soldier quickly scrambled to his feet as his colleagues marched past as if nothing had happened . tourist david meadwell recorded the unscheduled manouevre outside buckingham palace on thursday afternoon .	Coh: 5.0 Con: 5.0 Flu: 5.0 Rel: 5.0	Coh: 3.4 Con: 3.8 Flu: 3.4 Rel: 3.8
holidaymaker david meadwell recorded the unscheduled manouevre outside buckingham palace . he lost his footing and slid sideways , knocking bearskin on the side of the box . queen 's guard was left red-faced after he slipped on manhole cover . the entire incident was caught on a manhole cover . the embarrassed soldier quickly scrambled to his feet as his colleagues marched past .	Coh: 2.7 Con: 2.0 Flu: 4.7 Rel: 3.7	Coh: 3.2 Con: 3.4 Flu: 3.4 Rel: 4.0
buckingham palace guard slipped on manhole cover in front of hundreds of horrified tourists . the queen 's guard was left red-faced after he slipped on a manhole cover . he lost his footing and dropped his rifle on the side of the box and dropping his rifle . the incident was caught on camera camera camera . the guard is thought to have slipped because of metal shutters nailed to the soles of his boots .	Coh: 3.3 Con: 5.0 Flu: 1.7 Rel: 4.3	Coh: 3.0 Con: 3.2 Flu: 2.8 Rel: 3.2

(a) Generated summary examples illustrate common problems found in model outputs, such as ambiguous pronouns, incorrect references, and repetitive content.

Reference Summaries	Expert scores (avg.)	Crowd-worker scores (avg.)
river plate admit they ' dream ' of manchester united striker radamel falcao . the colombia international spent eight years with the argentine club . falcao has managed just four goals in 19 premier league appearances . read : falcao still ' has faith ' that he could continue at man utd next season . click here for the latest manchester united news .	Coh: 3.0 Con: 2.0 Flu: 5.0 Rel: 2.3	Coh: 3.0 Con: 3.6 Flu: 3.0 Rel: 4.4
the incident occurred on april 7 north of poland in the baltic sea . u.s. says plane was in international airspace . russia says it had transponder turned off and was flying toward russia	Coh: 2.0 Con: 1.7 Flu: 3.0 Rel: 2.3	Coh: 4.0 Con: 3.4 Flu: 4.2 Rel: 3.6

(b) Reference summaries highlight issues found in the CNN/DailyMail dataset, such as click-baits and references to other articles as well as unreferenced dates and low coherence caused by concatenating bullet-point summaries.

Table 9.1: Example summaries with the corresponding averaged expert and crowd-sourced annotations for *coherence*, *consistency*, *fluency*, and *relevance*. Expert annotations better differentiate coherence, consistency, and fluency among the examples when compared to the crowd-sourced annotations.

We also manually inspected the human annotations and present examples of annotated summaries, both generated and reference, as well as the differences in human judgments in Table 9.1a. The first row shows a well written, comprehensive summary. The high quality of the summary is reflected by top scores assigned by expert annotators, while being rated as average by crowd-sourced workers. The second row shows a summary with ambiguous pronoun usage and factual inconsistencies. The errors result in a decrease in coherence,

consistency, and relevance scores in the expert annotations, but do not see a corresponding decrease in crowd-worker annotations. The third row presents a factually correct summary that contains token and phrase repetitions. The errors were caught by the expert annotators resulting in a low fluency score, while crowd-sourced annotators incorrectly classified them as issues with factual consistency. These examples again illustrate the disparities in the understanding of evaluated dimensions between judges and underscore our observation above about the uniformity of crowd-sourced annotations; the crowd-sourced annotations tend to be similar across quality dimensions even when distinctions exist, which are captured in the expert annotations.

Results presented in this section highlight the difficulties of crowd-sourcing high-quality annotations and the necessity for protocols for improving human evaluation in text summarization.

Automatic Metrics Many automatic metrics have been proposed for evaluating both summarization and other text generation models. However, the field lacks a comprehensive study that would offer a consistent side-by-side comparison of their performance. We address this issue with the following experiments.

In Table 9.2 we show Kendall's tau rank correlations between automatic metrics and human judgments calculated on a system-level following Louis and Nenkova (2013). The statistics were computed using the available expert annotations to avoid possible quality problems associated with crowd-sourced ratings, as highlighted in the previous subsection. Automatic metrics were computed in a multi-reference setting, using the original reference summary included in the CNNDM dataset and 10 additional summaries coming from Kryscinski et al. (2020a), and the length of model outputs was not constrained. We report correlations without differentiating between abstractive and extractive models, as most metrics did not exhibit large differences in correlation when reported separately.

Correlation results show several trends. We find that most metrics have the lowest

Metric	Coherence	Consistency	Fluency	Relevance
ROUGE-1	0.2500	0.5294	**0.5240**	0.4118
ROUGE-2	0.1618	0.5882	0.4797	0.2941
ROUGE-3	0.2206	**0.7059**	**0.5092**	0.3529
ROUGE-4	**0.3088**	0.5882	**0.5535**	0.4118
ROUGE-L	0.0735	0.1471	0.2583	0.2353
ROUGE-su*	0.1912	0.2941	0.4354	0.3235
ROUGE-w	0.0000	0.3971	0.3764	0.1618
ROUGE-we-1	**0.2647**	0.4559	**0.5092**	**0.4265**
ROUGE-we-2	-0.0147	0.5000	0.3026	0.1176
ROUGE-we-3	0.0294	0.3676	0.3026	0.1912
$\boldsymbol{S^3}$-pyr	-0.0294	0.5147	0.3173	0.1324
$\boldsymbol{S^3}$-resp	-0.0147	0.5000	0.3321	0.1471
BertScore-p	0.0588	-0.1912	0.0074	0.1618
BertScore-r	0.1471	**0.6618**	0.4945	0.3088
BertScore-f	0.2059	0.0441	0.2435	**0.4265**
MoverScore	0.1912	-0.0294	0.2583	0.2941
SMS	0.1618	0.5588	0.3616	0.2353
SummaQAˆ	0.1176	**0.6029**	0.4059	0.2206
BLANCˆ	0.0735	0.5588	0.3616	0.2647
SUPERTˆ	0.1029	0.5882	0.4207	0.2353
BLEU	0.1176	0.0735	0.3321	0.2206
CHRF	**0.3971**	0.5294	0.4649	**0.5882**
CIDEr	0.1176	-0.1912	-0.0221	0.1912
METEOR	0.2353	**0.6324**	**0.6126**	**0.4265**
Lengthˆ	-0.0294	0.4265	0.2583	0.1618
Novel unigramˆ	0.1471	-0.2206	-0.1402	0.1029
Novel bi-gramˆ	0.0294	-0.5441	-0.3469	-0.1029
Novel tri-gramˆ	0.0294	-0.5735	-0.3469	-0.1324
Repeated unigramˆ	**-0.3824**	0.1029	-0.0664	-0.3676
Repeated bi-gramˆ	**-0.3824**	-0.0147	-0.2435	**-0.4559**
Repeated tri-gramˆ	-0.2206	0.1471	-0.0221	-0.2647
Stats-coverageˆ	-0.1324	0.3529	0.1550	-0.0294
Stats-compressionˆ	0.1176	-0.4265	-0.2288	-0.0147
Stats-densityˆ	0.1618	**0.6471**	0.3911	0.2941

Table 9.2: Kendall's tau correlation coefficients of expert annotations computed on a system-level along four quality dimensions with automatic metrics using 11 reference summaries per example. ˆ denotes metrics which use the source document. The five most-correlated metrics in each column are bolded.

correlation within the coherence dimension, where the correlation strength can be classified as weak or moderate. This finding follows intuition as the majority of metrics rely on hard or soft subsequence alignments, which do not measure well the interdependence between consecutive sentences. Low and moderate correlation scores were also found for the relevance dimension. As discussed in the previous subsection, such trends could result from the inherent subjectiveness of the dimension and the difficulty of collecting consistent human annotations. Model correlations increase considerably across the consistency and fluency

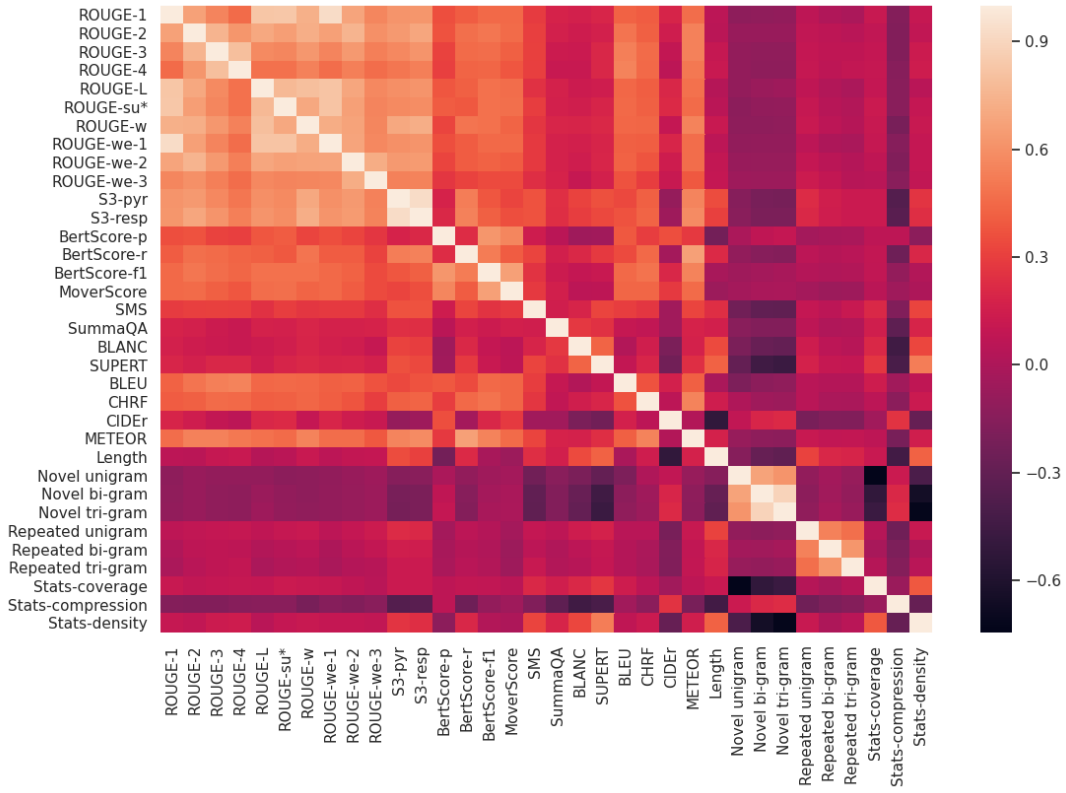

Figure 9.3: Pairwise Kendall's Tau correlations for all automatic evaluation metrics.

dimensions. While unexpected, the strong correlation with consistency could be attributed to the low abstractiveness of most neural models, which could increase the effectiveness of metrics using higher-order n-gram overlap, such as ROUGE-3 or Extractive Density. Referring back to the previous subsection, both of the mentioned dimensions achieved high inter-annotator agreement between expert judges which could also positively affect the correlation scores. Additionally, the results show a substantially higher correlation between all evaluated dimensions and ROUGE scores computed for higher-order n-grams in comparison to ROUGE-L, which corroborates with findings of Rankel et al. (2013).

To examine the dependencies between different metrics we computed Kendall's tau rank correlation coefficients, pairwise, between all metrics. Results are presented as a correlation matrix in Figure 9.3. Following intuition, we observe a strong correlation between all metrics that compute, implicitly or explicitly, the lexical overlap between generated and

reference summaries. Metrics measuring the n-gram novelty and repetitiveness show a weak negative correlation with all ROUGE-related metrics. Length as a feature is weakly correlated with most metrics apart from S^3, BLANC, and SuPERT which might suggest the mentioned metrics favor longer summaries. Worth noting is also the weak correlation of reference-less SummaQA, BLANC, and SuPERT metrics with most other evaluated metrics.

Results presented in this section highlight the evaluation dimensions that are not reliably covered by currently available metrics and pave the way for future work in model evaluation.

9.6 Model Re-evaluation

We now turn to an analysis of model scores across human evaluations and automatic metrics. The evaluated models were released between 2017 and 2019, represent different approaches to summarization: abstractive, extractive, and hybrid, and their architectures reflect the trends in summarization research. Although in many cases we obtained multiple variants of the same model, in the study we focus on the versions with the highest ROUGE-L scores.

Table 9.3 contains the results of human evaluation across the four dimensions described in Section 9.4. Scores for ground truth summaries are included as a point of reference. We find that pretrained models such as Pegasus, BART, and T5 consistently performed best on most dimensions. Notably, the mentioned models scored highest on consistency and fluency while obtaining lower scores for relevance and coherence. Scores for extractive models highlight the known shortcomings of such approaches, which are lack of coherence of summaries and issues with selecting relevant content. Abstractive model ratings show an increasing trend with respect to the date of publication. This is a promising result as it suggests that the quality of models is improving with time. Worth noting is also the fact that reference summaries did not score well on consistency, coherence, and relevance. Upon examination of the annotations, we found that the reference summaries often contained extraneous information, such as hyperlinks and click-bait descriptions of other articles. As

Method	Coherence	Consistency	Fluency	Relevance
CNN/DM Reference Summary	3.26	4.47	4.79	3.77
Extractive Models				
M0 - **LEAD-3**	**4.16**	4.98	4.94	4.14
M1 - **NEUSUM**	3.22	4.98	4.90	3.82
M2 - **BanditSum**	3.28	4.99	4.83	3.81
M5 - **RNES**	3.71	**4.97**	4.81	4.06
Abstractive Models				
M8 - **Pointer Generator**	3.29	4.65	4.79	3.55
M9 - **Fast-abs-rl**	2.38	4.67	4.50	3.52
M10 - **Bottom-Up**	2.73	4.25	4.42	3.38
M11 - **Improve-abs**	2.28	3.27	3.65	3.15
M12 - **Unified-ext-abs**	3.60	**4.96**	4.85	3.85
M13 - **ROUGESal**	3.44	4.82	4.86	3.83
M14 - **Multi-task (Ent + QG)**	3.20	4.90	4.74	3.63
M15 - **Closed book decoder**	3.35	**4.95**	4.80	3.67
M17 - **T5**	4.00	4.93	**4.93**	4.23
M20 - **GPT-2** (zero shot)	3.63	3.40	3.97	3.30
M22 - **BART**	**4.18**	4.94	**4.90**	**4.25**
M23 - **Pegasus** (C4)	**4.16**	4.91	**4.88**	**4.26**
M23 - **Pegasus** (dynamic mix)	4.09	4.85	4.79	**4.27**

Table 9.3: Human ratings of summaries along four evaluation dimensions, averaged over three expert annotators, broken down by extractive and abstractive models. The M* codes follow the notation described in Section 9.3. The three highest-rated models in each column are in bold.

this information was not present in the source documents nor relevant for the summaries, the annotators interpreted it as hallucinations and assigned lower consistency and relevance scores. Additionally, many reference summaries in the CNNDM dataset were constructed by naively concatenating bullet-point summaries into contiguous sequences. Such processing steps negatively affected the coherence of examples. Similar trends in human studies of reference summaries were reported by Stiennon et al. (2020). Examples of noisy reference summaries are shown in Table 9.1b.

Table 9.4 show scores for model outputs across all automatic evaluation metrics. Parameters of metrics used in this study can be found in the evaluation toolkit repository listed in Section 9.1. The results align with insights coming from the human evaluation of models. We found that for most metrics, the highest scores were assigned to large models pretrained on vast quantities of data. However, several metrics, such as $\boldsymbol{S^3}$, SummaQA, SMS, CHRF, and METEOR tended to favor extractive models, assigning the highest scores to their outputs.

Method	ROUGE-1/2/3/4/L/su*/w	ROUGE-WE-(1/2/3)	S^3 (pyr/resp)	BertScore	MoverScore	SummaQA	SMS	BLANC	SUPERT
Extractive Models									
M0 - **LEAD-3**	0.3994 / 0.1746 / 0.0990 / 0.0647 / 0.3606 / 0.1377 / 0.2072	0.4049 / 0.2260 / 0.2172	**0.5395 / 0.6328**	0.3742	0.1679	**0.1652**	0.1050	0.0480	**0.7259**
M1 - **NEUSUM**	0.4130 / 0.1893 / 0.1109 / 0.0742 / 0.3768 / 0.1495 / 0.2156	0.4186 / 0.2402 / 0.2310	**0.5562 / 0.6509**	0.3955	0.1839	**0.1700**	**0.1062**	**0.1087**	0.7010
M2 - **BanditSum**	0.4137 / 0.1868 / 0.1086 / 0.0721 / 0.3759 / 0.1513 / 0.2139	0.4195 / 0.2385 / 0.2300	**0.5339** / 0.6306	0.3938	0.1815	0.1324	**0.1058**	0.0909	**0.7018**
M3 - **LATENT**	0.4136 / 0.1867 / 0.1085 / 0.0721 / 0.3757 / 0.1512 / 0.2138	0.4194 / 0.2384 / 0.2299	0.5337 / 0.6305	0.3936	0.1814	**0.1645**	0.1058	0.0910	**0.7020**
M4 - **REFRESH**	0.3972 / 0.1807 / 0.1042 / 0.0690 / 0.3621 / 0.1340 / 0.2129	0.4023 / 0.2318 / 0.2238	**0.6395 / 0.7124**	0.3903	0.1720	**0.1944**	**0.1088**	**0.1406**	**0.7526**
M5 - **RNES**	0.4088 / 0.1878 / 0.1102 / 0.0736 / 0.3719 / 0.1446/ **0.2163**	0.4153 / 0.2395 / 0.2317	**0.6082 / 0.6894**	**0.3997**	0.1802	**0.1794**	**0.1107**	**0.1232**	**0.7434**
M6 - **JECS**	0.4144 / 0.1846 / 0.1063 / 0.0699 / 0.3760 / 0.1485 / 0.2135	0.4200 / 0.2371 / 0.2283	0.5337 / 0.6284	0.3925	0.1805	0.1644	0.1048	**0.1044**	0.6946
M7 - **STRASS**	0.3377 / 0.1237 / 0.0650 / 0.0416 / 0.2790 / 0.1052 / 0.1559	0.3477 / 0.1757 / 0.1656	0.3632 / 0.4939	0.3090	0.1079	0.1367	0.1023	**0.1042**	0.6566
Abstractive Models									
M8 - **Pointer Generator**	0.3921 / 0.1723 / 0.1003 / 0.0674 / 0.3599 / 0.1435 / 0.1999	0.3990 / 0.2226 / 0.2128	0.4328 / 0.5561	0.3763	0.1643	0.1398	0.0974	0.0704	0.6501
M9 - **Fast-abs-rl**	0.4057 / 0.1774 / 0.0975 / 0.0616 / 0.3806 / 0.1439 / 0.2112	0.4123 / 0.2302 / 0.2184	0.4818 / 0.5865	0.3918	0.1748	0.1431	0.0847	0.0855	0.6125
M10 - **Bottom-Up**	0.4124 / 0.1870 / 0.1064 / 0.0695 / **0.3815** / 0.1543 / 0.2084	0.4192 / 0.2400 / 0.2313	0.4450 / 0.5655	0.3964	0.1830	0.1408	0.0925	0.0570	0.6092
M11 - **Improve-abs**	0.3985 / 0.1720 / 0.0927 / 0.0567 / 0.3730 / 0.1431 / 0.2073	0.4045 / 0.2300 / 0.2228	0.4899 / 0.5897	0.3826	0.1652	0.1341	0.0816	0.0777	0.5972
M12 - **Unified-ext-abs**	0.4038 / 0.1790 / 0.1039 / 0.0695 / 0.3675 / 0.1484 / 0.2074	0.4097 / 0.2299 / 0.2204	0.4936 / 0.5995	0.3832	0.1739	0.1530	0.1038	0.0962	0.6826
M13 - **ROUGESal**	0.4016 / 0.1797 / 0.1053 / 0.0709 / 0.3679 / 0.1497 / 0.2058	0.4078 / 0.2294 / 0.2190	0.4643 / 0.5799	0.3837	0.1722	0.1475	0.1009	0.0882	0.6570
M14 - **Multi-task (Ent + QG)**	0.3952 / 0.1758 / 0.1037 / 0.0705 / 0.3625 / 0.1476 / 0.2007	0.4015 / 0.2253 / 0.2149	0.4246 / 0.5513	0.3759	0.1670	0.1360	0.0982	0.0648	0.6380
M15 - **Closed book decoder**	0.3976 / 0.1760 / 0.1031 / 0.0696 / 0.3636 / 0.1472 / 0.2033	0.4039 / 0.2263 / 0.2160	0.4591 / 0.5757	0.3783	0.1699	0.1456	0.1009	0.0896	0.6612
M16 - **SENECA**	0.4151 / 0.1836 / 0.1052 / 0.0681 / 0.3806 / 0.1520 / 0.2112	0.4211 / 0.2369 / 0.2282	0.4735 / 0.5836	0.3907	0.1811	0.1404	0.1005	0.0692	0.6519
M17 - **T5**	**0.4479 / 0.2205 / 0.1336 / 0.0920 / 0.4172 / 0.1879 / 0.2291**	**0.4543 / 0.2723 / 0.2631**	0.5168 / 0.6294	**0.4450**	**0.2376**	0.1437	0.1046	0.0773	0.6094
M18 - **NeuralTD**	0.4004 / 0.1762 / 0.1000 / 0.0650 / 0.3723 / 0.1452 / 0.2085	0.4063 / 0.2277 / 0.2187	0.4946 / 0.5975	0.3949	0.1697	0.1440	0.0916	0.0859	0.6290
M19 - **BertSum-abs**	**0.4163 / 0.1944 / 0.1156 / 0.0785** / 0.3554 / **0.1625** / 0.1979	**0.4230 / 0.2454 / 0.2351**	0.4664 / 0.5855	0.3855	**0.1894**	0.1385	**0.1071**	0.0815	0.6116
M20 - **GPT-2** (supervised)	0.3981 / 0.1758 / 0.0993 / 0.0649 / 0.3674 / 0.1470 / 0.2006	0.4048 / 0.2268 / 0.2170	0.4069 / 0.5373	0.3915	0.1750	0.1299	0.0930	0.0705	0.6053
M21 - **UniLM**	**0.4306 / 0.2044 / 0.1218 / 0.0824 / 0.4013 / 0.1714 / 0.2228**	**0.4369 / 0.2567 / 0.2483**	0.5143 / 0.6210	**0.4122**	**0.2112**	0.1455	0.0957	0.0841	0.6100
M22 - **BART**	**0.4416 / 0.2128 / 0.1285 / 0.0880 / 0.4100 / 0.1818 / 0.2266**	**0.4472 / 0.2646 / 0.2556**	0.5116 / 0.6215	**0.4264**	**0.2259**	0.1457	0.1037	0.0822	0.6184
M23 - **Pegasus** (dynamic mix)	**0.4407 / 0.2155 / 0.1307 / 0.0901 / 0.4101 / 0.1825 / 0.2260**	**0.4471 / 0.2668 / 0.2575**	0.5099 / **0.6233**	**0.4369**	**0.2283**	0.1422	0.1040	0.0797	0.6046
M23 - **Pegasus** (huge news)	**0.4408 / 0.2147 / 0.1295 / 0.0889 / 0.4103 / 0.1821 / 0.2273**	**0.4473 / 0.2663 / 0.2568**	0.5295 / **0.6372**	**0.4377**	**0.2286**	0.1497	0.1049	0.0845	0.6148

(a) Model scores from summarization-specific evaluation metrics.

Method	BLEU	CHRF	CIDEr	METEOR	Length	Stats (cov/comp/den)	Repeated (1/2/3)
Extractive Models							
M0 - **LEAD-3**	11.4270	0.3892	0.2125	**0.2141**	87.4475	0.9825 / 9.6262 / 57.8001	0.2086 / 0.0310 / 0.0310
M1 - **NEUSUM**	12.7784	0.3946	0.2832	**0.2183**	84.4075	0.9819 / 9.8047 / 32.8574	0.2325 / 0.0531 / 0.0531
M2 - **BanditSum**	12.9761	0.3897	0.3305	0.2124	78.5279	0.9836 / 10.2810 / 40.4265	0.2384 / 0.0573 / 0.0573
M3 - **LATENT**	12.9725	0.3897	0.3305	0.2123	78.5279	0.9834 / 10.2809 / 40.4095	0.2384 / 0.0573 / 0.0573
M4 - **REFRESH**	10.6568	**0.4526**	0.0677	**0.2395**	114.5684	0.9850 / 7.1059 / 53.1928	0.2127 / 0.0289 / 0.0289
M5 - **RNES**	11.2203	**0.4062**	0.1559	**0.2300**	99.9199	0.9938 / 7.9032 / 67.7089	0.2451 / 0.0540 / 0.0540
M6 - **JECS**	12.5659	**0.4310**	0.3090	0.2122	79.7797	0.9874 / 10.1111 / 26.6943	0.2041 / 0.0327 / 0.0327
M7 - **STRASS**	7.8330	0.3330	0.2945	0.1607	76.4859	0.9969 / 12.7835 / 59.9498	0.1864 / 0.0343 / 0.0343
Abstractive Models							
M8 - **Pointer Generator**	13.8247	0.3567	0.5065	0.1860	63.5211	0.9957 / 13.1940 / 26.0880	0.2015 / 0.0375 / 0.0375
M9 - **Fast-abs-rl**	12.9812	0.3778	0.4329	0.2014	70.8600	0.9860 / 11.0141 / **9.9859**	0.2157 / 0.0370 / 0.0370
M10 - **Bottom-Up**	**15.1293**	0.3523	**0.6176**	0.1887	**56.5715**	0.9811 / **14.7771** / 12.6181	**0.1856 / 0.0211 / 0.0211**
M11 - **Improve-abs**	11.9816	0.3715	0.3356	0.2005	75.9512	**0.9674** / 10.6043 / **8.9755**	0.2499 / 0.0542 / 0.0542
M12 - **Unified-ext-abs**	12.8457	0.3786	0.3851	0.2017	74.4663	0.9868 / 10.7510 / 33.1106	0.2177 / 0.0493 / 0.0493
M13 - **ROUGESal**	13.8882	0.3668	0.4746	0.1936	66.5575	0.9853 / 13.0369 / 25.2893	0.2102 / 0.0458 / 0.0458
M14 - **Multi-task (Ent + QG)**	14.5276	0.3539	0.5749	0.1831	**60.0294**	0.9853 / **14.1828** / 22.2296	0.1985 / 0.0411 / 0.0411
M15 - **Closed book decoder**	13.4158	0.3675	0.4648	0.1925	68.2858	0.9866 / 12.0588 / 27.3686	0.2074 / 0.0444 / 0.0444
M16 - **SENECA**	13.7676	0.3660	0.5233	0.1966	64.9710	0.9880 / 12.3610 / 16.7640	0.2146 / 0.0303 / 0.0303
M17 - **T5**	**19.3891**	0.3833	**0.7763**	0.2140	**59.5288**	**0.9775 / 14.2002** / 12.9565	**0.1810 / 0.0209 / 0.0209**
M18 - **NeuralTD**	12.9241	0.3783	0.3543	0.2038	74.4033	0.9830 / 10.7768 / **12.4443**	0.2645 / 0.0901 / 0.0901
M19 - **BertSum-abs**	14.9525	0.3649	**0.6240**	0.1876	**60.8893**	**0.9517 / 13.9197 / 12.3254**	**0.1697 / 0.0156 / 0.0156**
M20 - **GPT-2** (supervised)	13.9364	0.3678	0.5787	0.1759	**51.8352**	0.9791 / **15.9839** / 15.4999	0.1875 / 0.0362 / 0.0362
M21 - **UniLM**	**15.5736**	**0.4230**	0.5294	0.2084	67.1960	**0.9685** / 11.5672 / **11.7908**	**0.1722 / 0.0180 / 0.0180**
M22 - **BART**	**17.1005**	**0.4271**	**0.7573**	0.2105	62.2989	**0.9771** / 12.8811 / 15.2999	**0.1627 / 0.0127 / 0.0127**
M23 - **Pegasus** (dynamic mix)	**18.6517**	0.4261	**0.7280**	**0.2131**	64.1348	0.9438 / 13.7208 / 11.6003	0.1855 / 0.0355 / 0.0081
M23 - **Pegasus** (huge news)	**17.8102**	0.3912	**0.6595**	**0.2189**	66.7559	0.9814/12.9473/14.9850	0.1883/0.0251/0.0251

(b) Model scores from other text generation evaluation metrics.

Table 9.4: Model scores from automatic evaluation metrics available in the evaluation toolkit. The five highest scores for each metric (and lowest for Length and Repeated-1/2/3) are bolded.

Presented results provide a comprehensive perspective on the current state of the field and highlight directions for future modeling work.

9.7 Summary

We introduced SummEval, a set of resources for summarization model and evaluation research that include: a collection of summaries generated by recent summarization models on the CNNDM dataset, an extensible and unified toolkit for summarization model evaluation, and a diverse collection of human annotations of model outputs collected from the crowd-source and expert annotators. Using the accumulated resources we re-evaluated a broad selection of current models and evaluation metrics in a consistent and comprehensive manner. We hope that this work will prove to be a valuable resource for future research on text summarization evaluation and models. We also encourage the research community to join our efforts by contributing model outputs and extending the evaluation toolkit with new metrics.

Chapter 10

Conclusion and Future Work

In this thesis, we address the increasingly important task of automatic text summarization. Specifically, we divide our work into settings where large-scale data is available and those where it is not, as well as an evaluation of the current state of summarization research.

In particular, Chapter 3, Chapter 5, and Chapter 4 introduce several datasets and modeling techniques applicable for training and evaluating multi-document summarization models. In Chapter 3, we introduce TutorialBank, a new, publicly available dataset that aims to facilitate NLP education and research which motivates the task of survey generation, among other applications. In 4, we introduce the first large-scale multi-document summarization datasets in the news domain. Furthermore, we present a novel model for reducing redundancy in multi-document summarization. While we see the abilities of these models in large-scale data settings, Chapter 5 focuses on understanding how a simple pipeline consisting of state-of-the-art pretrained language models that shows large improvements in one task fails to generalize to a real-world setting.

Although the models trained on large-scale datasets are not always ideal, in many settings, they have achieved state-of-the-art performance, raising the question of how far we can push these models when such data is not available in certain domains. This question is addressed in Chapter 6, Chapter 7, and Chapter 8. In Chapter 6, we introduce

a retrieval, template-based framework which achieves state-of-the-art results on SQuAD for unsupervised models, particularly when the answer is a named entity. In Chapter 7, we introduce a dataset generation pipeline for multi-answer summarization, where such data was lacking. Importantly, we introduce and evaluate RL reward functions on answer summarization, including entailment as a measure of faithfulness and volume of semantic space as a way to increase answer coverage. In Chapter 8 we focus on improving zero-shot and few-shot transfer abilities of summarization models across domains, introducing the WikiTransfer method to create pseudo-summaries with subaspects of the target dataset which can be used as unlabeled data for intermediate fine-tuning. We show that this method improves zero-shot domain transfer over transfer from other domains.

Finally, in Chapter 9, we take stock of the current evaluation protocol, showing that pretrained models have advanced the state-of-the-art across automatic and human evaluations. Furthermore, we point to areas of improvement in current models for coherence and relevance and the necessity of further evaluation metrics in more abstractive datasets.

10.1 Future Work

Despite the tremendous progress in summarization, much work remains to make such systems viable for real-world applications. Building upon my past work, I plan to further explore the following directions.

Evaluation of Summarization Faithfulness: For summarization models to be used in production environments, they must remain faithful to the source text. Part of the challenge in this direction is properly evaluating the faithfulness of summary model outputs. While we have made some progress in this direction, work remains, especially in judging the faithfulness of very abstractive text. Furthermore, the community at large must address what it means for a text to be faithful. For example, text may not directly be stated in the source but may be very plausible, such as the summary of subjective answers on a

community question-answering platform. As a whole, the community must properly define what our end-goal is for determining faithfulness. Furthermore, faithfulness scores often do not correlate strongly with other evaluation metrics such as ROUGE. However, papers are typically published due to the shown increase in ROUGE performance. We must standardize this evaluation and quantify the degree to which a decrease in ROUGE performance is allowable for insurance of faithfulness.

Controllable and Personalizable Text Summarization: Summaries should be controlled for faithfulness and be personalized for a given context. Part of the problem of current abstractive summarization models is a lack of control over the content selection and realization stages. I believe that tying in the two, either through a two-step pipeline or through additional loss functions as in the span-prediction discussed in Chapter 7, require further examination. When such methods are controllable, they also offer interpretability of knowing where the summary text comes from and why it was chosen, which can increase trust in such models as they make their way into production environments. Furthermore, summaries should be personalizable to a particular audience. A summary of a scientific article should look different depending on whether the reader is an academic or a layperson. However, most work in summarization has assumed a monolithic audience, which will not serve the diverse audience which such technology can reach.

Real-world Domain-transfer Summarization: We have only touched the surface in terms of applying data-efficient models in new domains. We argued that subaspects are most important for transfer to new domains, but we would like to also take advantage of unsupervised in-domain data. Furthermore, lexical subaspects were used, while we want to explore deeper semantic and stylistic variations from domain to domain and encode other stylistic aspects such as sentence structure. Ultimately, such methods should work in zero-shot settings without any training examples but with the help of in-domain data to provide semantic and stylistic guidelines.

Bibliography

Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. Synthetic QA corpora generation with roundtrip consistency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6168–6173, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1620. URL https://www.aclweb.org/anthology/P19-1620.

Stefanos Angelidis and Mirella Lapata. 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, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1403. URL https://www.aclweb.org/anthology/D18-1403.

Alexei Baevski and Michael Auli. Adaptive input representations for neural language modeling. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=ByxZX20qFQ.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1409.0473.

Siddhartha Banerjee and Prasenjit Mitra. Wikiwrite: Generating wikipedia articles automatically. In Subbarao Kambhampati, editor, *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 2740–2746. IJCAI/AAAI Press, 2016. URL http://www.ijcai.org/Abstract/16/389.

Regina Barzilay, Kathleen R. McKeown, and Michael Elhadad. Information fusion in the context of multi-document summarization. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 550–557, College Park, Maryland, USA, June 1999. Association for Computational Linguistics. doi: 10.3115/1034678.1034760. URL https://www.aclweb.org/anthology/P99-1071.

Tal Baumel, Matan Eyal, and Michael Elhadad. Query Focused Abstractive Summarization: Incorporating Query Relevance, Multi-Document Coverage, and Summary Length Constraints into seq2seq Models. *CoRR*, abs/1801.07704, 2018.

Christos Baziotis, Ion Androutsopoulos, Ioannis Konstas, and Alexandros Potamianos. SEQˆ3: Differentiable sequence-to-sequence-to-sequence autoencoder for unsupervised abstractive sentence compression. 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 673–681, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1071. URL https://www.aclweb.org/anthology/N19-1071.

Steven Bird, Robert Dale, Bonnie Dorr, Bryan Gibson, Mark Joseph, Min-Yen Kan, Dongwon Lee, Brett Powley, Dragomir Radev, and Yee Fan Tan. The ACL Anthology reference corpus: A reference dataset for bibliographic research in computational linguistics. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, May 2008. European Language Resources Associa-

tion (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2008/pdf/445_paper.pdf.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. In Thomas G. Dietterich, Suzanna Becker, and Zoubin Ghahramani, editors, *Advances in Neural Information Processing Systems 14 [Neural Information Processing Systems: Natural and Synthetic, NIPS 2001, December 3-8, 2001, Vancouver, British Columbia, Canada]*, pages 601–608. MIT Press, 2001. URL https://proceedings.neurips.cc/paper/2001/hash/296472c9542ad4d4788d543508116cbc-Abstract.html.

Florian Böhm, Yang Gao, Christian M. Meyer, Ori Shapira, Ido Dagan, and Iryna Gurevych. Better rewards yield better summaries: Learning to summarise without references. 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 3110–3120, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1307. URL https://www.aclweb.org/anthology/D19-1307.

Léo Bouscarrat, Antoine Bonnefoy, Thomas Peel, and Cécile Pereira. STRASS: A light and effective method for extractive summarization based on sentence embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 243–252, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-2034. URL https://www.aclweb.org/anthology/P19-2034.

Arthur Bražinskas, Mirella Lapata, and Ivan Titov. Few-shot learning for abstractive multi-document opinion summarization. In *EMNLP*, 2020a.

Arthur Bražinskas, Mirella Lapata, and Ivan Titov. Unsupervised opinion summarization as

copycat-review generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5151–5169, Online, July 2020b. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.461. URL https://www.aclweb.org/anthology/2020.acl-main.461.

Ziqiang Cao, Wenjie Li, Sujian Li, and Furu Wei. Improving multi-document summarization via text classification. In Satinder P. Singh and Shaul Markovitch, editors, *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 3053–3059. AAAI Press, 2017. URL http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14525.

Jaime Carbonell and Jade Goldstein. The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 335–336. ACM, 1998.

Asli Celikyilmaz, Antoine Bosselut, Xiaodong He, and Yejin Choi. Deep communicating agents for abstractive summarization. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1662–1675, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1150. URL https://www.aclweb.org/anthology/N18-1150.

Arun Chaganty, Stephen Mussmann, and Percy Liang. The price of debiasing automatic metrics in natural language evalaution. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 643–653, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1060. URL https://www.aclweb.org/anthology/P18-1060.

Wen Chan, Xiangdong Zhou, Wei Wang, and Tat-Seng Chua. Community answer summa-

rization for multi-sentence question with group L1 regularization. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 582–591, Jeju Island, Korea, July 2012. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P12-1061.

Yan-Min Chen, Xiao-Long Wang, and Bing-Quan Liu. Multi-document summarization based on lexical chains. In *2005 international conference on machine learning and cybernetics*, volume 3, pages 1937–1942. IEEE, 2005.

Yen-Chun Chen and Mohit Bansal. 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, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1063. URL https://www.aclweb.org/anthology/P18-1063.

Jianpeng Cheng and Mirella Lapata. 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, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1046. URL https://www.aclweb.org/anthology/P16-1046.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1179. URL https://www.aclweb.org/anthology/D14-1179.

Sumit Chopra, Michael Auli, and Alexander M. Rush. 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, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1012. URL https://www.aclweb.org/anthology/N16-1012.

Tanya Chowdhury and Tanmoy Chakraborty. Cqasumm: Building references for community question answering summarization corpora. In *Proceedings of the ACM India Joint International Conference on Data Science and Management of Data*, pages 18–26, 2019.

Tanya Chowdhury, Sachin Kumar, and Tanmoy Chakraborty. Neural abstractive summarization with structural attention. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 3716–3722. ijcai.org, 2020. doi: 10.24963/ijcai.2020/514. URL https://doi.org/10.24963/ijcai.2020/514.

Janara Christensen, Mausam, Stephen Soderland, and Oren Etzioni. Towards coherent multi-document summarization. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1163–1173, Atlanta, Georgia, June 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/N13-1136.

Eric Chu and Peter J. Liu. Meansum: A neural model for unsupervised multi-document abstractive summarization. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 1223–1232. PMLR, 2019a. URL http://proceedings.mlr.press/v97/chu19b.html.

Eric Chu and Peter J. Liu. Meansum: A neural model for unsupervised multi-document abstractive summarization. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors,

Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA, volume 97 of *Proceedings of Machine Learning Research*, pages 1223–1232. PMLR, 2019b. URL http://proceedings.mlr.press/v97/chu19b.html.

Yu-An Chung, Hung-Yi Lee, and James Glass. Supervised and unsupervised transfer learning for question answering. 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 1585–1594, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1143. URL https://www.aclweb.org/anthology/N18-1143.

Elizabeth Clark, Asli Celikyilmaz, and Noah A. Smith. Sentence mover's similarity: Automatic evaluation for multi-sentence texts. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2748–2760, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1264. URL https://www.aclweb.org/anthology/P19-1264.

Arman Cohan and Nazli Goharian. Revisiting summarization evaluation for scientific articles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 806–813, Portorož, Slovenia, May 2016. European Language Resources Association (ELRA). URL https://www.aclweb.org/anthology/L16-1130.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 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, June

2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2097. URL https://www.aclweb.org/anthology/N18-2097.

Jacob Cohen. A Coefficient of Agreement for Nominal Scales. *Educational and psychological measurement*, 20(1):37–46, 1960.

Alexis Conneau and Guillaume Lample. Cross-lingual language model pretraining. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 7057–7067, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/c04c19c2c2474dbf5f7ac4372c5b9af1-Abstract.html.

John M Conroy and Dianne P O'leary. Text summarization via hidden markov models. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 406–407, 2001.

Zihang Dai, Zhilin Yang, Yiming Yang, William W. Cohen, Jaime Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. Transformer-XL: Language modeling with longer-term dependency, 2019.

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

Hoa Trang Dang and Karolina Owczarzak. Overview of the tac 2008 update summarization task. In *TAC*, 2008.

Hoa Trang Dang and Karolina Owczarzak. Overview of the tac 2009 summarization track. In *proceedings of the Text Analysis Conference*, 2009.

Hal Daumé III and Daniel Marcu. Bayesian query-focused summarization. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 305–312, Sydney, Australia, July 2006. Association for Computational Linguistics. doi: 10.3115/1220175.1220214. URL https://www.aclweb.org/anthology/P06-1039.

Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407, 1990.

Yang Deng, Wai Lam, Yuexiang Xie, Daoyuan Chen, Yaliang Li, Min Yang, and Ying Shen. Joint learning of answer selection and answer summary generation in community question answering. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7651–7658. AAAI Press, 2020. URL https://aaai.org/ojs/index.php/AAAI/article/view/6266.

Franck Dernoncourt, Mohammad Ghassemi, and Walter Chang. A repository of corpora for summarization. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 2018. European Language Resources Association (ELRA). URL https://www.aclweb.org/anthology/L18-1509.

Shrey Desai, Jiacheng Xu, and Greg Durrett. 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, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.507. URL https://www.aclweb.org/anthology/2020.emnlp-main.507.

Daniel Deutsch and Dan Roth. Summary cloze: A new task for content selection in topic-focused summarization. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3720–3729, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1386. URL https://www.aclweb.org/anthology/D19-1386.

Daniel Deutsch and Dan Roth. SacreROUGE: An open-source library for using and developing summarization evaluation metrics. In *Proceedings of Second Workshop for NLP Open Source Software (NLP-OSS)*, pages 120–125, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.nlposs-1.17. URL https://www.aclweb.org/anthology/2020.nlposs-1.17.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://www.aclweb.org/anthology/N19-1423.

Bhuwan Dhingra, Danish Danish, and Dheeraj Rajagopal. Simple and effective semi-supervised question answering. 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 582–587, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2092. URL https://www.aclweb.org/anthology/N18-2092.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unified language model pre-training for natural language understanding and generation. In Hanna M. Wallach, Hugo Larochelle, Alina

Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 13042–13054, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/c20bb2d9a50d5ac1f713f8b34d9aac5a-Abstract.html.

Yue Dong, Yikang Shen, Eric Crawford, Herke van Hoof, and Jackie Chi Kit Cheung. BanditSum: Extractive summarization as a contextual bandit. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3739–3748, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1409. URL https://www.aclweb.org/anthology/D18-1409.

Zi-Yi Dou, Pengfei Liu, Hiroaki Hayashi, Zhengbao Jiang, and Graham Neubig. Gsum: A general framework for guided neural abstractive summarization. *arXiv preprint arXiv:2010.08014*, 2020.

Esin Durmus, He He, and Mona Diab. FEQA: A question answering evaluation framework for faithfulness assessment in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5055–5070, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.454. URL https://www.aclweb.org/anthology/2020.acl-main.454.

Harold P Edmundson. New methods in automatic extracting. *Journal of the ACM (JACM)*, 16(2):264–285, 1969.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1045. URL https://www.aclweb.org/anthology/D18-1045.

Günes Erkan and Dragomir R Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479, 2004.

Alexander R Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. Summeval: Re-evaluating summarization evaluation. *arXiv preprint arXiv:2007.12626*, 2020.

Tobias Falke, Leonardo F. R. Ribeiro, Prasetya Ajie Utama, Ido Dagan, and Iryna Gurevych. Ranking generated summaries by correctness: An interesting but challenging application for natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2214–2220, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1213. URL https://www.aclweb.org/anthology/P19-1213.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. ELI5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1346. URL https://www.aclweb.org/anthology/P19-1346.

Joseph L. Fleiss, Bruce Levin, and Myunghee Cho Paik. *The Measurement of Interrater Agreement*, pages 598–626. John Wiley & Sons, Inc., 2004. ISBN 9780471445425. doi: 10.1002/0471445428.ch18. URL http://dx.doi.org/10.1002/0471445428.ch18.

Kavita Ganesan. Rouge 2.0: Updated and improved measures for evaluation of summarization tasks, 2015.

Kavita Ganesan, ChengXiang Zhai, and Jiawei Han. Opinosis: A graph based approach to abstractive summarization of highly redundant opinions. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 340–348,

Beijing, China, August 2010. Coling 2010 Organizing Committee. URL https://www.aclweb.org/anthology/C10-1039.

Yang Gao, Wei Zhao, and Steffen Eger. SUPERT: Towards new frontiers in unsupervised evaluation metrics for multi-document summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1347–1354, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.124. URL https://www.aclweb.org/anthology/2020.acl-main.124.

Siddhant Garg, Thuy Vu, and Alessandro Moschitti. TANDA: transfer and adapt pre-trained transformer models for answer sentence selection. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7780–7788. AAAI Press, 2020. URL https://aaai.org/ojs/index.php/AAAI/article/view/6282.

Sebastian Gehrmann, Yuntian Deng, and Alexander Rush. Bottom-up abstractive summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4098–4109, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1443. URL https://www.aclweb.org/anthology/D18-1443.

Dan Gillick and Yang Liu. Non-expert evaluation of summarization systems is risky. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 148–151, Los Angeles, June 2010. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W10-0722.

Yihong Gong and Xin Liu. Generic text summarization using relevance measure and latent

semantic analysis. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 19–25, 2001.

Jonathan Gordon, Linhong Zhu, Aram Galstyan, Prem Natarajan, and Gully Burns. Modeling concept dependencies in a scientific corpus. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 866–875, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1082. URL https://www.aclweb.org/anthology/P16-1082.

Jonathan Gordon, Stephen Aguilar, Emily Sheng, and Gully Burns. Structured generation of technical reading lists. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 261–270, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-5029. URL https://www.aclweb.org/anthology/W17-5029.

Yvette Graham. Re-evaluating automatic summarization with BLEU and 192 shades of ROUGE. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 128–137, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1013. URL https://www.aclweb.org/anthology/D15-1013.

Max Grusky, Mor Naaman, and Yoav Artzi. 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, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1065. URL https://www.aclweb.org/anthology/N18-1065.

Han Guo, Ramakanth Pasunuru, and Mohit Bansal. Soft layer-specific multi-task summarization with entailment and question generation. In *Proceedings of the 56th Annual Meeting*

of the Association for Computational Linguistics (Volume 1: Long Papers), pages 687–697, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1064. URL https://www.aclweb.org/anthology/P18-1064.

Amir Hadifar, Lucas Sterckx, Thomas Demeester, and Chris Develder. A self-training approach for short text clustering. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 194–199, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-4322. URL https://www.aclweb.org/anthology/W19-4322.

Aria Haghighi and Lucy Vanderwende. Exploring content models for multi-document summarization. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 362–370, Boulder, Colorado, June 2009. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/N09-1041.

Hardy, Shashi Narayan, and Andreas Vlachos. HighRES: Highlight-based reference-less evaluation of summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3381–3392, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1330. URL https://www.aclweb.org/anthology/P19-1330.

Helia Hashemi, Mohammad Aliannejadi, Hamed Zamani, and W Bruce Croft. Antique: A non-factoid question answering benchmark. In *European Conference on Information Retrieval*, pages 166–173. Springer, 2020.

Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 28: Annual Conference*

on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada, pages 1693–1701, 2015. URL https://proceedings.neurips.cc/paper/2015/hash/afdec7005cc9f14302cd0474fd0f3c96-Abstract.html.

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

Kai Hong and Ani Nenkova. Improving the estimation of word importance for news multi-document summarization. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 712–721, Gothenburg, Sweden, April 2014. Association for Computational Linguistics. doi: 10.3115/v1/E14-1075. URL https://www.aclweb.org/anthology/E14-1075.

Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1031. URL https://www.aclweb.org/anthology/P18-1031.

Wan-Ting Hsu, Chieh-Kai Lin, Ming-Ying Lee, Kerui Min, Jing Tang, and Min Sun. A unified model for extractive and abstractive summarization using inconsistency loss. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 132–141, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1013. URL https://www.aclweb.org/anthology/P18-1013.

Xinyu Hua and Lu Wang. 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, September 2017. Association for Computa-

tional Linguistics. doi: 10.18653/v1/W17-4513. URL https://www.aclweb.org/anthology/W17-4513.

Luyang Huang, Lingfei Wu, and Lu Wang. Knowledge graph-augmented abstractive summarization with semantic-driven cloze reward. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5094–5107, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.457. URL https://www.aclweb.org/anthology/2020.acl-main.457.

Kokil Jaidka, Muthu Kumar Chandrasekaran, Sajal Rustagi, and Min-Yen Kan. Overview of the CL-SciSumm 2016 shared task. In *Proceedings of the Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL)*, pages 93–102, June 2016. URL https://www.aclweb.org/anthology/W16-1511.

James Gregory Jardine. *Automatically Generating Reading Lists*. PhD thesis, University of Cambridge, UK, 2014.

Rahul Jha, Amjad Abu-Jbara, and Dragomir Radev. 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, August 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P13-2102.

Rahul Jha, Reed Coke, and Dragomir R. Radev. Surveyor: A system for generating coherent survey articles for scientific topics. In Blai Bonet and Sven Koenig, editors, *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*, pages 2167–2173. AAAI Press, 2015. URL http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9855.

Yichen Jiang and Mohit Bansal. Closed-book training to improve summarization encoder

memory. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4067–4077, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1440. URL https://www.aclweb.org/anthology/D18-1440.

Taehee Jung, Dongyeop Kang, Lucas Mentch, and Eduard Hovy. Earlier isn't always better: Sub-aspect analysis on corpus and system biases in summarization. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3324–3335, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1327. URL https://www.aclweb.org/anthology/D19-1327.

Chris Kedzie, Kathleen McKeown, and Hal Daumé III. 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, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1208. URL https://www.aclweb.org/anthology/D18-1208.

Byeongchang Kim, Hyunwoo Kim, and Gunhee Kim. Abstractive summarization of Reddit posts with multi-level memory networks. 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 2519–2531, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1260. URL https://www.aclweb.org/anthology/N19-1260.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

Klaus Krippendorff. Computing krippendorff's alpha-reliability, 2011.

Wojciech Kryściński, Romain Paulus, Caiming Xiong, and Richard Socher. Improving abstraction in text summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1808–1817, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1207. URL https://www.aclweb.org/anthology/D18-1207.

Wojciech Kryscinski, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. 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, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1051. URL https://www.aclweb.org/anthology/D19-1051.

Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. 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, November 2020a. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.750. URL https://www.aclweb.org/anthology/2020.emnlp-main.750.

Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. 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, November 2020b. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.750. URL https://www.aclweb.org/anthology/2020.emnlp-main.750.

Matt J. Kusner, Yu Sun, Nicholas I. Kolkin, and Kilian Q. Weinberger. From word embed-

dings to document distances. In Francis R. Bach and David M. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 957–966. JMLR.org, 2015. URL http://proceedings.mlr.press/v37/kusnerb15.html.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7: 452–466, March 2019. doi: 10.1162/tacl_a_00276. URL https://www.aclweb.org/anthology/Q19-1026.

Philippe Laban, Andrew Hsi, John Canny, and Marti A. Hearst. The summary loop: Learning to write abstractive summaries without examples. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5135–5150, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.460. URL https://www.aclweb.org/anthology/2020.acl-main.460.

J Richard Landis and Gary G Koch. The Measurement of Observer Agreement for Categorical Data. *Biometrics*, pages 159–174, 1977.

Alon Lavie and Abhaya Agarwal. METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 228–231, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W07-0734.

Quoc V. Le and Tomás Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014,*

Beijing, China, 21-26 June 2014, volume 32 of *JMLR Workshop and Conference Proceedings*, pages 1188–1196. JMLR.org, 2014. URL http://proceedings.mlr.press/v32/le14.html.

Logan Lebanoff, Kaiqiang Song, and Fei Liu. 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, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1446. URL https://www.aclweb.org/anthology/D18-1446.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 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, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.703. URL https://www.aclweb.org/anthology/2020.acl-main.703.

Patrick Lewis, Ludovic Denoyer, and Sebastian Riedel. Unsupervised question answering by cloze translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4896–4910, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1484. URL https://www.aclweb.org/anthology/P19-1484.

Haoran Li, Junnan Zhu, Jiajun Zhang, and Chengqing Zong. Ensure the correctness of the summary: Incorporate entailment knowledge into abstractive sentence summarization. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1430–1441, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/C18-1121.

Irene Li, Alexander R. Fabbri, Robert R. Tung, and Dragomir R. Radev. What should I learn first: Introducing lecturebank for NLP education and prerequisite chain learning. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6674–6681. AAAI Press, 2019. doi: 10.1609/aaai.v33i01.33016674. URL https://doi.org/10.1609/aaai.v33i01.33016674.

Chen Liang, Jianbo Ye, Zhaohui Wu, Bart Pursel, and C. Lee Giles. Recovering concept prerequisite relations from university course dependencies. In Satinder P. Singh and Shaul Markovitch, editors, *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 4786–4791. AAAI Press, 2017. URL http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14654.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004a. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W04-1013.

Chin-Yew Lin. Looking for a few good metrics: Automatic summarization evaluation-how many samples are enough? In *NTCIR*, 2004b.

Hui Lin and Jeff Bilmes. Multi-document summarization via budgeted maximization of submodular functions. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 912–920, Los Angeles, California, June 2010. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/N10-1134.

Feifan Liu and Yang Liu. Correlation between ROUGE and human evaluation of extractive

meeting summaries. In *Proceedings of ACL-08: HLT, Short Papers*, pages 201–204, Columbus, Ohio, June 2008. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P08-2051.

Hanxiao Liu, Wanli Ma, Yiming Yang, and Jaime G. Carbonell. Learning Concept Graphs from Online Educational Data. *J. Artif. Intell. Res.*, 55:1059–1090, 2016.

Peter J. Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. Generating wikipedia by summarizing long sequences. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=Hyg0vbWC-.

Yang Liu and Mirella Lapata. 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, November 2019a. Association for Computational Linguistics. doi: 10.18653/v1/D19-1387. URL https://www.aclweb.org/anthology/D19-1387.

Yang Liu and Mirella Lapata. Hierarchical transformers for multi-document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5070–5081, Florence, Italy, July 2019b. Association for Computational Linguistics. doi: 10.18653/v1/P19-1500. URL https://www.aclweb.org/anthology/P19-1500.

Yang Liu and Mirella Lapata. 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, November 2019c. Asso-

ciation for Computational Linguistics. doi: 10.18653/v1/D19-1387. URL https://www.aclweb.org/anthology/D19-1387.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A Robustly Optimized Bert Pretraining Approach. *arXiv preprint arXiv:1907.11692*, 2019.

Yuanjie Liu, Shasha Li, Yunbo Cao, Chin-Yew Lin, Dingyi Han, and Yong Yu. Understanding and summarizing answers in community-based question answering services. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 497–504, Manchester, UK, August 2008. Coling 2008 Organizing Committee. URL https://www.aclweb.org/anthology/C08-1063.

Annie Louis and Ani Nenkova. Automatically assessing machine summary content without a gold standard. *Computational Linguistics*, 39(2):267–300, 2013. doi: 10.1162/COLI_a_00123. URL https://www.aclweb.org/anthology/J13-2002.

Yao Lu, Linqing Liu, Zhile Jiang, Min Yang, and Randy Goebel. A multi-task learning framework for abstractive text summarization. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 9987–9988. AAAI Press, 2019. doi: 10.1609/aaai.v33i01.33019987. URL https://doi.org/10.1609/aaai.v33i01.33019987.

Hans Peter Luhn. The automatic creation of literature abstracts. *IBM Journal of research and development*, 2(2):159–165, 1958.

Kazuki Matsumaru, Sho Takase, and Naoaki Okazaki. Improving truthfulness of headline generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1335–1346, Online, July 2020. Association for Computational

Linguistics. doi: 10.18653/v1/2020.acl-main.123. URL https://www.aclweb.org/anthology/2020.acl-main.123.

Mani Maybury. *Advances in automatic text summarization*. MIT press, 1999.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.173. URL https://www.aclweb.org/anthology/2020.acl-main.173.

Ryan McDonald. A study of global inference algorithms in multi-document summarization. In *European Conference on Information Retrieval*, pages 557–564. Springer, 2007.

Kathleen R. McKeown and Dragomir R. Radev. Generating summaries of multiple news articles. In *Proceedings, ACM Conference on Research and Development in Information Retrieval SIGIR'95*, pages 74–82, Seattle, Washington, July 1995.

Rada Mihalcea and Paul Tarau. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W04-3252.

Tomas Mikolov, Martin Karafiát, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur. Recurrent neural network based language model. In Takao Kobayashi, Keikichi Hirose, and Satoshi Nakamura, editors, *INTERSPEECH*, pages 1045–1048. ISCA, 2010. URL http://dblp.uni-trier.de/db/conf/interspeech/interspeech2010.html#MikolovKBCK10.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. *CoRR*, abs/1301.3781, 2013a.

Tomás Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In Christopher J. C. Burges, Léon Bottou, Zoubin Ghahramani, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 3111–3119, 2013b. URL https://proceedings.neurips.cc/paper/2013/hash/9aa42b31882ec039965f3c4923ce901b-Abstract.html.

Daniel Müllner. Modern hierarchical, agglomerative clustering algorithms. *arXiv preprint arXiv:1109.2378*, 2011.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. 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, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/K16-1028. URL https://www.aclweb.org/anthology/K16-1028.

Courtney Napoles, Matthew Gormley, and Benjamin Van Durme. Annotated Gigaword. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX)*, pages 95–100, Montréal, Canada, June 2012. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W12-3018.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 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, October-November 2018a. Association

for Computational Linguistics. doi: 10.18653/v1/D18-1206. URL https://www.aclweb.org/anthology/D18-1206.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Ranking sentences for extractive summarization with reinforcement learning. 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 1747–1759, New Orleans, Louisiana, June 2018b. Association for Computational Linguistics. doi: 10.18653/v1/N18-1158. URL https://www.aclweb.org/anthology/N18-1158.

Ani Nenkova and Kathleen McKeown. A survey of text summarization techniques. In *Mining text data*, pages 43–76. Springer, 2012.

Jun-Ping Ng and Viktoria Abrecht. Better summarization evaluation with word embeddings for ROUGE. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1925–1930, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1222. URL https://www.aclweb.org/anthology/D15-1222.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. Facebook FAIR's WMT19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 314–319, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-5333. URL https://www.aclweb.org/anthology/W19-5333.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota,

June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-4009. URL https://www.aclweb.org/anthology/N19-4009.

Karolina Owczarzak and Hoa Trang Dang. Overview of the tac 2011 summarization track: Guided task and aesop task. In *Proceedings of the Text Analysis Conference (TAC 2011), Gaithersburg, Maryland, USA, November*, 2011.

Karolina Owczarzak, Peter A. Rankel, Hoa Trang Dang, and John M. Conroy. Assessing the effect of inconsistent assessors on summarization evaluation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 359–362, Jeju Island, Korea, July 2012. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P12-2070.

Liangming Pan, Chengjiang Li, Juanzi Li, and Jie Tang. Prerequisite relation learning for concepts in MOOCs. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1447–1456, Vancouver, Canada, July 2017a. Association for Computational Linguistics. doi: 10.18653/v1/P17-1133. URL https://www.aclweb.org/anthology/P17-1133.

Liangming Pan, Xiaochen Wang, Chengjiang Li, Juanzi Li, and Jie Tang. Course concept extraction in MOOCs via embedding-based graph propagation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 875–884, Taipei, Taiwan, November 2017b. Asian Federation of Natural Language Processing. URL https://www.aclweb.org/anthology/I17-1088.

Vinay Pande, Tanmoy Mukherjee, and Vasudeva Varma. Summarizing answers for community question answer services. In *Language Processing and Knowledge in the Web*, pages 151–161. Springer, 2013.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the*

Association for Computational Linguistics, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL https://www.aclweb.org/anthology/P02-1040.

Ramakanth Pasunuru and Mohit Bansal. Multi-reward reinforced summarization with saliency and entailment. 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 646–653, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2102. URL https://www.aclweb.org/anthology/N18-2102.

Over Paul and Yen James. An Introduction to DUC-2004. In *Proceedings of the 4th Document Understanding Conference (DUC 2004)*, 2004.

Romain Paulus, Caiming Xiong, and Richard Socher. A deep reinforced model for abstractive summarization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=HkAClQgA-.

Laura Perez-Beltrachini, Yang Liu, and Mirella Lapata. 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, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1504. URL https://www.aclweb.org/anthology/P19-1504.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237,

New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1202. URL https://www.aclweb.org/anthology/N18-1202.

Maxime Peyrard. Studying summarization evaluation metrics in the appropriate scoring range. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5093–5100, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1502. URL https://www.aclweb.org/anthology/P19-1502.

Maxime Peyrard, Teresa Botschen, and Iryna Gurevych. Learning to score system summaries for better content selection evaluation. In *Proceedings of the Workshop on New Frontiers in Summarization*, pages 74–84, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-4510. URL https://www.aclweb.org/anthology/W17-4510.

Maja Popović. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/W15-3049. URL https://www.aclweb.org/anthology/W15-3049.

Dragomir R. Radev and Kathleen R. McKeown. Generating natural language summaries from multiple on-line sources. *Computational Linguistics*, 24(3):469–500, 1998. URL https://www.aclweb.org/anthology/J98-3005.

Dragomir R. Radev, Hongyan Jing, and Malgorzata Budzikowska. Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation, and user studies. In *NAACL-ANLP 2000 Workshop: Automatic Summarization*, 2000. URL https://www.aclweb.org/anthology/W00-0403.

Dragomir R. Radev, Pradeep Muthukrishnan, and Vahed Qazvinian. The ACL Anthology

Network Corpus. In *Proceedings, ACL Workshop on Natural Language Processing and Information Retrieval for Digital Libraries*, Singapore, 2009.

Dragomir R. Radev, Pradeep Muthukrishnan, Vahed Qazvinian, and Amjad Abu-Jbara. The ACL Anthology Network Corpus. *Language Resources and Evaluation*, 47(4):919–944, 2013. doi: 10.1007/s10579-012-9211-2. URL https://doi.org/10.1007/s10579-012-9211-2.

Dragomir R. Radev, Mark Thomas Joseph, Bryan R. Gibson, and Pradeep Muthukrishnan. A Bibliometric and Network Analysis of the Field of Computational Linguistics. *JASIST*, 67(3):683–706, 2016.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding with unsupervised learning. 2018.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv e-prints*, 2019.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1264. URL https://www.aclweb.org/anthology/D16-1264.

Prajit Ramachandran, Peter Liu, and Quoc Le. Unsupervised pretraining for sequence to sequence learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 383–391, Copenhagen, Denmark, September

2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1039. URL https://www.aclweb.org/anthology/D17-1039.

Peter A. Rankel, John M. Conroy, Hoa Trang Dang, and Ani Nenkova. A decade of automatic content evaluation of news summaries: Reassessing the state of the art. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 131–136, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P13-2024.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL http://arxiv.org/abs/1511.06732.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China, November 2019a. Association for Computational Linguistics. doi: 10.18653/v1/D19-1410. URL https://www.aclweb.org/anthology/D19-1410.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China, November 2019b. Association for Computational Linguistics. doi: 10.18653/v1/D19-1410. URL https://www.aclweb.org/anthology/D19-1410.

Steven J. Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. Self-

critical sequence training for image captioning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 1179–1195. IEEE Computer Society, 2017. doi: 10.1109/CVPR.2017.131. URL https://doi.org/10.1109/CVPR.2017.131.

Christina Sauper and Regina Barzilay. 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, August 2009. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P09-1024.

Thomas Scialom, Sylvain Lamprier, Benjamin Piwowarski, and Jacopo Staiano. Answers unite! unsupervised metrics for reinforced summarization models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3246–3256, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1320. URL https://www.aclweb.org/anthology/D19-1320.

Abigail See, Peter J. Liu, and Christopher D. Manning. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1099. URL https://www.aclweb.org/anthology/P17-1099.

Elaheh ShafieiBavani, Mohammad Ebrahimi, Raymond Wong, and Fang Chen. A graph-theoretic summary evaluation for ROUGE. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 762–767, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1085. URL https://www.aclweb.org/anthology/D18-1085.

Ori Shapira and Ran Levy. Massive multi-document summarization of product reviews with weak supervision, 2020.

Eva Sharma, Luyang Huang, Zhe Hu, and Lu Wang. An entity-driven framework for abstractive summarization. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3280–3291, Hong Kong, China, November 2019a. Association for Computational Linguistics. doi: 10.18653/v1/D19-1323. URL https://www.aclweb.org/anthology/D19-1323.

Eva Sharma, Chen Li, and Lu Wang. 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, July 2019b. Association for Computational Linguistics. doi: 10.18653/v1/P19-1212. URL https://www.aclweb.org/anthology/P19-1212.

Dou Shen, Jian-Tao Sun, Hua Li, Qiang Yang, and Zheng Chen. Document summarization using conditional random fields. In *IJCAI*, volume 7, pages 2862–2867, 2007.

Emily Sheng, Prem Natarajan, Jonathan Gordon, and Gully Burns. An investigation into the pedagogical features of documents. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 109–120, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-5012. URL https://www.aclweb.org/anthology/W17-5012.

Hongya Song, Zhaochun Ren, Shangsong Liang, Piji Li, Jun Ma, and Maarten de Rijke. Summarizing answers in non-factoid community question-answering. In Maarten de Rijke, Milad Shokouhi, Andrew Tomkins, and Min Zhang, editors, *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, WSDM 2017, Cambridge,*

United Kingdom, February 6-10, 2017, pages 405–414. ACM, 2017. doi: 10.1145/3018661.3018704. URL https://doi.org/10.1145/3018661.3018704.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. Learning to summarize from human feedback. *CoRR*, abs/2009.01325, 2020. URL https://arxiv.org/abs/2009.01325.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper/2014/file/a14ac55a4f27472c5d894ec1c3c743d2-Paper.pdf.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2818–2826. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.308. URL https://doi.org/10.1109/CVPR.2016.308.

Tiffany Ya Tang and Gordon I. McCalla. On the Pedagogically Guided Paper Recommendation for an Evolving Web-Based Learning System. In *FLAIRS Conference*, pages 86–92. AAAI Press, 2004.

Tiffany Ya Tang and Gordon I. McCalla. The Pedagogical Value of Papers: a Collaborative-Filtering based Paper recommender. *J. Digit. Inf.*, 10(2), 2009.

Mattia Tomasoni and Minlie Huang. Metadata-aware measures for answer summarization in community question answering. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 760–769, Uppsala, Sweden, July 2010. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P10-1078.

Lucy Vanderwende, Hisami Suzuki, Chris Brockett, and Ani Nenkova. Beyond sumbasic: Task-focused summarization with sentence simplification and lexical expansion. *Information Processing & Management*, 43(6):1606–1618, 2007.

Oleg Vasilyev, Vedant Dharnidharka, and John Bohannon. Fill in the BLANC: Human-free quality estimation of document summaries. In *Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems*, pages 11–20, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.eval4nlp-1.2. URL https://www.aclweb.org/anthology/2020.eval4nlp-1.2.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 4566–4575. IEEE Computer Society, 2015. doi: 10.1109/CVPR.2015.7299087. URL https://doi.org/10.1109/CVPR.2015.7299087.

Alex Wang, Kyunghyun Cho, and Mike Lewis. Asking and answering questions to evaluate the factual consistency of summaries. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5008–5020, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.450. URL https://www.aclweb.org/anthology/2020.acl-main.450.

Danqing Wang, Pengfei Liu, Ming Zhong, Jie Fu, Xipeng Qiu, and Xuanjing Huang. Exploring domain shift in extractive text summarization, 2019.

Liang Wang, Sujian Li, Wei Zhao, Kewei Shen, Meng Sun, Ruoyu Jia, and Jingming Liu. Multi-perspective context aggregation for semi-supervised cloze-style reading comprehension. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 857–867, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/C18-1073.

Lu Wang, Hema Raghavan, Claire Cardie, and Vittorio Castelli. Query-focused opinion summarization for user-generated content. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1660–1669, Dublin, Ireland, August 2014. Dublin City University and Association for Computational Linguistics. URL https://www.aclweb.org/anthology/C14-1157.

Mengqiu Wang, Noah A. Smith, and Teruko Mitamura. What is the Jeopardy model? a quasi-synchronous grammar for QA. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 22–32, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D07-1003.

Sarah Wiegreffe and Yuval Pinter. Attention is not not explanation. 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 11–20, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1002. URL https://www.aclweb.org/anthology/D19-1002.

Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Con-*

ference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pages 1112–1122, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1101. URL https://www.aclweb.org/anthology/N18-1101.

R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992a.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992b.

Ronald J Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280, 1989.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *arXiv preprint arXiv:1910.03771*, 2019.

Yuxiang Wu and Baotian Hu. Learning to extract coherent summary via deep reinforcement learning. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5602–5609. AAAI Press, 2018. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16838.

Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V. Le. Unsupervised data augmentation for consistency training. *arXiv preprint arXiv:1904.12848*, 2019.

Jiacheng Xu and Greg Durrett. Neural extractive text summarization with syntactic compression. In *Proceedings of the 2019 Conference on Empirical Methods in Nat-*

ural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 3292–3303, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1324. URL https://www.aclweb.org/anthology/D19-1324.

Jiaming Xu, Bo Xu, Peng Wang, Suncong Zheng, Guanhua Tian, and Jun Zhao. Self-taught convolutional neural networks for short text clustering. *Neural Networks*, 88:22–31, 2017.

Zhilin Yang, Junjie Hu, Ruslan Salakhutdinov, and William Cohen. Semi-supervised QA with generative domain-adaptive nets. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1040–1050, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1096. URL https://www.aclweb.org/anthology/P17-1096.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 5754–5764, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/dc6a7e655d7e5840e66733e9ee67cc69-Abstract.html.

Ziyi Yang, Chenguang Zhu, Robert Gmyr, Michael Zeng, Xuedong Huang, and Eric Darve. 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, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.168. URL https://www.aclweb.org/anthology/2020.findings-emnlp.168.

Michihiro Yasunaga, Rui Zhang, Kshitijh Meelu, Ayush Pareek, Krishnan Srinivasan, and Dragomir Radev. Graph-based neural multi-document summarization. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 452–462, Vancouver, Canada, August 2017. Association for Computational Linguistics. doi: 10.18653/v1/K17-1045. URL https://www.aclweb.org/anthology/K17-1045.

Dani Yogatama, Fei Liu, and Noah A. Smith. Extractive summarization by maximizing semantic volume. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1961–1966, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1228. URL https://www.aclweb.org/anthology/D15-1228.

Adams Wei Yu, David Dohan, Quoc Le, Thang Luong, Rui Zhao, and Kai Chen. Fast and accurate reading comprehension by combining self-attention and convolution. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=B14TlG-RW.

Fangfang Zhang, Jin-ge Yao, and Rui Yan. On the abstractiveness of neural document summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 785–790, Brussels, Belgium, October-November 2018a. Association for Computational Linguistics. doi: 10.18653/v1/D18-1089. URL https://www.aclweb.org/anthology/D18-1089.

Jianmin Zhang, Jiwei Tan, and Xiaojun Wan. Towards a Neural Network Approach to Abstractive Multi-Document Summarization. *CoRR*, abs/1804.09010, 2018b.

Jianmin Zhang, Jiwei Tan, and Xiaojun Wan. Adapting neural single-document summarization model for abstractive multi-document summarization: A pilot study. In *Proceedings of the 11th International Conference on Natural Language Generation*,

pages 381–390, Tilburg University, The Netherlands, November 2018c. Association for Computational Linguistics. doi: 10.18653/v1/W18-6545. URL https://www.aclweb.org/anthology/W18-6545.

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

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=SkeHuCVFDr.

Xingxing Zhang, Mirella Lapata, Furu Wei, and Ming Zhou. Neural latent extractive document summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 779–784, Brussels, Belgium, October-November 2018d. Association for Computational Linguistics. doi: 10.18653/v1/D18-1088. URL https://www.aclweb.org/anthology/D18-1088.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger. MoverScore: Text generation evaluating with contextualized embeddings and earth mover distance. 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 563–578, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1053. URL https://www.aclweb.org/anthology/D19-1053.

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

tional Linguistics. doi: 10.18653/v1/P19-1628. URL https://www.aclweb.org/anthology/P19-1628.

Jiawei Zhou and Alexander Rush. Simple unsupervised summarization by contextual matching. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5101–5106, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1503. URL https://www.aclweb.org/anthology/P19-1503.

Liang Zhou, Chin-Yew Lin, Dragos Stefan Munteanu, and Eduard Hovy. ParaEval: Using paraphrases to evaluate summaries automatically. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 447–454, New York City, USA, June 2006. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/N06-1057.

Qingyu Zhou, Nan Yang, Furu Wei, Shaohan Huang, Ming Zhou, and Tiejun Zhao. Neural document summarization by jointly learning to score and select sentences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 654–663, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1061. URL https://www.aclweb.org/anthology/P18-1061.

Haichao Zhu, Li Dong, Furu Wei, Bing Qin, and Ting Liu. Transforming Wikipedia into Augmented Data for Query-Focused Summarization. *arXiv preprint arXiv:1911.03324*, 2019.

Yukun Zhu, Ryan Kiros, Richard S. Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13,*

2015, pages 19–27. IEEE Computer Society, 2015. doi: 10.1109/ICCV.2015.11. URL https://doi.org/10.1109/ICCV.2015.11.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

Markus Zopf. Auto-hMDS: Automatic construction of a large heterogeneous multilingual multi-document summarization corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 2018. European Language Resources Association (ELRA). URL https://www.aclweb.org/anthology/L18-1510.

Markus Zopf, Eneldo Loza Mencía, and Johannes Fürnkranz. Which scores to predict in sentence regression for text summarization? In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1782–1791, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1161. URL https://www.aclweb.org/anthology/N18-1161.