Contrastive Text Generation

by

Darsh J. Shah

B.Tech, Indian Institute of Technology, Bombay (2016)
S.M., Massachusetts Institute of Technology (2019)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2021

Author . . . .
Department of Electrical Engineering and Computer Science
July 14, 2021

Certified by . . . .
Regina Barzilay
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by . . . .
Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

Contrastive Text Generation

by

Darsh J. Shah

Submitted to the Department of Electrical Engineering and Computer Science
on July 14, 2021, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract

This thesis focuses on developing summaries that present multiple view-points on issues of interest. Such capacity is important in many areas like medical studies, where articles may not agree with each other. While the automatic summarization methods developed in the recent decade excel in single document and multi-document scenarios with high content overlap amongst inputs, there is an increasing need to automate comparative summarization. This is evident by the number of services for such reviews in the domains of law and medicine. Building on a traditional generation pipeline of planning and realization, I propose models for three scenarios with contradictions where the planners identify pertinent pieces of information and consensus to adequately realize relations between them.

First, I tackle contradictions between an old piece of text and a claim for the task of factual updates. As there is no supervision available to solve this task, our planner utilizes a fact-checking dataset to identify disagreeing phrases in an old text with respect to the claim. Subsequently, we use agreeing pairs from the fact-checking dataset to learn a text fusion realizer. Our approach outperforms several baselines on automatically updating text and on a fact-checking augmentation task, demonstrating the importance of a planner-realizer pipeline which can deal with a pair of contrastive inputs.

Second, I describe an approach for multi-document summarization, where input articles have varying degrees of consensus. In a scenario with very few parallel data points, we utilize a planner to identify key content and consensus amongst inputs, and leverage large amounts of free data to train a fluent realizer. Compared to state-of-the-art baselines, our method produces more relevant and consensus cognisant summaries.

Third, I describe an approach for comparative summarization, where a new research idea is compared and contrasted against related past works. Our planner predicts citation reasons for each input article with current research to generate a tree of related papers. Utilizing an iterative realizer to produce citation reason aware text spans for every branch, our model outperforms several state-of-the-art summarization models in generating related work for scholarly papers.

Thesis Supervisor: Regina Barzilay
Title: Professor of Electrical Engineering and Computer Science

Acknowledgments

This thesis would not have been possible without my advisor, Regina Barzilay. I am forever grateful to Regina for the opportunity to work and study in the MIT NLP group. Her intellectual energy and vision make her a superstar researcher; her patience, care and guidance have also been instrumental in my progress. Regina's high standards have allowed me to work on novel and exciting problems. I have had a wonderful experience with her as my advisor.

My thesis committee members Tommi Jaakkola and Sebastian Riedel have been extremely supportive and patient. Tommi has always been available for research discussions, is very encouraging and inspiring. Sebastian was a great mentor from whom I learned a lot during my Facebook internship.

I am also grateful to my wonderful collaborators, Tal Schuster, Serene Yeo, Enrico Santus, Lili Yu, Tao Lei, Anh Tuan Luu, Jiang Guo, Raghav Gupta, Alessandro Moschitti, Salvatore Romeo and Preslav Nakov for the role they played in my learning. Tal and my shared interests in fact verification led to a very productive collaboration. Lili's meticulous feedback, support and resourcefulness led to an enjoyable research experience. These collaborations allowed me to think out-of-the-box, iron out cool ideas and be productive. I would also like to thank my labmates, Jiaming Guo, Yujia Bao, Tianxiao Shen, Wengong Jin, Benson Chen, Yujie Qian, Adam Yala, Adam Fisch, Victor Quach, Rachel Wu, Yuan Zhang, Karthik Narasimhan, Octavian Ganea, Bracha Laufer, Peter Mikhael and Jeremy Wohlwend for their help and interesting discussions during my PhD. A special thanks to our administrative assistant, Marcia Davidson for her help.

MIT is an extremely open, welcoming and progressive community. I had the fortune of developing cherished bonds with a lot of students from MIT during my PhD. I would like to thank my chess friends, Aviv Adler, Jerry Li, Luke Schaeffer, Jennifer Tang, Shalev Ben-David, Daniel Grier, Greg Bodwin, Vitaly Abdrashitov, Govind Ramnarayan and Siddhartha Jayanti for hours of *bughouse* on Fridays. The banter and carefree discussions during Theory Tea made me fall in love with the game again. My graduate housing friends Dheeraj Nagaraj, Prashanth Prakash, Prateesh Goyal, Malvika Verma, Srinivasan Raghuraman, Suhas Kowshik, Nithin

Reddy, Rohan Chitnis, Ranjan Anantharaman, Rohan Jaishankar, Nikhilesh Ghanta and Sharan Raja were a constant support. Movie nights, board games and weekly dinners at Honest Lowell made graduate school enjoyable, even during a pandemic. A special thanks to Janet Fischer from the EECS department for her constant help.

Lastly, I would like to thank my family, to whom I dedicate this thesis. My sister, Rushina Shah, who has always been a guardian and a companion, from Primary School to PhD. Getting a PhD from MIT CSAIL is the direct consequence of the values inculcated in me by my parents. My mother, Mamta Shah, whose constant support and ambition to not settle for the second best, led me to this stage, while my father, Jaidip Shah's emphasis on being creative and ignoring the noise inspired me to do research. I thank my sister, parents, grandmothers and other family members for their love, support and blessings.

Bibliographic Notes

The main ideas in this thesis have been published in four peer-reviewed conferences while one is under-review. The list of publications is as follows:

- **Chapter 2 - Automatic Fact-guided Sentence Modification** In the Proceedings of the AAAI Conference on Artificial Intelligence, 2020.

- **Chapter 3- Nutri-bullets Hybrid: Consensual Multi-document Summarization** In the Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2021.

- **Chapter 3 Dataset Creation- Nutri-bullets: Summarizing health studies by composing segments** In the Proceedings of the AAAI Conference on Artificial Intelligence, 2021.

- **Chapter 4- Generating Related Work for Scholarly Papers** submitted to the Conference of Empirical Methods in Natural Language Processing, 2021, currently under peer-review.

- **Appendix A- Towards debiasing fact verification models** In the Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, 2019.

The code for the work presented in this thesis is publicly available at `https://github.com/darsh10`

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In domains of medical recommendation and opinionated reporting, documents written on the same topic often disagree. In such scenarios, an ideal summary must consider all sources and highlight not only similarities in documents but also the differences. The demand for such capacity is underlined by the prevalence of services like Cochrane, where experts write summaries on various medical issues to present a collective view of the relevant scientific studies. Despite substantial manual efforts, such services are unable to keep up with the growing number of articles, consequently leading to summaries which are outdated and incomplete. In this thesis we develop novel models to automate the generation of comparative summaries.

Standard methods for automatic summarization are designed for one input document or multiple input documents with high content overlap. These methods typically require corpora with large number of parallel data points for training. For example, one of the benchmakr datasets, Multi-news, comprises of summaries from more than 250,000 articles. However, for several domains like medical studies that have widespread contradictions, only hundreds of parallel instances are available for training. Standard systems struggle to produce comparative summaries in such scenarios.

In the past, generation pipelines of content planning and surface realization models were proposed for comparative summaries [2]. These systems assume symbolic inputs from articles, acquired through straightforward extraction systems. The content planners apply simple rules on this symbolic data to identify consensus operators amongst inputs such as disagreement in facts, change of perspective about events, unavailable information or agreement amongst documents. Then, the operators are used to select hand-written templates, which are applied to the symbolic data to realize summaries. Such solutions can be effective in the narrow domains they are curated for (e.g., terrorism). However, as disparities in documents are prevalent across domains such as sports, politics, finance and scientific studies, the over-reliance on manual effort makes it difficult to scale such systems.

In this thesis, we take a hybrid approach. We aim to develop models based on a pipeline of **Content Planning** and **Surface Realization**, learned from the training data. Specifically, our goal is to have planners explicitly identify critical facts and relations in inputs such as disagreements in results, contrast in research methodology or polarizing spans. Utilizing this skeleton, we exceed the capabilities of modern systems when dealing with contradictions in inputs. Furthermore, we propose to generate summaries using learned realizers, in contrast to early solutions utilizing hand-written rules and templates [2]. This leads to robust solutions that would be applicable in a variety of settings.

In particular, we develop models for three real-world problems. In **Factual Updates**, we study a prompt driven text summarization task, when the modifications contradict old text. We study this task for Wikipedia articles, where outdated texts are updated using contradicting claims. In **Consensual Summarization**, we study a multi-document summarization task when inputs do not have full agreement. We aim to present summaries which are cognizant of consensus amongst inputs. In **Comparative Summarization**, we study a query driven summarization task, where outputs must emphasise similarities and differences with past work. We aim to generate related work sections for new research ideas by citing relevant past scholarly papers.

The settings above form a thorough test bed to study comparative text generation. Ubiquitous contradictions and limited parallel data underline the challenges to tackle these problems. In order to successfully address these constraints, we prioritize the following metrics.

1. **Content:** We aim to develop models that produce outputs which are relevant and closely mimic gold summaries for the tasks. In particular, models must be able to deal with contradictions amongst inputs and produce outputs which are

cognizant of consensus.

2. **Generalizability:** We aim to develop models which are applicable across a variety of settings. In particular, models must be able to utilize the limited parallel data available and vast amounts of unlabelled data to alleviate the need for manual efforts to generate novel text.

3. **Faithfulness:** Despite producing fluent text, modern neural network models can produce hallucinated text, especially in scenarios with limited training data. We aim to develop models which produce outputs that are faithful to their inputs. Models must explore the trade-off between extracting spans from inputs and generating novel text to avoid hallucinations.

I now provide a summary of our three applications and briefly describe our proposed models for each scenario.

1.1 Factual Updates

Task and Challenge: Our first task is a prompt driven update problem, where the goal is to train a generation system that can modify outdated text upon receiving contradictory claims. Online encyclopediae like Wikipedia contain large amounts of text that need frequent corrections and updates. Often, new information that drive updates may contradict existing content in encyclopediae. In this setting, we focus on automatically rewriting such dynamically changing articles.

In this constrained generation task we consider two contradicting inputs – old text and a new guiding claim. The output must be consistent with new information and fit into the rest of the document. This constrains solutions to modify only the necessary portions of old text, to maintain the rest for coherency. Unlike standard text rewriting problems, we do not have access to a parallel corpus that can be utilized to learn a model for our task.

Our Approach: In this work, we propose a two-step method to overcome challenges of a contrastive generation task with no direct supervision. The key idea is to have a hybrid solution where a planner incorporates contradictions between inputs to guide the generation by the realizer. Specifically, we incorporate a neutralizing planner to identify and remove contradicting components in the old text with respect to the claim. The planner is trained akin to a rationale-style extractor [3]. The planner's objective to mask minimal text spans in sentences such that polarizing pairs of a fact-checking dataset become neutral. The surface realizer fuses the amortized text with the claim to produce a sentence coherent with the rest of the document and consistent with the latest facts. The realizer is a novel two-encoder pointer generator network, trained using neutralized pairs constructed from agreeing pairs.

We apply our method to two tasks: fact-guided modifications and data augmentation for fact-checking. On the first task, our method is able to generate corrected Wikipedia sentences guided by unstructured textual claims. Evaluation on Wikipedia modifications demonstrates that our model's outputs were the most successful in making the requisite updates, compared to strong baselines. On the FEVER fact-checking dataset [4], our model is able to successfully generate new claim-evidence supporting pairs, starting with claim-evidence refuting pairs. This augmentation is intended to eliminate the bias due to giveaway phrases in refuted claims. Using these outputs to augment the dataset, we attain a 13% decrease in relative error on an unbiased evaluation set.

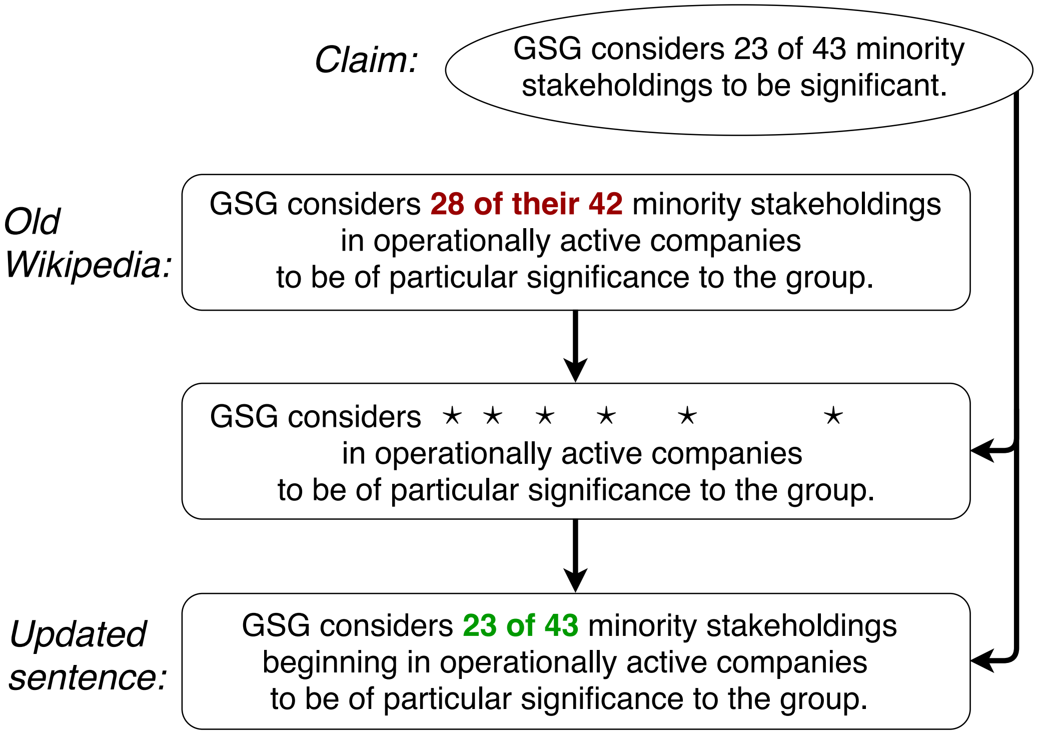

Claim:
GSG considers 23 of 43 minority stakeholdings to be significant.
Old Wikipedia:
GSG considers 28 of their 42 minority stakeholdings in operationally active companies to be of particular significance to the group.
GSG considers ⋆ ⋆ ⋆ ⋆ ⋆ ⋆ in operationally active companies to be of particular significance to the group.
Updated sentence:
GSG considers 23 of 43 minority stakeholdings beginning in operationally active companies to be of particular significance to the group.

Figure 1-1: Scenario demonstrating a factual update to an old sentence in a text compendium.

Figure 1-1 demonstrates a scenario where a model must perform the necessary update to an old Wikipedia sentence.

1.2 Consensual Summarization

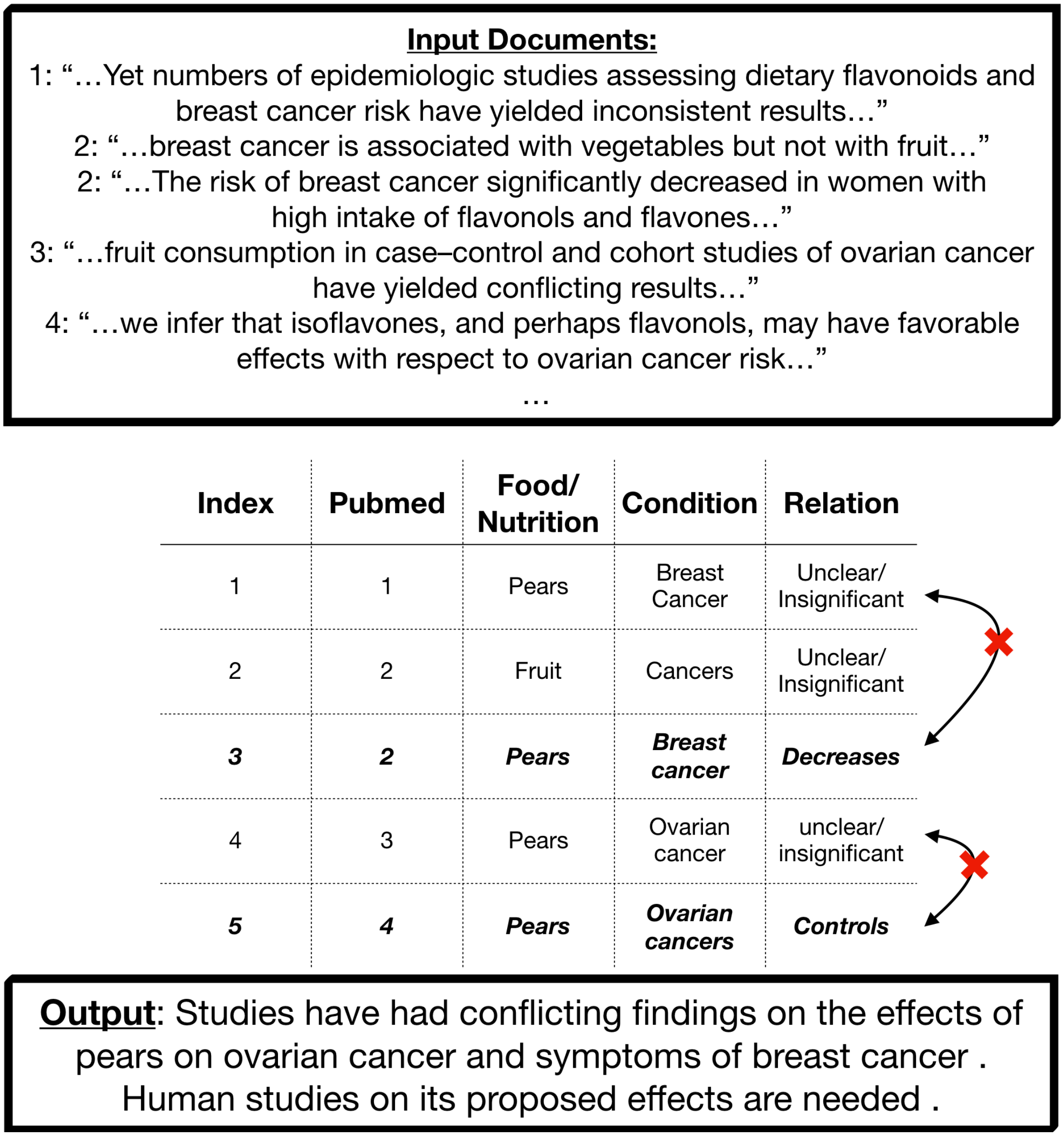
Task and Challenge: In our second task, we consider a multi-document summarization problem when inputs do not have full consensus. In domains like healthcare, studies often exhibit a wide divergence in findings. To present an unbiased overview of such material, human experts write comparative summaries which identify points of consensus and highlight contradictions. In this task, we aim to automate this capacity.

Standard summarization models are typically trained on hundreds of thousands of parallel instances. These solutions rely on similarities amongst input articles to produce summaries. However, in this task the amount of parallel data available is very limited and standard models struggle to present an unbiased view of all information.

Our Approach: In this task, we introduce a hybrid generation approach inspired by traditional concept-to-text systems. Utilizing the limited parallel data, the model first learns to extract pertinent relations from input documents. The content planning component uses deterministic operators to aggregate these relations after identifying a subset for inclusion. Having identified the content and aggregation amongst inputs, the surface realization component lexicalizes this information using a text-infilling language model. The surface realization model is trained using large amounts of unlabelled data. By separately modeling content selection and realization, we can effectively train them with limited annotations – maximizing the resources available.

Our model is able to generate robust summaries which are faithful to content and cognizant of the varying consensus in the input documents. Our approach is applicable to summarization and textual updates. Extensive experimentation, evaluated both automatically and subjectively underline its superiority over state-of-the-art baselines. For instance, on presenting the consensus amongst inputs and generating faithful outputs, our model outperforms the best baseline by 20% and 7% respectively.

Figure 1-2 demonstrates the kind of consensual summarization we want models to perform.



Input Documents:

1: “…Yet numbers of epidemiologic studies assessing dietary flavonoids and breast cancer risk have yielded inconsistent results…”

2: “…breast cancer is associated with vegetables but not with fruit…”

2: “…The risk of breast cancer significantly decreased in women with high intake of flavonols and flavones…”

3: “…fruit consumption in case–control and cohort studies of ovarian cancer have yielded conflicting results…”

4: “…we infer that isoflavones, and perhaps flavonols, may have favorable effects with respect to ovarian cancer risk…”

…

Index	Pubmed	Food/ Nutrition	Condition	Relation
1	1	Pears	Breast Cancer	Unclear/ Insignificant
2	2	Fruit	Cancers	Unclear/ Insignificant
3	***2***	***Pears***	***Breast cancer***	***Decreases***
4	3	Pears	Ovarian cancer	unclear/ insignificant
5	***4***	***Pears***	***Ovarian cancers***	***Controls***

Output: Studies have had conflicting findings on the effects of pears on ovarian cancer and symptoms of breast cancer . Human studies on its proposed effects are needed .

Figure 1-2: Generating a summary from multiple input documents that do not agree.

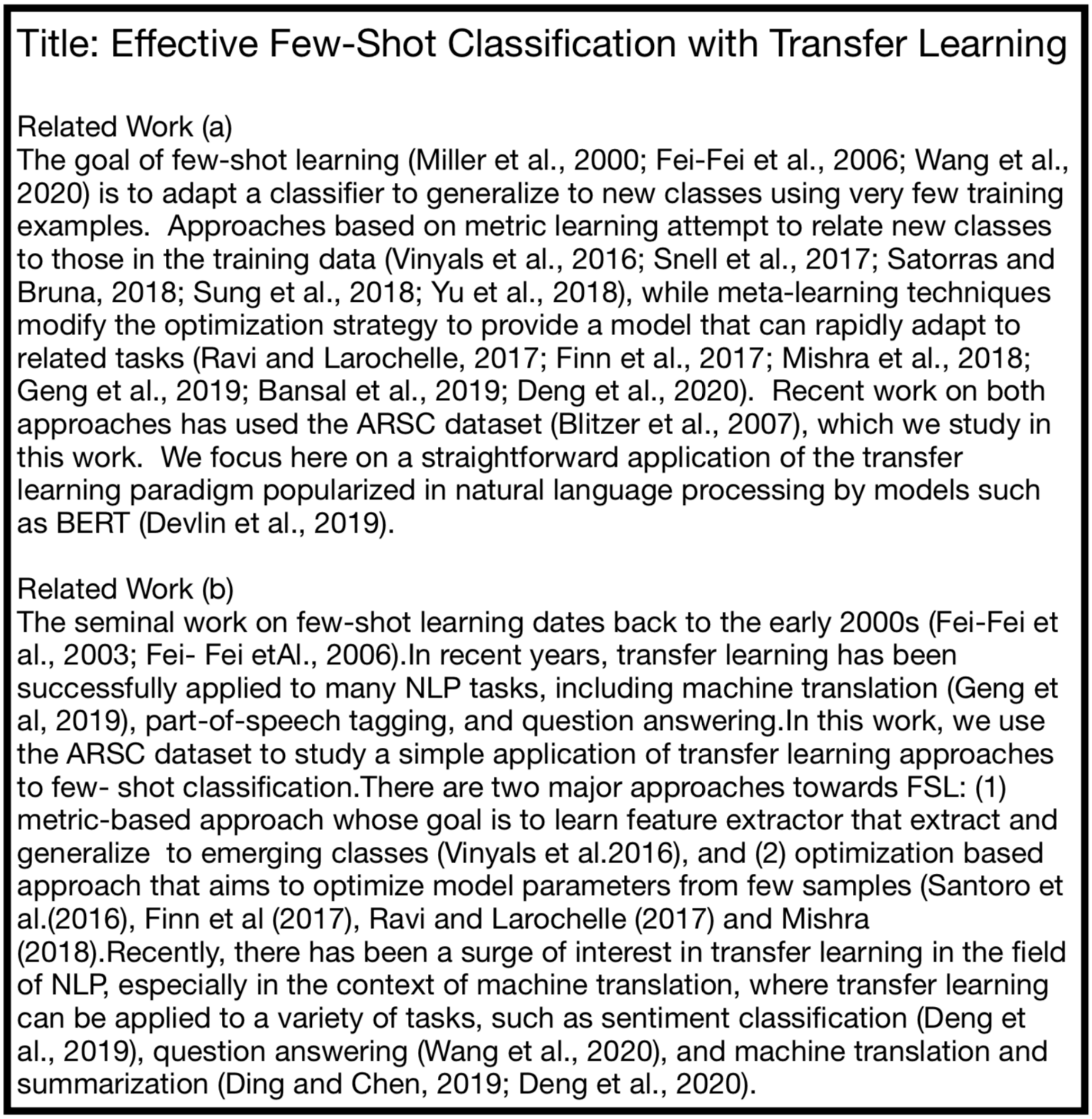
1.3 Comparative Summarization

Task and Challenge: In our final task, we study query driven comparative multi-document summarization. We address this problem in the domain of scholarly writing. An essential component of scientific writing is positioning new research in the landscape of existing work. Commonly presented in a related work section of scholarly papers, this comparison synthesizes information from multiple past papers related to the current research. While identification of such papers can be partially automated, the related work section itself is written by a human writer.

Generating a related work section is a multi-document summarization task. However, most existing summarization approaches operate over input documents with significant content overlap such as news. In a scenario with limited parallel data, standard systems are unable to identify and highlight specific relations of the current discovery to each article.

Our Approach: In this work, we model generating related work sections while predicting the reasons behind citing past papers. As in our previous solutions, we propose a two-step approach of content planning and surface realization. Our planner takes in all available past papers and the abstract of the new paper. It generates a (depth=2) tree by predicting the individual reasons for citing past papers, and subsequently sorting selected papers and grouping them into respective branches with a combined reason for citing each branch. Our surface realization model iteratively generates a text span for every branch by fluently lexicalizing the reason behind citing the set of papers. We apply our method for generating related work on an ACL Anthology dataset that we collect. We consider numerous experiments to compare our method against state-of-the-art multi-document and query driven summarization techniques. On the RougeL metric, our approach gets an absolute 5% improvement over the best baseline. The method strongly outperforms others in an update setting, further highlighting its real world applicability in a scenario where a human written section is missing a few papers.

Figure 1-3 demonstrates a machine driven generation of a related work section and compares it to a human written one.



Title: Effective Few-Shot Classification with Transfer Learning

Related Work (a)
The goal of few-shot learning (Miller et al., 2000; Fei-Fei et al., 2006; Wang et al., 2020) is to adapt a classifier to generalize to new classes using very few training examples. Approaches based on metric learning attempt to relate new classes to those in the training data (Vinyals et al., 2016; Snell et al., 2017; Satorras and Bruna, 2018; Sung et al., 2018; Yu et al., 2018), while meta-learning techniques modify the optimization strategy to provide a model that can rapidly adapt to related tasks (Ravi and Larochelle, 2017; Finn et al., 2017; Mishra et al., 2018; Geng et al., 2019; Bansal et al., 2019; Deng et al., 2020). Recent work on both approaches has used the ARSC dataset (Blitzer et al., 2007), which we study in this work. We focus here on a straightforward application of the transfer learning paradigm popularized in natural language processing by models such as BERT (Devlin et al., 2019).

Related Work (b)
The seminal work on few-shot learning dates back to the early 2000s (Fei-Fei et al., 2003; Fei- Fei etAl., 2006).In recent years, transfer learning has been successfully applied to many NLP tasks, including machine translation (Geng et al, 2019), part-of-speech tagging, and question answering.In this work, we use the ARSC dataset to study a simple application of transfer learning approaches to few- shot classification.There are two major approaches towards FSL: (1) metric-based approach whose goal is to learn feature extractor that extract and generalize to emerging classes (Vinyals et al.2016), and (2) optimization based approach that aims to optimize model parameters from few samples (Santoro et al.(2016), Finn et al (2017), Ravi and Larochelle (2017) and Mishra (2018).Recently, there has been a surge of interest in transfer learning in the field of NLP, especially in the context of machine translation, where transfer learning can be applied to a variety of tasks, such as sentiment classification (Deng et al., 2019), question answering (Wang et al., 2020), and machine translation and summarization (Ding and Chen, 2019; Deng et al., 2020).

Figure 1-3: Model generated related work section for a new paper.
Related Work (a) is human written and Related Work (b) is produced by our model.

1.4 Contributions

The primary contributions of this thesis are three-fold:

- **Incorporating contradictions amongst inputs to generate summaries.** We propose novel methods for three settings which deal with contradictory inputs. In Factual Updates, we propose a novel masking module to identify and delete phrases in an old text which contradict a new factual claim, before the ammortized text is fused into a factually updated sentence. In Consensual Summarization, our model identifies key content and pertinent relations amongst inputs, which are used to generate summaries that are cognizant of the degree of agreement or disagreement amongst input documents. In Comparative Summarization, our model describes the similarities and differences between an old text and a target text. Our related work generation model produces outputs which are cognizant of the reasons behind citing papers.

- **Generating factually consistent summaries in the presence of limited parallel data.** Our Factual Updates and Consensual Summarization tasks can leverage limited parallel data. We implement a novel split-encoder copy-generate realizer for the former, where the model decides whether to generate new words or copy from inputs to maintain faithfulness. In the latter case, our realizer is a text-infilling language model, with entities and relations from the inputs set as pivots to guide faithful generation.

- **Developing models for novel text generation applications.** We propose models for three novel real world settings. In Factual Updates, we automatically modify outdated Wikipedia sentences using new claims. In Consensual Summarization, we automatically generate summaries of medical studies, an extremely relevant domain. In Comparative Summarization, we automatically generate long and coherent related work sections for scholarly papers.

While our models are thoroughly examined in specific domains, they are applicable to a variety of settings – (1) Factual-updates are applicable to several text compendia and our model outputs can augment other understanding tasks such as natural language inference; (2) Symbolic hybrid solutions are applicable to a variety of settings, like Finance and opinionated reporting where inputs contradict as these models require minimal supervision for content planning and strongly leverage unlabelled data; (3) Related work generation can be extended to other domains of scholarly writing and law, where our pairwise reasoning between past articles and new ones would drive citation motivation cognizant generation.

1.5 Outline

The rest of this thesis is organised as follows:

- **Chapter 2** proposes a rationalization-inspired content planner which neutralizes contradictory phrases in old sentences. Following which a two-encoder surface realizer fuses new facts from the claim in an updated sentence.

- **Chapter 3** proposes a importance classification based content planner combined with a rule-based content aggregator which guide a surface realizer's generation of consensus cognizant summaries.

- **Chapter 4** proposes a content planner which generates a tree of cited papers along with motivations for citing each branch, before a surface realizer iteratively generates a coherent and motivation cognizant related work section.

- **Chapter 5** summarizes the thesis and proposes directions for future work.

Chapter 2

Factual Updates

In this chapter, we study a prompt driven fact update task. We explore the utilization of fact-checking data for textual updates. We demonstrate competent factual updates in scenarios where new information contradicts with existing text. Experimental results show our model outperforms strong baselines in producing accurate updates and that our model outputs are most suitable for augmenting a fact-checking dataset.

2.1 Introduction

Online text resources like Wikipedia contain millions of articles that must be continually updated. Some updates involve expansions of existing articles, while others modify the content. In this work, we are interested in the latter scenario where the modification contradicts the current articles. Such changes are common in online sources and often cover a broad spectrum of subjects ranging from the changing of dates for events to modifications of the relationship between entities. In these cases, simple solutions like negating the original text or concatenating it with the new information would not apply. In this work, our goal is to automate these updates. Specifically, given a claim and an outdated sentence from an article, we rewrite the sentence to be consistent with the given claim while preserving non-contradicting content.

Consider the Wikipedia update scenario depicted in Figure 2-1. The claim, informing that *23 of 43* minority stakeholdings are significant, contradicts the old infor-

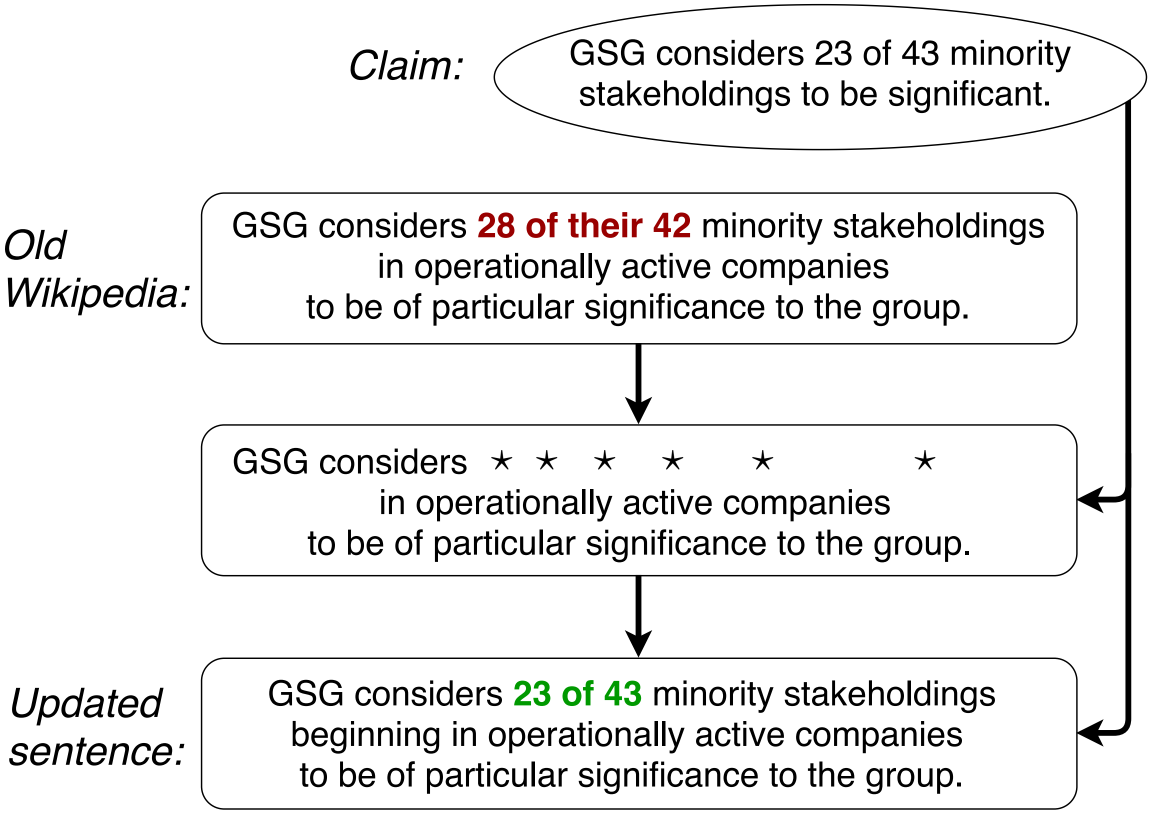

Claim:
GSG considers 23 of 43 minority stakeholdings to be significant.
Old Wikipedia:
GSG considers 28 of their 42 minority stakeholdings in operationally active companies to be of particular significance to the group.
GSG considers ⋆ ⋆ ⋆ ⋆ ⋆ ⋆ in operationally active companies to be of particular significance to the group.
Updated sentence:
GSG considers 23 of 43 minority stakeholdings beginning in operationally active companies to be of particular significance to the group.

Figure 2-1: Our fact-guided update pipeline.
Given a claim which refutes incorrect information, a planner is applied to remove the contradicting parts from the original text while preserving the rest of the context. Then, the residual neutral text and claim are fused to create an updated text that is consistent with the claim.

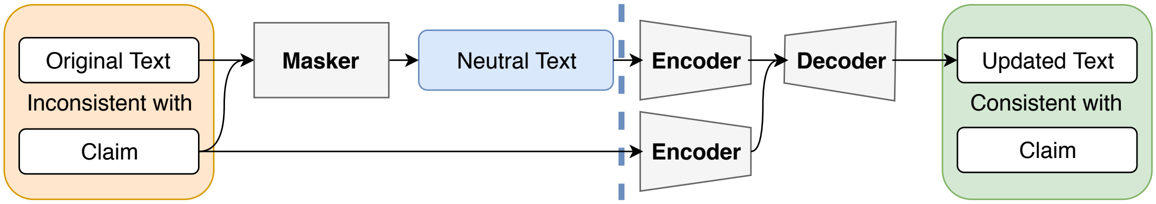


Figure 2-2: A summary of our pipeline.
Given a sentence that is inconsistent with a claim, a planner is applied to mask out the contradicting parts from the original text while preserving the rest of the content. Then, the residual neutral text and claim are fused to create an updated text that is consistent with the claim. The Content Planner and the Surface Realizer are trained separately.

mation in the Wikipedia sentence, requiring modification. Directly learning a model for this task would demand supervision, i.e. demonstrated updates with the corresponding claims. For Wikipedia, however, the underlying claims which drive the changes are not easily accessible. Therefore, we need to utilize other available sources of supervision.

In order to make the corresponding update, we develop a two step solution: (1) Content Planner to identify and remove the contradicting segments of the text (in this case, *28 of their 42 minority stakeholdings*); (2) Surface Realizer to rewrite the residual sentence to include the updated information (e.g. fraction of significant stakeholdings) while also preserving the rest of the content.

For the first step, we utilize a neutrality stance classifier as indirect supervision to identify the polarizing spans in the target sentence. We consider a sentence span as polarizing if its absence increases the neutrality of the claim-sentence pair. To identify and mask such sentence spans, we introduce an interpretability-inspired [5] neural architecture to effectively explore the space of possible spans. We formulate our objective in a way that the masking is minimal, thus preserving the context of the sentence.

For the second step, we introduce a novel, two-encoder decoder architecture, where two encoders fuse the claim and the residual sentence with a more refined control over their interaction.

We apply our method to two tasks: automatic fact-guided modifications and data augmentation for fact-checking. On the first task, our method is able to generate corrected Wikipedia sentences guided by unstructured textual claims. Evaluation on Wikipedia modifications demonstrates that our model's outputs were the most successful in making the requisite updates, compared to strong baselines. On the FEVER fact-checking dataset, our model is able to successfully generate new claim-evidence supporting pairs, starting with claim-evidence refuting pairs — intended to reduce the bias in the dataset. Using these outputs to augment the dataset, we attain a 13% decrease in relative error on an unbiased evaluation set.

2.2 Related Work

2.2.1 Text Rewriting

Recently, there have been several advancements in the field of text rewriting, such as style transfer [6, 7, 8] and sentence fusion [9, 10, 11]. In style transfer, models are constrained to maintain content of the original text while modifying its style (e.g. sentiment). While there is no parallel data for this task, the settings typically assume sufficient data for each predefined style. In our scenario, however, we must make modifications to text not along predefined style classes but along dimensions described by a guiding claim. In text fusion, models are encouraged to take two related spans of text and combine them into a coherent unit text. Unlike previous approaches, our sentence modification task addresses potential contradictions between two sources of information.

Our work is fairly related to the delete and generate approach of style transfer [12], which separates the task of sentiment transfer into deleting strong markers of sentiment in a sentence and retrieving markers of the target label to generate a sentence with the opposite sentiment. In contrast to such work, where the deletions are very task specific (e.g. sentiment), in our setting, an arbitrary input sentence (the claim) dictates the space of desired modifications. Therefore, in order to succeed at our task, a more general system is required which identifies the varying degree of polarization in the spans of the outdated sentence against the claim before modifying the sentence to be consistent with the claim.

2.2.2 Wikipedia Edits

Wikipedia edit history provides a realistic test bed for update tasks in Natural Language Processing. This setting was first studied for insights into the kinds of modifications made [13, 14, 15]. Recently, corpora based on edit history have been used for text generation tasks such as sentence compression and simplification [16], paraphrasing [17] and writing assistance [18]. However, in this work, we are interested in the novel task of automating the editing process with the guidance of a textual claim. Compared to previous generation tasks in this domain, our task incorporates information from two sources which could be contradictory in nature.

2.2.3 Fact Verification Datasets

Automatically detecting misinformation and fake news remains a challenge in modern Natural Language Processing. This has led to the development of several fact-checking datasets [19, 20, 21, 22] which tackle false information by verifying it against evidence. FEVER, the largest fact-checking dataset, contains 185K human written fake and real claims, generated by crowd-workers, verified against sentences from Wikipedia articles. We utilize this dataset as indirect supervision to identify polarizing spans in Wikipedia sentences with respect to guiding claims.

The FEVER dataset contains biases that allow a model to identify many of the false claims without any evidence [23]. This bias affects the generalization capabilities of models trained on such data. In this work, we show that our automatic modification method can also be used to augment a fact-checking dataset and to improve the inference of models trained on it.

2.2.4 Data Augmentation

Methods for data augmentation are commonly used in computer vision [24]. Synthetically augmenting datasets is an easy way to alleviate the need for additional crowd-sourcing to satisfy data hungry machine learning models. However, compared to images where rotation, translation and zooming are very effective augmentation tools, text requires careful modifications – an adversarial modification in a word or two can completely alter the semantics of text.

Recently, we have observed successes in Natural Language Processing where augmentation techniques such as paraphrasing and word replacement were applied to text classification [25, 26]. Adversarial examples in Natural Language Inference with syntactic modifications can also be considered as methods of data augmentation [27, 28]. In this work, we create constrained modifications, based on a reference claim, to augment data for our task at hand. Our additions are specifically aimed towards reducing the bias in the training data, by having a false claim appear in both "Agrees" and "Disagrees" classes.

2.3 Model

Problem Statement We assume access to a corpus $\mathcal{D}$ of claims and knowledge-book sentences. Specifically, $\mathcal{D} = \{\{C_1, ..., C_n\}, \{S_1, ..., S_m\}\}$, where C is a short factual sentence (claim), and S is a sentence from Wikipedia. Each pair of claim and Wikipedia sentence has a relation $rel(S, C)$, of either agree (A), disagree (D) or neutral (N). In this corpus, a Wikipedia sentence S is defined as outdated with respect to C if $rel(S, C) = D$ and updated if $rel(S, C) = A$. The neutral relation holds for pairs in which the sentence doesn't contain specific information about the claim.

Our goal is to automatically update a given sentence S, which is outdated with respect to a C. Specifically, given a claim and a pair for which $rel(S, C) = D$, our objective is to apply minimal modifications to S such that the relation of the modified sentence S^+ will be: $rel(S^+, C) = A$. In addition, S^+ should be structurally similar to S.

Framework Currently, to the best of our knowledge, there is no large dataset for fact-guided modifications. Instead, we utilize a large dataset with pairs of claims and sentences that are labeled to be consistent, inconsistent or neutral. In order to compensate the lack of direct supervision, we develop a two-step solution. First, using a pretrained fact-checking classifier for indirect supervision, we identify the polarizing spans of the outdated sentence and mask them to get a $S^{\emptyset}$ such that $rel(S^{\emptyset}, C) = N$. Then, we fuse this pair to generate the updated sentence which is consistent with the claim. This is done with a sequence-to-sequence model trained with consistent pairs through an auto-encoder style objective. The two steps are trained independently to simplify optimization (see 2-2).

2.3.1 Content Planner: Eliminate Polarizing Spans

In this section we describe the module to identify the polarizing spans within a Wikipedia sentence. Masking these spans ensures that the residual sentence-claim pairs attain a neutral relation. Here, neutrality is determined by a classifier trained on claim and Wikipedia sentence pairs as described below. Using this classifier, the

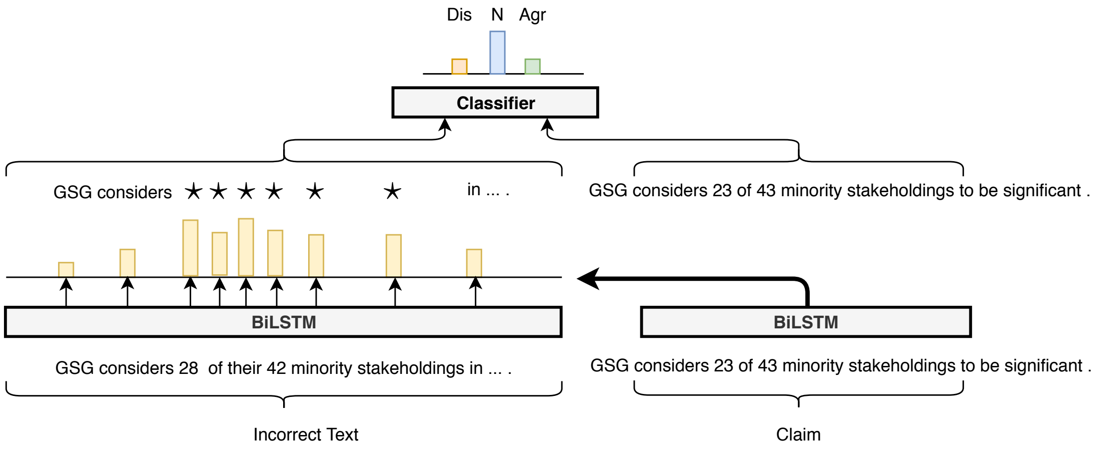

Dis
N
Agr
Classifier
GSG considers ★ ★ ★ ★ ★ ★ in ... .
GSG considers 23 of 43 minority stakeholdings to be significant .
BiLSTM
BiLSTM
GSG considers 28 of their 42 minority stakeholdings in ... .
GSG considers 23 of 43 minority stakeholdings to be significant .
Incorrect Text
Claim

Figure 2-3: Illustrating the flow of the Content Planner module.

masking module is trained to identify the polarizing spans by maximizing the neutrality of the residual-sentence and claim pairs. In order to preserve the context of the original sentence, we include optimization constraints to ensure minimal deletions. This approach is similar to neural rationale-based models [5], where a module tries to identify the spans of the input that justify the model's prediction.

Neutrality Content Planner Given a knowledge-book sentence (S) and a claim (C), the planner's goal is to create $S^{\emptyset}$ such that $rel(S^{\emptyset}, C) = N$. For the original sentence with l tokens, $S = \{x_i\}_{i=1}^{l}$, the output is a mask $m \in [0,1]^l$. The neutral sentence $S^{\emptyset}$ is constructed as:

$$S_i^{\emptyset} = \begin{cases} x_i, & \text{if } m_i = 0 \\ \star, & \text{otherwise} \end{cases} \tag{2.1}$$

where $\star$ is a special token.[1] The details of the content planner architecture are stated below and depicted in 2-3.

Encoding We encode S with a sequence encoder to get $e_i = f(x; \boldsymbol{w}_f)_i$. Since the neutrality of the sentence needs to be measured with respect to a claim, we also encode the claim and enhance S's representations with that of C using attention

[1] The special token is treated as an out-of-vocabulary token for the following models.

mechanism. Formally, we compute

$$z_i = e_i + \sum_{j=1}^{n} a_{i,j} \cdot c_j, \tag{2.2}$$

where c_j are the encoded representations of the claim and $a_{i,j}$ are the parameterized bilinear attention [29] weights computed by:

$$a_{i,j} = softmax_j(atten(e_i, c_j)), \tag{2.3}$$

$$atten(e_i, c_j) = e_i W c_j^T + b. \tag{2.4}$$

Finally, the aggregated representations are used as input to a sequence encoder $g(\cdot; \boldsymbol{w}_g)$.

Masking The encoded sentence is used to predict a per token masking probability:

$$p(m_i = 1) = \sigma(g(z; \boldsymbol{w}_g)_i). \tag{2.5}$$

Then, the mask is applied to achieve the residual sentence:

$$S^{\emptyset} = S \circ (1 - m), \tag{2.6}$$

where $\circ$ denotes element-wise multiplication. During training, we perform soft deletions over the token embeddings and add the out-of-vocabulary embedding in place. During inference, the values of m are rounded to create a discrete mask.

Training A pretrained fact-checking neutrality classifier's prediction $rel(S, C)$ is used to guide the training of the content planner. In order to encourage maximal retention of the context, we utilize a regularization term to minimize the fraction of the masked words. The joint objective is to minimize:

$$\mathcal{L}(S, C, m) = -\log\Big(p(rel(S^{\emptyset}, C) = N)\Big) + \frac{\lambda}{l}\sum_{i=1}^{l} m_i. \tag{2.7}$$

Fact-checking Neutrality Classifier Our fact-checking classifier is pretrained on agreeing and disagreeing (S, C) pairs from $\mathcal{D}$, in addition to neutral examples constructed through negative sampling. For each claim we construct a *neutral* pair by sampling a random sentence from the same paragraph of the polarizing sentence, making it contextually close to the claim, but unlikely to polarize it. We pretrain the classifier on these examples and fix its parameters during the training of the content planner.

Optional Syntactic Regularization Currently the model is trained with distant supervision, so, we pre-compute a valid neutrality mask as additional signal, when possible. To this end, we parse the original sentences using a constituency parser and iterate over continuous syntactic phrases by increasing length. For each sentence, the shortest successful neutrality mask (if any) is selected as a target mask.[2] In the event of successfully finding such a mask, the masking module is regularized to emulate the target mask by adding the following term to (2.7):

$$\frac{1}{l}||m - m'||^2, \tag{2.8}$$

where m' is the target mask.

Empirically, we find that the model can perform well even without this regularization, but it can help to stabilize the training. Additional details and analysis are available in the appendix.

2.3.2 Surface Realization: Constructing a Fact-updated Sentence

In this section we describe our method to generate an output which agrees with the claim. If the earlier masking step is done perfectly, the merging boils down to a simple fusion task. However, in certain cases, especially ones with a strong contradiction, our minimal deletion constraint might leave us with some residual contradictions in $S^{\emptyset}$. Thus, we develop a model which can control the amount of information to consider

[2] If there are several successful masks of the same length, we use the one with the highest neutrality score.

from either input.

We extend the pointer-generator model of [30] to enable multiple encoders. While sequence-to-sequence models support the encoding of multiple sentences by simply concatenating them, our use of a per input encoder allows the decoder to better control the use of each source. This is especially of interest to our task, where the context of the claim must be translated to the output while ignoring contradicting spans from the outdated Wikipedia sentence.

Next, we describe the details of our generator's architecture. Here, we use one encoder for the outdated sentence and one encoder for the claim. In order to reduce the size of the model, we share the parameters of the two encoders. The model can be similarly extended to any number of encoders.

Encoding At each time step t, the decoder output h^t, is a function of a weighted combination of the two encoders' context representations r^t, the decoder output in the previous step h^{t-1} and the representation of the word output at the end of the previous step $emb(y^{t-1})$:

$$h^t = RNN([r^t, emb(y^{t-1})], h^{t-1}). \tag{2.9}$$

As the decoder should decide at each time step which encoder to attend more, we introduce an encoder weight α. The shared encoder context representation r^t is based on their individual representations r_1^t and r_2^t:

$$\begin{gathered} \alpha = \sigma(u_{enc}^T [r_1^t, r_2^t]), \\ r^t = \alpha \cdot r_1^t + (1-\alpha) r_2^t. \end{gathered} \tag{10}$$

The context representation r_i^t ($i{\in}\{1,2\}$) is the attention score over the encoder representation r_i for a particular decoder state h^{t-1}:

$$z_j^t = u^T \tanh(r_{i,j} + h^{t-1}), \tag{2.10}$$

$$a_i^t = softmax(z^t), \tag{2.11}$$

$$r_i^t = \sum_j a_{i,j}^t r_{i,j}. \tag{11}$$

Decoding Following standard copy mechanism, predicting the next word y^t, involves deciding whether to *generate* (p_{gen}) or *copy*, based on the decoder input $x^t = [r^t, emb(y^{t-1})]$, the decoder state h^t and context vector r^t:

$$p_{gen} = \sigma(v_x^T x^t + v_h^T h^t + v_r^T r^t). \tag{12}$$

In case of copying, we need an additional gating mechanism to select between the two sources:

$$p_{enc1} = \sigma(u_x^T x^t + u_h^T h^t + u_r^T r^t). \tag{13}$$

When generating a new word, the probability over words from the vocabulary is computed by:

$$P_{vocab} = softmax(V^T[h^t, r^t]). \tag{14}$$

The final output of the decoder at each time step is then computed by:

$$\begin{aligned} P(w) = {} & p_{gen} P_{vocab}(w) + \\ & (1 - p_{gen})(p_{enc1}) \sum_{j:w_j=w} a_{1,j}^t + \\ & (1 - p_{gen})(1 - p_{enc1}) \sum_{j:w_j=w} a_{2,j}^t, \\ y^t = {} & \operatorname{argmax}_w P(w). \end{aligned} \tag{15}$$

where a^t are the input sequence attention scores from (11).

Training Since we have no training data for claim guided sentence updates, we train the generator module to reconstruct a sentence S to be consistent with an agreeing claim C. The training input is the residual up-to-date neutral sentence $S^{\emptyset}$ and the guiding claim C.

During inference, we utilize only guiding claims and residual outdated sentences $S^{\emptyset}$ to create S^{+}. While generating the updated sentences S^{+}, we would like to preserve as much context as possible from the contradicting sentence, while ensuring the correct relation with the claim. Therefore, for each case, if the later goal is not achieved, we gradually increase the focus on the claim by increasing α and p_{enc1} values until the output S^{+} satisfies $rel(S^{+}, C) = A$, or until a predefined maximum weight.

2.4 Experimental Setup

We evaluate our model on two tasks: (1) Automatic fact updates of Wikipedia sentences, where we update outdated wikipedia sentences using guiding fact claims; and (2) Generation of synthetic claim-evidence pairs to augment an existing biased fact-checking dataset in order to improve the performance of trained classifiers on an unbiased dataset.

2.4.1 Datasets

Training Data from FEVER We use FEVER [22], the largest available Wikipedia based fact-checking dataset to train our models for both of our tasks. This dataset contains claim-evidence pairs where the claim is a short factual sentence and the evidence is a relevant sentence retrieved from Wikipedia. We use these pairs as our claim-setnence samples and use the "refutes", "not enough information", "supports" labels of that dataset as our D, N, A relations, respectively.

Evaluation Data for Automatic Fact Updates We evaluate the automatic fact updates task on an evaluation set based on part of the symmetric dataset from [23] and the fact-based cases from a Wikipedia updates dataset [14]. For the symmetric dataset, we use the modified Wikipedia sentences with their guiding claims to generate the true Wikipedia sentence. For the cases from the updates dataset, we have human annotators write a guiding claim for each update and use it, together with the outdated sentence, to generate the updated Wikipedia sentence. Overall we have a total of 201 tuples of fact update claims, outdated sentences and updated sentences.

Evaluation Data for Augmentation To measure the proficiency of our generated outputs for data augmentation, we use the unbiased FEVER-based evaluation set of [23]. As shown by [23], the claims in the FEVER dataset contain give-away phrases that can make FEVER-trained models overly rely on them, resulting in decreased performance when evaluated on unbiased datasets.

The classifiers trained on our augmented dataset are evaluated on the unbiased symmetric dataset of [23]. This dataset (version 0.2) contains 531 claim-evidence pairs for validation and 534 claim-evidence pairs for testing.

In addition, we extend the symmetric test set by creating additional FEVER-based pairs. We hired crowd-workers on Amazon Mechanical Turk and asked them to simulate the process of generating synthetic training pairs. Specifically, for a "refutes" claim-evidence FEVER pair, the workers were asked to generate a modified supporting evidence while preserving as much information as possible from the original evidence. We collected responses of workers for 500 refuting pairs from the FEVER training set. This process extends the symmetric test set (+TURK) by 1000 cases — 500 "refutes" pairs, and corresponding 500 "supports" pairs generated by turkers.

2.4.2 Implementation Details

Content Planner We implemented the Content Planner using the AllenNLP framework [31]. For a neutrality classifier, we train an ESIM model [32] to classify a relation of A, D or N. To train this classifier, we use the A and D pairs from the FEVER dataset and for each claim we add a neutral sentence which is sampled from the sentences in the same document as the polarizing one. The classifier and planner are trained with GloVe [33] word embeddings. We use BiLSTM [34] encoders with hidden dimensions of 100 and share the parameters of the claim and original sentence encoders. The model is trained for up to 100 epochs with a patience value of 10, where the stopping condition is defined as the highest delta between accuracy and deletion size on the development set (Δ in 2.3).

	Automatic Evaluation				Human's Scores	
model	SARI	Keep	Add	Del	Grammar	Agreement
Fact updates:						
Split-no-Copy	15.1	36.9	1.9	49.5	-	-
Paraphrase	15.9	18.7	4.2	50.7	3.75	3.65
Claim Ext.	12.9	22.6	1.9	50.4	1.75	2.65
M. Concat	26.5	**61.7**	6.7	44.9	3.28	2.75
Ours	**31.5**	45.4	**13.2**	**52.1**	**3.85**	**4.00**
Human					4.80	4.70
Data augmentation:						
Paraphrase	18.2	12.5	10.6	45.7	4.12	3.92
Claim Ext.	12.2	9.8	4.0	46.4	1.58	2.84
M. Concat	22.1	**71.6**	6.8	22.3	**4.45**	2.05
Ours	**34.4**	33.0	**26.0**	**47.5**	4.14	**3.98**
Human					4.69	4.15

Table 2.1: Human evaluation results for our model's outputs for the fact update task (top) and for the data augmentation task (bottom).
The left part of the table shows the geometric SARI score with the three F1 scores that construct it. The right part shows the human's scores in a 1-5 Likert scale on grammatically of the output sentence and on agreement with the given claim.

For syntactic guidance, we use the constituency parser of [35] and consider continuous spans of length 2 to 10 as masking candidates (without combinations). By doing so, we obtain valid neutrality masks for 38% of the A and D pairs from the FEVER training dataset. These masks are used for 2.8.

Surface Realizer We implemented our proposed multi-sequence-to-sequence model, based on the pointer-generator framework.We use a one layer BiLSTM for encoding and decoding with a hidden dimension of 256. The parameters of the two encoders are shared. The model is trained with batches of size 64 for a total of 50K steps.

BERT Fact-Checking Classifier We use a BERT [36] classifier, which takes in as input a (claim-evidence) pair separated by a special token, to predict out of 3 labels (A, D or N). The model is fine-tuned for 3 epochs, which is sufficient to perform well on the task.

Evidence Regeneration Since we are interested in using the generated supporting pairs for data augmentation, we add machine generated cases to the A set of the dataset. Adding machine generated sentences to only one of the labels in the data can be ineffective. Therefore, we balance this by regenerating paraphrased refuting evidence for the false claims. This is then added along with all models' outputs for a balanced augmentation.

2.4.3 Baselines

We consider the following baselines for constructing a fact-guided updated sentence:

- **Copy Claim** The sentence of the claim is copied and used as the updated sentence for itself (used only for data augmentation).
- **Paraphrase** The claim is paraphrased using the back-translation method of [37][3], and the output is used as the updated sentence.
- **Claim Extension [Claim Ext.]** A pointer-generator network is trained to generate the updated sentence from an input claim alone. The model is trained on FEVER's agreeing pairs and applied on the to-be-updated claims during inference.
- **Masked Concatenation [M. Concat]** Instead of our Two-Encoder Generator, we use a pointer-generator network. The residual sentence (output from the content planner) and the claim are concatenated and used as input.
- **Split Encoder without Copy [Split-no-Copy]** Our Two-Encoder Generator, without the copy mechanism. The original text and contradicting claim are passed through each of the encoders.

2.5 Results

We report the performance of the model outputs for automatic fact-updates by comparing them to the corresponding correct wikipedia sentences. We also have crowd workers score the outputs on grammar and for agreeing with the claim. Additionally, we report the results on a fact-checking classifier using model outputs from the FEVER training set as data augmentation.

[3] `https://github.com/vsuthichai/paraphraser`

Model	Dev	Test	+Turk
No Augmentation	62.7	66.1	77.0
Paraphrase	60.8	64.6	77.4
Copy Claim	62.1	63.6	77.4
Claim Ext.	62.5	65.0	76.8
M. Concat	60.1	63.7	78.5
Ours	**63.8**	**67.8**	**80.0**

Table 2.2: Classifiers' accuracy on the symmetric Dev and Test splits. The right column (+Turk) shows the accuracy on the Test set extended to include the 500 responses of turkers for the simulated process and the refuted pairs that they originated from. The BERT classifiers were trained on the FEVER training dataset augmented by outputs of the different methods.

Fact Updates Following recent text simplification work, we use the SARI [38] method. The SARI method takes 3 inputs: (i) original sentence, (ii) human written updated sentence and (iii) model output. It measures the similarity of the machine generated and human reference sentences based on the deletions, additions and kept n-grams[4] with respect to the original sentence.[5] For human evaluation of the model's outputs, 20% of the evaluation dataset was used. Crowd-workers were provided with the model outputs and the corresponding supposably consistent claims. They were instructed to score the model outputs from 1 to 5 (1 being the poorest and 5 the highest), on grammaticality and agreement with the claim.

Table 2.1 reports the automatic and human evaluation results. Our model gets the highest SARI score, showing that it is the closest to humans in modifying the text for the corresponding tasks. Humans also score our outputs the highest for consistency with the claim, an essential criterion of our task. In addition, the outputs are more grammaticality sound compared to those from other methods.

Examining the gold answers, we notice that many of them include very minimal and local modifications, keeping much of the original sentence. The M. Concat model keeps most of the original sentence as is, even at the cost of being inconsistent with the claim. This corresponds to a high Keep score but a lower SARI score overall, and a low human score on supporting the claim. Claim Ext. and Paraphrase do

[4] We use the default up to 4-grams setting.

[5] Following [11] we use the F1 measure for all three sets, including deletions. The final SARI score is the geometric mean of the Add, Del and Keep score.

λ	Acc	size	Δ	Prec	Rec	F1
.5	5.1	0.0	5	0.0	0.0	0.0
.4	80.0	26.3	**54**	27.2	75.1	**39.9**
.3	77.0	27.5	50	25.9	71.6	38.0
.2	81.6	31.1	51	23.1	74.8	35.3

Table 2.3: Results of different values of λ for the planner with syntactic regularization. The left three columns describe the accuracy and average mask size (% of the sentence) over the FEVER development set with the masked evidence and a neutral target label. Δ is Acc – size. The right three columns contain the precision, recall and F1 of the masks that we have human annotations for. For results without syntactic regularization see the appendix.

not maintain the structure of the original sentence, and perform poorly on Keep, leading to a low SARI score. The Split-no-Copy model has the same low ADD score as Claim Ext. since instead of copying the accurate information from the claim, it generates other tokens.

Data Augmentation For 41850 D pairs in the FEVER training data, our method generates synthetic evidence sentences leading to 41850 A pairs. We train the BERT fact-checking classifier with this augmented data and report the performance on the symmetric dataset in Table 2.2. In addition, we repeat the human evaluation process on the generated augmentation pairs and report it in Table 2.1.

Our method's outputs are effective for augmentation, outperforming a classifier trained only on the original biased training data by an absolute 1.7% on the Test set and an absolute 3.0% on the +Turk set. The outputs of the Paraphrase and Copy Claim baselines are not Wikipedia-like, making them ineffective for augmentation. All the baseline approaches augment the false claims with a supported evidence. However, the success of our method in producing supporting evidence while trying to maintain a Wikipedia-like structure, leads to more effective augmentations.

Content Planner Analysis To evaluate the performance of the planner model, we test its capacity to modify A and D pairs from the FEVER development set to a neutral relation. We measure the accuracy of the pretrained classifier in predicting neutral versus the percentage of masked words from the sentence. For a finer evaluation, we manually annotated 75 A and 76 D pairs with the minimal required mask

Original Text	Born in Lawton , Oklahoma and raised in Anaheim , California , Hillenburg became fascinated with the sky as a child and also developed an interest in art .
Claim	Stephen Hillenburg was fascinated with the ocean as a child .
Claim Ext.	He in Huntington , Trinidad City Tommy in the , Hillenburg developed he became the of the stage , a senior . business in the adopted in 1847 .
Concat	Born in Lawton , Oklahoma and raised in Anaheim Anaheim , , Hillenburg became fascinated with the sky as a child and also developed an interest in art .
M. Concat	Born in Lawton , Oklahoma and raised in Anaheim , California , Hillenburg became the with the United as the condition and also developed an interest in art .
Ours	Born in Lawton , Oklahoma and raised in Anaheim , California , Hillenburg became fascinated with the ocean as a child and also developed an interest in art .
Original Text	Albert S. Ruddy -LRB- born March 28 , 1940 -RRB- is a Canadian - born film and television producer .
Claim	In 1930, Albert S. Ruddy is born.
Claim Ext.	Albert S. S. -LRB- -LSB- Hiram 23 , 1939 -RRB- is an former actor born theoretical marketer American . .
Concat	Albert S. Ruddy -LRB- born March March , , 1940 -RRB- is a Canadian - born film and television producer
M. Concat	Albert S. Ruddy -LRB- born Hiram 12 , 1930 -RRB- is a German - American film and television producer .
Ours	Albert S. Ruddy -LRB- born December 18 , 1930 -RRB- is a Chinese - born film and television producer .

Table 2.4: We compare our model outputs against different models.
Each example is showing the two input sentences following the output of each model. The Concat model setting is similar to the M. Concat one but the original text is left unmasked. For the Claim Ext. model, only the claim sentence is given as input.

for neutrality and compute the per token F1 score of the planner against them.

The results for different values of the regularization coefficient are reported in Table 2.3. Increasing the regularization coefficient helps to minimize the mask size and to improve the precision while maintaining the classifier accuracy and the mask recall. However, setting λ too large, can collapse the solution to no masking at all. The generation experiments use the outputs of the $\lambda = 0.4$ model.

Example Outputs Examples of outputs from different models are provided in 2.4. For the first example, our model produces a perfect update. In the last example, even though our model gets the year 1930 correct, it modifies the month and nationality to hallucinated, incorrect values. This is a result of a too aggressive deletion by the masker. The Claim Ext. model typically produces wrong and non-grammatical sentences. The Concat model doesn't capture the polarizing relation between the two inputs and mostly ignores the claim. The M. Concat model tends to overly generate made-up content instead of copying it from the claim.

Modeling Improvements While our model generates competent outputs which strongly outperform state-of-the-art methods, we see instances where improvements in modeling can benefit our performance.

- **Content Planner:** Our content planner is inspired by rationality span extractor models. For a text matching task, such a model identifies spans in one text (or both texts) which are responsible for their class label. A further refined solution can involve learning an alignment across spans between the two inputs. This would not only help eliminate erroneous insertions, but also ensure that factual updates fall into the appropriate positions in the output text. Optimal transport methods to identify such alignments should be explored [39].

- **Surface Realizer:** Our model is able to produce fluent text. Existing approaches, also rely on extensive pre-training on unlabelled data to further improve generation fluency. We consciously avoid such pre-training, so as to not bias our model with existing facts during generation. One solution to this problem could be to consider only those text spans in Wikipedia which have been updated, where more than one version of a text is available, and utilize such a corpora for pre-training.

2.6 Conclusion

In this chapter, we introduce the task of automatic fact-guided sentence modification. Given a claim and an old sentence, we learn to rewrite it to produce the updated sentence. Our method overcomes the challenges of this conditional generation task by breaking it into two steps. First, we implement a content planner to identify the polarizing components in the original sentence and mask them. Then, using the residual sentence and the claim, we generate a new sentence which is consistent with the claim, through our surface realizer. Applied to a Wikipedia fact update evaluation set, our method successfully generates correct Wikipedia sentences using the guiding claims. Our method can also be used for data augmentation, to alleviate the bias in fact verification datasets without any external data, reducing the relative error by 13%.

Chapter 3

Consensual Summarization

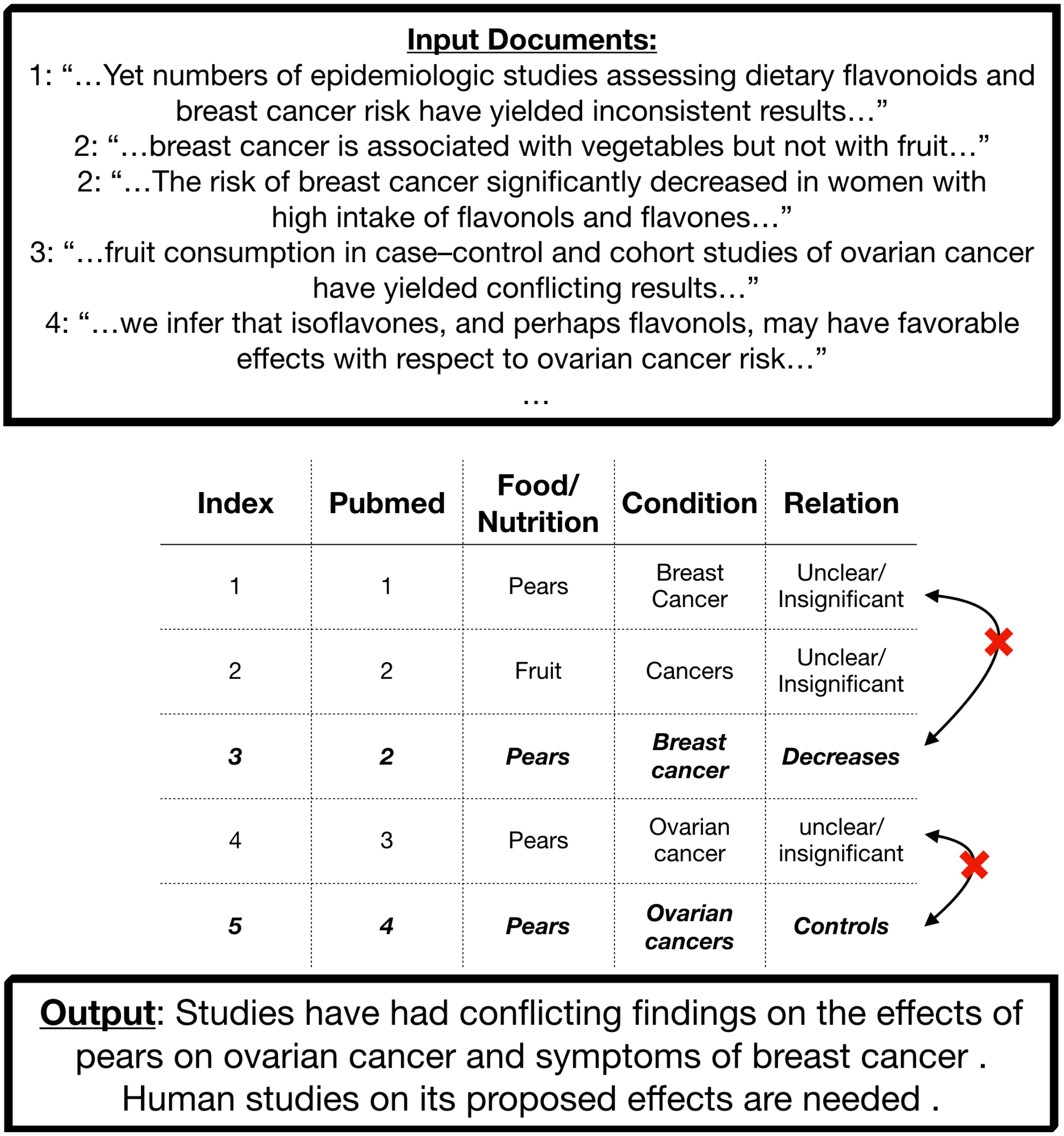
In this chapter, we study the generation of consensual summaries from inputs of multiple scientific documents which may not have complete agreement. We demonstrate the utiliziation of simple rules to identify aggregation and a data-driven generation system to produce robust outputs. Experimental results show our model outperforms state-of-the-art summarization methods in a domain with limited parallel data.

3.1 Introduction

Articles written about the same topic rarely exhibit full agreement. To present an unbiased overview of such material, a summary has to identify points of consensus and highlight contradictions. For instance, in the healthcare domain, where studies often exhibit wide divergence of findings, such comparative summaries are generated by human experts for the benefit of the general public.[1] Ideally, this capacity will be automated given a large number of relevant articles and continuous influx of new ones that require a summary update to keep it current. However, standard summarization architectures cannot be utilized for this task since the amount of comparative summaries is not sufficient for their training.

While modern language models [40] generate fluent text, they are not able to bring out the consensus amongst the input documents. In this paper, we propose a novel approach to multi-document summarization based on a neural interpretation

[1] Examples include `https://www.healthline.com` and `https://foodforbreastcancer.com`.



Input Documents:

1: "…Yet numbers of epidemiologic studies assessing dietary flavonoids and breast cancer risk have yielded inconsistent results…"

2: "…breast cancer is associated with vegetables but not with fruit…"

2: "…The risk of breast cancer significantly decreased in women with high intake of flavonols and flavones…"

3: "…fruit consumption in case–control and cohort studies of ovarian cancer have yielded conflicting results…"

4: "…we infer that isoflavones, and perhaps flavonols, may have favorable effects with respect to ovarian cancer risk…"

…

Index	Pubmed	Food/ Nutrition	Condition	Relation
1	1	Pears	Breast Cancer	Unclear/ Insignificant
2	2	Fruit	Cancers	Unclear/ Insignificant
3	***2***	***Pears***	***Breast cancer***	***Decreases***
4	3	Pears	Ovarian cancer	unclear/ insignificant
5	***4***	***Pears***	***Ovarian cancers***	***Controls***

Output: Studies have had conflicting findings on the effects of pears on ovarian cancer and symptoms of breast cancer . Human studies on its proposed effects are needed .

Figure 3-1: We consider the database extracted from four Pubmed studies on Pears and Cancer.
The key facts (*bold*) and consensus (*contradiction*) are realized in the text generated by our model.

of traditional concept-to-text generation systems. [41] study the aspect of consensus amongst input documents in a terror news domain and propose a symbolic system. While their system was based on human-crafted templates and thus limited to a narrow domain, we propose to learn different components of the generation pipeline from data. Hence, our symbolic solution is applicable to a variety of domains by leveraging limited parallel data for selection and unsupervised text for generation.

To fully control generated content, we frame the task of comparative summarization as concept-to-text generation. As a pre-processing step, we extract pertinent entity pairs and relations (see Figure 4-1) from input documents. The *Content Planning* component identifies the key tuples to be presented in the final output and establishes their comparative relations (e.g., consensus) via aggregation operators. Finally, the *surface realization* component utilizes a text-infilling language model to translate these relations into a summary. Figure 4-1 exemplifies this pipeline, showing selected key pairs (marked in bold), their comparative relation – *Contradiction* (rows 1 &3 and rows 4&5 conflict), and the final summary.[2]

This generation architecture supports refined control over the summary content, but at the same time does not require large amounts of parallel data for training. The latter is achieved by separately training Content Planning and content realization components. Since the Content Planning component operates over relational tuples, it can be robustly trained to identify salient relations utilizing limited parallel data. Aggregation operators are implemented using simple deterministic rules over the database where comparative relations between different rows are apparent. On the other hand, to achieve a fluent summary we have to train a language model on large amounts of data, but such data is readily available.

In addition to training benefits, this hybrid architecture enables human writers to explicitly guide Content Planning. This can be achieved by defining new aggregation operators and including new inference rules into the Content Planning component. Moreover, this architecture can flexibly support other summarization tasks, such as generation of updates when new information on the topic becomes available.

We apply our method for generating summaries of Pubmed publications on nutrition and health. Typically, a single topic in this domain is covered by multiple studies which often vary in their findings making it particularly appropriate for our model. We perform extensive automatic and human evaluation to compare our method against state-of-the-art summarization and text generation techniques. While seq2seq models receive competent fluency scores, our method performs stronger on

[2] We compare the selected content with other entries in the database, identifying two contradictions.

task-specific metrics including *relevance*, *content faithfulness* and *aggregation cognisance*. Our method is able to produce summaries that receive an absolute 20% more on aggregation cognisance, an absolute 7% more on content relevance and 7% on faithfulness to input documents than the next best baseline in traditional and update settings.

3.2 Related Work

3.2.1 Text-to-text Summarization

Over the past decade, neural sequence-to-sequence models [42, 43, 44] have become the standard for document summarizatio – tokens from input articles are fed to a neural encoder, whose representations are used by a similar neural decoder to produce a summary sequence of tokens. Trained on large amounts of data, such methods have shown promise and have been successfully adapted for multi-document summarization [45, 46, 47, 48, 49]. This is achieved through manipulations in the input space, such as concatenating articles using special tokens or modifications on the modeling side using hierarchical encoders [50]. Despite outperforming traditional statistical models in producing fluent text, these techniques may generate false information which is not faithful to the original inputs [51, 52]. Such phenomenon, is especially prevalent in low resource scenarios. In this work, we are interested in producing faithful and fluent text cognizant of aggregation amongst input documents, where few parallel examples are available.

Language modeling is the task of predicting the next word given context which is being utilized as a self-supervision task for pre-training Natural Language Processing Models [53, 36]. Instead of predicting the next word, modern language models also predict masked words from an otherwise complete context [36, 54, 55, 56]. Having been trained on large amounts of data for missing word prediction, such language models can also be extended for text completion. Our surface realizer is a text-infilling language model where we generate words in place of relation specific blanks to produce a faithful summary.

Aggregation Operator	Deterministic Rule
Under-Reported	$\lvert \text{Pubmed Studies} \rvert < \text{Threshold}$
Population Scoping	$\lvert \text{Specific Population} \rvert < \text{Threshold}$
Contradiction	$(e_1^m == e_1^n \&\& e_2^m == e_2^n \&\& r^m ! = r^n)$ for any two tuples m, n from different studies
Agreement	None of the Above

Table 3.1: Deterministic Rules to identify the Aggregation Operator.

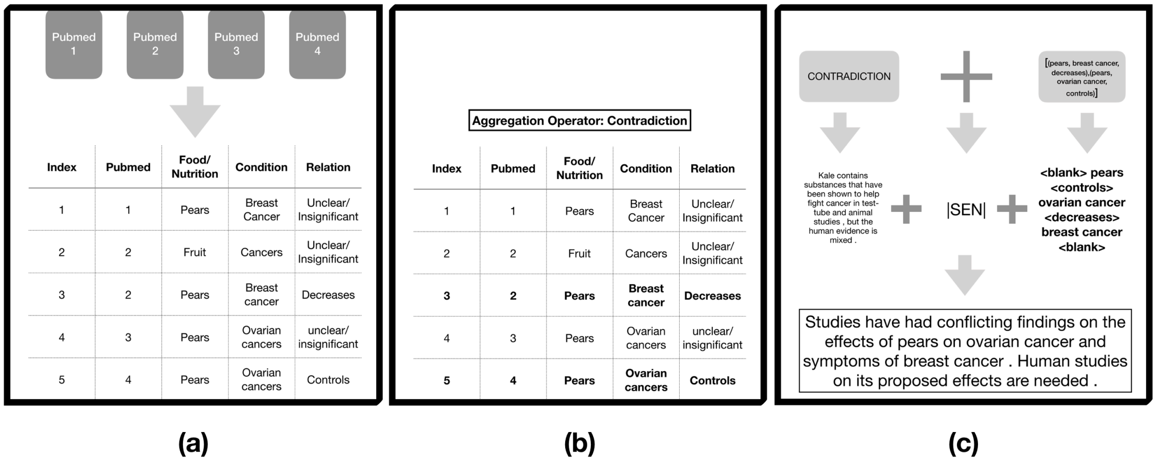
Prior work [57, 58, 59] on text generation also control aspects of the produced text, such as style and length. While these typically utilize tokens to control the modification, using prototypes to generate text is also very common [60, 61, 62]. In this work, we utilize aggregation specific prototypes to guide aggregation cognizant surface realization.

3.2.2 Data-to-text Summrization

While summarization is being studied in text-to-text settings, for several domains such as Sport games, Weather and Finance, generating a synopsis of a table or database is more appropriate. Traditional approaches for data-to-text generation have operated on symbolic data from databases. These works [63, 41, 64] introduce two components of content planning and surface realization. Content planning identifies and aggregates key symbolic data from the database which can then be realized into text using templates. Unlike modern systems [65, 51, 66, 67] these approaches capture document consensus and aggregation cognisance. While the neural approaches alleviate the need for human intervention, they do need an abundance of parallel data, which are typically from one source only. Hence, modern techniques do not deal with input documents' consensus in low resource settings. As a result, in this work, we are interested in modeling a fusion of the two approaches where simple rules can identify aggregation consensus and large unlabelled data can be utilized for neural generation.

3.3 Method

Our goal is to generate a text summary y for a food from a pool of multiple scientific abstracts X. In this section, we describe the framework of our *Nutribullets Hybrid*


Pubmed 1
Pubmed 2
Pubmed 3
Pubmed 4
(a)
Aggregation Operator: Contradiction
(b)
CONTRADICTION
[(pears, breast cancer, decreases),(pears, ovarian cancer, controls)]
Kale contains substances that have been shown to help fight cancer in test-tube and animal studies , but the human evidence is mixed .
|SEN|
<blank> pears <controls> ovarian cancer <decreases> breast cancer <blank>
(c)

Index	Pubmed	Food/ Nutrition	Condition	Relation
1	1	Pears	Breast Cancer	Unclear/ Insignificant
2	2	Fruit	Cancers	Unclear/ Insignificant
3	2	Pears	Breast cancer	Decreases
4	3	Pears	Ovarian cancers	unclear/ insignificant
5	4	Pears	Ovarian cancers	Controls

Index	Pubmed	Food/ Nutrition	Condition	Relation
1	1	Pears	Breast Cancer	Unclear/ Insignificant
2	2	Fruit	Cancers	Unclear/ Insignificant
3	**2**	**Pears**	**Breast cancer**	**Decreases**
4	3	Pears	Ovarian cancers	unclear/ insignificant
5	**4**	**Pears**	**Ovarian cancers**	**Controls**

Studies have had conflicting findings on the effects of pears on ovarian cancer and symptoms of breast cancer . Human studies on its proposed effects are needed .

Figure 3-2: Illustrating the flow of our *Nutribullets Hybrid* system.
In this example, our model takes in four Pubmed studies to produce a database (a). The *Content Planning* model selects two tuples (bold) and identifies the aggregation operator as Contradiction (b). Finally, the *Surface Realization* model takes in the tuples and aggregation operator to produces a summary which is faithful to input entities and aggregation cognizant (c).

system, illustrated in Figure 3-2.

3.3.1 Overview

We attain food health entity-entity relations, for both input documents X and the summary y, from entity extraction and relation classification modules trained on corresponding annotations (Table 3.4).

Notations: For N input documents, we collect $X_{\mathcal{G}} = \{\mathcal{G}_p^x\}_{p=1}^N$, a database of entity-entity relations $\mathcal{G}_p^x$. $\mathcal{G}_p = (e_1^k, e_2^k, r^k)_{k=1}^K$ is a set of K tuples of two entities e_1, e_2 and their relation r. r represents relations such as the effect of a nutrition entity e_1 on a condition e_2 (see Table 3.4).[3]We have raw text converted into symbolic data.

Similarly, we denote the corpus of summaries as $Y = \{(y_m, \mathcal{G}_m^y, O_m^y)_{m=1}^M\}$, where y_m is a concise summary, $\mathcal{G}_m^y$ is the set of entity-entity relation tuples and O_m^y is the realized aggregation, in M data points.

Modeling: Joint learning of Content Planning, information aggregation and text generation for multi-document summarization can be challenging. This is further

[3] We train an entity tagger and relation classifier to predict $\mathcal{G}$ and also for computing knowledge based evaluation scores. More details on models and results are shared later.

exacerbated in our technical domain with few parallel examples and varied consensus amongst input documents. To this end, we propose a solution using Content Planning and Aggregation and Surface Realization models.

Raw text from N input documents is converted into a mini-database $X_{\mathcal{G}}$ of relation tuples. The Content Planning and aggregation model operates on such symbolic data. We use $X_{\mathcal{G}}$ and Y to train the Content Planning model. During inference, we identify from $X_{\mathcal{G}}$ a subset C of content to present in the final output. In order to produce a summary cognizant of consensus amongst inputs, we identify the aggregation operator O based on C and other relevant tuples in $X_{\mathcal{G}}$.

The surface realization model produces a relevant, faithful and aggregation cognizant output. The model is trained only using Y. During inference, the model realizes text using the selected content C and the aggregation operator O.

3.3.2 Content Planning and Consensual Aggregation

Our Content Planning model takes a mini-database of entity-entity relation tuples $X_{\mathcal{G}}$ as input, and outputs the key tuples C and the aggregation operator O.

Content Planning and aggregation consists of two parts – (i) identifying key content $P(C|X_{\mathcal{G}})$ and (ii) subsequently identifying the aggregation operator O using $C, X_{\mathcal{G}}$.

Content Planning Identifying key content involves selecting important, diverse and representative tuples from a database. While clustering and selecting from the database tuples is a possible solution, we model our Content Planning as a finite Markov decision process (MDP). This allows for an exploration of different tuple combinations while incorporating delayed feedback from various critical sources of supervision (similarity with target tuples, diversity amongst selected tuples etc). We consider a multi-objective reinforcement learning algorithm [68] to train the model. Our rewards (Eq. 3.2) allow for the selection of informative and diverse relation tuples.

The MDP's state is represented as $s_t = (t, \{c_1, \ldots, c_t\}, \{z_1, z_2, ..., z_{m-t}\})$ where t is the current step, $\{c_1, \ldots, c_t\}$ is the content selected so far and $\{z_1, z_2, ..., z_{m-t}\}$ is the remaining entity-entity relation tuples in the m-sized database. The action space is all the remaining tuples plus one special token, $Z \cup \{STOP\}$.[4] The number of actions is equal to $|m-t|+1$. As the number of actions is variable yet finite, we parameterize the policy $\pi_\theta(a|s_t)$ with a model f which maps each action and state (a, s_t) to a score, in turn allowing a probability distribution over all possible actions using softmax. At each step, the probability that the policy selects z_i as a candidate is:

$$\pi_\theta(a{=}z_i|s_t) = \frac{\exp(f(t, \hat{z_i}, \hat{c_i}*))}{\sum_{j=1}^{m-t+1} \exp(f(t, \hat{z_j}, \hat{c_j}*))} \tag{3.1}$$

where $c_i* = argmax_{c_j}(cos(\hat{z_i}, \hat{c_j}))$ is the selected content closest to z_i, $\hat{z_i}$ and $\hat{c_i}*$ are the encoded dense vectors, $cos(u, v) = \frac{u \cdot v}{||u|| \cdot ||v||}$ is the cosine similarity of two vectors and f is a feed-forward neural network with non-linear activation functions that outputs a scalar score for each action a.

The selection process starts with Z. Our module iteratively samples actions from $\pi_\theta(a|s_t)$ until selecting $STOP$, ending with selected content C and a corresponding reward. We can even allow for the selection of partitioned tuple sets by adding an extra action of "NEW LIST", which allows the model to include subsequent tuples in a new group.

We consider the following individual rewards:

- $\mathcal{R}_e = \sum_{c \in C} cos(\hat{e_{1c}}, \hat{e_{1y}}) + cos(\hat{e_{2c}}, \hat{e_{2y}})$ is the cosine similarity of the structures of the selected content C with the structures present in the summary y (each summary structure accounted with only one c), encouraging the model to select relevant content.
- $\mathcal{R}_d = 1[max_{i,j}(cos(\hat{c_j}, \hat{c_i})) < \delta]$ computes the similarity between pairs within selected content C, encouraging the selection of diverse tuples.
- r_p is a small penalty for each action step to encourage concise selection.

The multi-objective reward is computed as

$$\mathcal{R} = w_e \mathcal{R}_e + w_d \mathcal{R}_d - |C| r_p, \tag{3.2}$$

where w_e, w_d and r_p are hyper-parameters.

[4] STOP and NEW LIST get special embeddings.

Relation Type	e_1	e_2	r	**Example**
Causing	Food, Nutrition	Condition	Increase, Decrease, Satisfy, Control, Unclear/Insignificant	(tart cherry juice, melatonin levels, increase), (water, daily fluid needs, satisfy)
Containing	Food, Nutrition	Nutrition	Contain	(blueberries, antioxidants, contain)

Table 3.2: Details of entity-entity relationships that we study and some examples of (e_1, e_2, r)

During training the model is updated based on the rewards. During inference the model selects an ordered set of key and diverse relation tuples corresponding to appropriate health conditions.

Consensus Aggregation Identifying the consensus amongst the input documents is critical in our multi-document summarization task. We model the aggregation operator of our *Content Planning* using simple one line deterministic rules as shown in Table 3.1. The rules are applied to the key C entity-entity relation pairs in context of $X_{\mathcal{G}}$. In our example in Figure 4-1, O is Contradiction because of rows 1&3 and rows 4&5 (rows 1&3 only would also make it Contradiction).

3.3.3 Surface Realization

The surface realization model $P(y|O, C)$, performs the critical task of generating a summary guided by both the entity-entity relation tuples C and the aggregation operator O. The model allows for robust, diverse and faithful summarization compared to traditional template and modern seq2seq approaches.

We propose to model this process as a prototype-driven text infilling task. The entities from C are used as fixed tokens with relations as special blanks in between these entities. This is prefixed by a prototype summary corresponding to O. For the example shown in Figure 3-2, we concatenate using $|SEN|$ a randomly sampled contradictory summary *"Kale contains substances ... help fight cancer ... but the human evidence is mixed ."* to C *"<blank> pears <controls> ovarian cancer <decreases> breast cancer <blank>"*. The infilling language model produces text corresponding to relations between entities while maintaining an overall structure which is cognizant of O. [5]

[5] Summaries in our training data are labelled with O_m^y as belonging to one of the four categories of *Under-reported, Population Scoping, Contradiction or Agreement* to accommodate such training.

Data	Task	# annotations	mean κ
Scientific	entity	83543	0.75
Abstracts	relation	28088	0.79, 0.81
HealthLine	entity	7860	0.86
	relation	5974	0.73, 0.90

Table 3.3: Entity and relation annotation statistics. Each annotation is from three annotators. Mean κ is the mean pairwise Cohen's κ score.

The model is trained on the few sample summaries from the training set using $\mathcal{G}_m^y$ and O_m^y to produce y_m. Providing aggregation and content guidance during generation alleviates the low-resource issue.

3.4 Data and Annotations

In this section, we describe the dataset collected for our *Nutri-bullet* system.

3.4.1 Corpus Collection

Our Healthline[6] dataset consists of scientific abstracts as inputs and human written summaries as outputs.

Scientific Abstracts We collect 6640 scientific abstracts from Pubmed, each averaging 327 words. The studies in these abstracts are cited by domain experts when writing summaries in the Healthline dataset. A particular food and its associated abstracts are fed as inputs to our *Nutri-bullet* systems. We exploit the large scientific abstract corpus when gathering entity and relation annotations (see Table 3.3) to overcome the challenge of limited parallel examples. Modules trained on these annotations can be applied to any food health scientific abstract.

Summaries Domain experts curate summaries for a general audience in the Healthline dataset. These summaries describe nutrition and health benefits of a specific food. In the HealthLine dataset, each food has multiple bullet summaries, where each bullet typically talks about a different health impact (hydration, anti-diabetic etc).

[6] https://www.healthline.com/nutrition

Relation Type	e_i	e_j	r	**Example**
Containing	Food, Nutrition	Nutrition	Contain	(apple, fiber, contain)
Causing	Food, Nutrition, Condition	Condition	Increase, Decrease, Satisfy, Control	(bananas, metabolism, increase), (orange juice, hydration, satisfy)

Table 3.4: Details of entity-entity relationships that we study and some tuple examples.

Parallel Instances The references in the human written summaries form natural pairings with the scientific abstracts. We harness this to collect 1894 parallel (abstracts, summary) instances in HealthLine. Summaries in HealthLine average 24.46 words, created using an average of 3 articles.

3.4.2 Entity and Relation Annotations

Despite having a small parallel data compared to [69, 70], we conduct large-scale crowd-sourcing tasks to collect entity and relation annotations on Amazon Mechanical Turk. The annotations (see Table 3.3) are designed to capture the rich technical information ingrained in such domains, alleviating the difficulty of multi-document summarization and are broadly applicable to different systems [71].

Entity and Relation Annotations Workers identify *food*, *nutrition*, *condition* and *population* entities by highlighting the corresponding text spans.

Given the annotated entities in text, workers are asked to enumerate all the valid relation tuples (e_i, e_j, r). Table 3.4 lists possible combinations of e_i, e_j and r for each relation type, along with some examples.

The technical information present in our domain can make annotating challenging. To collect reliable annotations, we set up several rounds of qualification tasks [7], offer direct communication channels to answer annotators' questions and take majority vote among 3 annotators for each data point. We collected 91K entities, 34K pairs of relations.

[7] To set up the qualification, the authors first annotate tens of examples which serve as gold answers. We leverage Mturk APIs to grade the annotation by comparing with the gold answers.

3.5 Experiments

	Automatic Evaluation				Human Scores	
Model	RougeL	KG(G)	KG(I)	Ag	Relevance	Fluency
Copy-gen	0.12	0.21	0.50	0.64	1.93	1.89
GraphWriter	0.14	0.03	0.69	0.64	1.86	2.76
Entity Data2text	0.16	0.13	0.57	0.67	2.03	3.43
Transformer	**0.20**	0.21	0.64	0.67	2.66	**3.76**
Ours	0.18	**0.30**	**0.76**	**0.89**	**3.03**	3.46

Table 3.5: Automatic evaluation – Rouge-L score (RougeL), KG in gold(G), KG in input(I) and Aggregation Cognisance (Ag) in our model and various baselines in the single issue setting, is reported.
Human evaluation on Relevance and Fluency, on 1-4 Likert scale from 3 annotators, is also reported. The best results are in **bold**.

Dataset We utilize a real world dataset for Food and Health summaries, crawled from `https://www.healthline.com/nutrition` [72]. The HealthLine dataset consists of scientific abstracts as inputs and human written summaries as outputs. The dataset consists of 6640 scientific abstracts from Pubmed, each averaging 327 words. The studies in these abstracts are cited by domain experts when writing summaries in the Healthline dataset, forming natural pairings of parallel data. Individual summaries average 24.5 words and are created using an average of 3 Pubmed abstracts. Each food has multiple bullet summaries, where each bullet typically talks about a different health impact (hydration, diabetes etc). We assign each food article randomly into one of the train, development or test splits. Entity tagging and relation classification annotations are provided for the Pubmed abstracts and the healthline summaries.

Settings: We consider three settings.

1. Single Issue: We use the individual food and health issue summaries as a unique instance of food and single issue setting. We split 1894 instances 80%,10%,10% to train, dev and test.

2. Multiple Issues: We group each food's article Pubmed abstract inputs and multiple summary outputs as a single parallel instance. 464 instances are split 80%,10%,10% to train, dev and test.

3. Summary Update: We consider two kinds of updates – new information is fused to an existing summary and new information contradicts an existing summary. For fusion we consider single issue summaries that have multiple conditions from different Pubmed studies (bananas + low blood pressure from one study and bananas + heart health from another study). We partition the Pubmed studies to stimulate an update. The contradictory update setting is where we artificially introduce conflicting results in the input document set so that the aggregation changes from Agreement to Contradictory. We have a total of 103 test instances. All models are trained atop of Single issue data.

Evaluation We evaluate our systems using the following automatic metrics. *Rouge* is an automatic metric used to compare the model output with the gold reference [73]. *KG(G)* computes the number of entity-entity pairs with a relation in the gold reference, that are generated in the output.[8] This captures relevance in context of the reference. *KG(I)*, similarly, computes the number of entity-entity pairs in the output that are present in the input scientific abstracts. This measures faithfulness with respect to the input documents. *Aggregation Cognisance (Ag)* measures the accuracy of the model in producing outputs which are cognizant of the right aggregation from the input, (Under-reported, Contradiction or Agreement). We use a rule-based classifier to identify the aggregation implied by the model output and compare it to the actual aggregation operator based on the input Pubmed studies.

In addition to automatic evaluation, we have human annotators score our models on relevance and fluency. Given a reference summary, *relevance* indicates if the generated text shares similar information. *Fluency* represents if the generated text is grammatically correct and written in well-formed English. Annotators rate relevance and fluency on a 1-4 likert scale [74]. We have 3 annotators score every data point and report the average across the scores.

Baselines In order to demonstrate the effectiveness of our method, we compare it against text2text and data2text state-of-the-art (*sota*) methods.

Copy-gen (Text2text): [44] is a *sota* technique for summarization, which can copy from the input or generate words.

[8] We run entity tagging plus relation classification on top of the model output and gold summaries. We match the gold (e_i^g, e_j^g, r^g) tuples using word embedding based cosine similarity with the corresponding entities in the output structures (e_i^o, e_j^o, r^o). A cosine score exceeds a threshold of 0.7 is set (minimize false positives) to identify a match.

Aggregation Operator	Deterministic Rule
Population Scoping	$\lvert$Specific Population$\rvert$ < Threshold
Contradiction	”evidence is mixed”, ”conflicting” or ”contradiction” in y_m
Under-Reported	”more research is needed”, ”more studies are needed” or ”more human studies” in y_m
Agreement	None of the Above

Table 3.6: Deterministic Rules to identify the Aggregation Operator on outputs.

Transformer (Text2text): [75] is a summarization system using a pretrained Transformer.
GraphWriter (Data2text): [71] is a graph transformer based model, which generates text using a seed title and a knowledge graph. Takes the database $X_{\mathcal{G}}$ as input.
Entity (Data2text): [51] is an entity based data2text model, takes $X_{\mathcal{G}}$ as input.

Implementation Details Our policy network is a three layer feedforward neural network. We use a Transformer [76] implementation for Surface Realization. We train an off-the-shelf Neural CRF tagger [77] for entity extraction. We use BERT [36] based classifiers to predict the relation between two entities in a text trained using crowdsourced annotations from [72].

The hyper-parameters for the content selection model are shared along with the code. The hyper-parameters for surface realization [78] as the default values present in `https://github.com/pytorch/fairseq/blob/master/examples/bart/README.md`.

Baselines We use publicly available implementations for all our baselines. Copygen is from `https://github.com/atulkum/pointer_summarizer`. GraphWriter is from `https://github.com/rikdz/GraphWriter`. The Transformer for abstractive summarization, pretrained implementation is from `https://github.com/Andrew03/transformer-abstractive-summarization`. Entity Data2text implementation is the closest working and usable implementation of the model in Opennmt-py `https://github.com/OpenNMT/OpenNMT-py/blob/master/config/config-rnn-summarization.yml`. We provide the full database $X_{\mathcal{G}}$ as input along with the food name.

Aggregation Operator for Outputs We specify our rule based classifier for summary y_m's aggregation operator O^y_m identification. The following patterns are checked.

Transformer (baseline)
* Whole - grain cereals may protect against obesity , diabetes and certain cancers. However , more research is needed .
* Whole grains , such as mozambican grass , are safe to eat with no serious side effects .
* Whole - grain cereals may protect against obesity , diabetes and certain cancers. However , more research is needed .
* Whole grains , such as blueberries , are likely safe to eat with no serious side effects .
* Whole grains are safe to eat. However , people with type 2 diabetes should avoid whole grains .
* Whole grains are lower in carbs than whole grains , making them a good choice for people with type 2 diabetes.
Our Method
* Whole grains has been shown to lower weight gain and improve various type 2 diabetes risk factors .
* Whole grains has been shown to lower insulin resistance and improve various cancer risk factors .
* Whole grains has been linked to several other potential health benefits , such as improved CVD risk , eyesight , and memory. However, more studies are needed to draw stronger conclusions.
* There is some evidence , in both animals and humans , that whole grains can reduce mortality by regulating the hormone ghrelin.

Table 3.7: Example outputs of our model and the Transformer baseline for a multi-issues summary.

Trained on limited parallel data, the Transformer baseline produces repetitive text with factual inaccuracies, while our method is able to provide more accurate and diverse summarization.

3.6 Results

In this section, we describe the performance of our *Nutribullet Hybrid* system and baselines on summarization and summary updates. We report empirical results , human evaluation and present sample outputs, highlighting the benefits of our method.

Model	KG(G)	KG(I)
Copy-gen	0.43	0.69
Transformer	0.33	0.73
Ours	**0.5**	**0.90**

Table 3.8: KG in gold(G) and KG in input(I) in our model and baselines in the food and multi-issues setting . The best results are in **bold**.

Single and Multi-issues Summarization: We describe the results on the task of generating summaries. Table 3.5 presents the automatic evaluation results for the food and single issue summarization task. High KG(I) and KG(G) scores for our method indicate that the generated text is faithful to input entities and relevant.

In particular, a high Aggregation Cognisance (Ag) score indicates that our model generates summaries which are cognizant of the varying degrees of consensus in the input Pubmed documents. Compared to other baselines we also receive a competitive score on the automatic Rouge metric, beating Copy-gen, Entity Data2text and GraphWriter baselines while falling short (by 1.7%) of the Transformer baseline. The baselines, especially Transformer, tend to produce similar outputs for different inputs (see Table 3.7). Since a lot of these patterns are learned from the human summaries, Transformer receives a high Rouge score. However, as in the low resource regime, the baseline does not completely capture the content and aggregation, it fails to get a very high KG(G) or Ag score. A similar trend is observed for the other baselines too, which in this low resource regime produce a lot of false information, reflected in their low KG(I) scores.

Human evaluation, conducted by considering scores,on a 1-4 Likert scale, from three annotators for each instance, shows the same pattern. Our model is able to capture the most relevant information, when compared against the gold summaries while producing fluent summaries. The Transformer baseline produces fluent summaries, which are not as relevant. The performance is poorer for the Copy-gen, Entity Data2text and GraphWriter models.

In the multi-issues setting, the baselines access the gold annotations with respect to the input documents' clustering. Our model conducts the extra task of grouping the selected tuples, using the "New List" action. Our model performs better than the baselines on both the KG(I) and KG(G) metrics as seen in Table 3.8. Again, the pattern of producing very similar and repetitive sentences hurts the baselines. They fail to cover different issues and tend to produce false information, in this low resource setting. Our model scores an 7% higher on KG(G) and 17% higher on KG(I) compared to the next best performance, in absolute terms. Table 3.7 shows the comparison between the outputs produced by our method and the Transformer baseline on the benefits of whole-grains. Our method conveys more relevant, factual and organized information in a concise manner.

Summary Update: We study the efficacy of our model to fuse information in existing summaries on receiving new Pubmed studies. As the KG(G) metric in 3.9 shows, our model is able to select and fuse more relevant information. Table 3.10

Model	Fusion Update KG(G)	Contradictory Update Ag
Copy-gen	0.16	0.50
GraphWriter	0.0	0.50
Entity Data2text	0.16	0.50
Transformer	0.16	0.46
Ours	**0.33**	**0.76**

Table 3.9: The middle column shows KG in gold(G) in our model and baselines for fusion updates .

The last column shows Aggregation Cognisance (Ag) in our model and baselines in the contradictory update setting. The best results are in **bold**.

Old Summary	Flax seeds contain a group of nutrients called lignans , which have powerful antioxidant and estrogen properties .
New Inputs	(i):"...current overall evidence indicates that FS and its components are effective in the risk reduction and treatment of breast cancer and safe for consumption by breast cancer patients..." (ii): "...Consumption of flaxseed was associated with a significant reduction in breast cancer risk as was consumption of flax bread ..." (iii): "...a flaxseed-supplemented, fat-restricted diet may affect the biology of the prostate and associated biomarkers..."
Copy-gen	Avocados may help fight cancer risk, boost inflammation. In a pasteurized called polyphenols, which may aid weight loss.
Transformer	Flaxseed oil is high in antioxidants that may help reduce the risk of several chronic diseases .
Ours	Flax seeds are rich in antioxidant , especially through lignans. They contain beneficial nutrients which can help protect your body against certain types of breast cancer .
Old Summary	Flax seeds, high in fiber, can be a beneficial addition to the diet of people with diabetes .
New Input	"...showed fasting blood sugar in the experimental group decreased...the total cholesterol reduced...Results showed a decrease in low-density lipoprotein cholesterol...The study demonstrated the efficacy of flax gum in the blood biochemistry profiles of type 2 diabetes."
Copy-gen	Eating apart has been linked to increased growth cholesterol, and cholesterol levels. However, more studies are needed to confirm possible effect.
Transformer	Flaxseed extract may help lower blood sugar levels .
Ours	Flax seeds are high in fiber , which is beneficial for people with diabetes and associated with a reduced low-density lipoprotein cholesterol .

Table 3.10: Example outputs of our model and baselines for a summary update upon receiving new information about flaxseeds + cancer and flaxseeds + cholesterol, respectively.

Our model maintains old information and updates accurately. In the cholesterol case, Transformer adds new information but misses the old information.

shows two examples of summaries on flaxseeds where our model successfully fuses new information.

Table 3.9's last column presents the automatic evaluation results to demonstrate the efficacy of maintaining Aggregation Cognisance (Ag), which is critical when updating summaries on receiving contradictory results. The high performance in this update setting demonstrates the *Surface Realization* model's ability to produce aggregation cognizant outputs, in contrast to the baselines that do not learn this reasoning in a low resource regime.

Analysis: Information Extraction and Content Aggregation Information extraction is the critical first step performed for the input documents in order to get symbolic data for Content Planning and aggregation. To this end, we report the performance of the information extraction system, which is composed of two models – entity extraction and relation classification. As reported in Table 3.11, the entity extraction model, a crf-based sequence tagging model, receives a token-level F1 score of 79%. The relation classification model, a BERT based text classifier, receives an accuracy of 69%.

The performance of the information extraction models is particularly important for the content aggregation sub-task. In order to analyse this quantitatively, we perform manual analysis of the 179 instances in the dev set and compare them to the system identified aggregation – information extraction followed by the deterministic rules in Table 3.1. Given the simplicity of our rules, system's 78% accuracy in Table 3.11 is acceptable. Deeper analysis shows that the performance is lowest for Population Scoping and Contradiction with an accuracy of 52% and 56% respectively. The performance of Population Scoping being low is down predominantly to the simplicity of the rules. Most mistakes occur when the input studies are review studies that don't mention any population but analyze results from several past work. Contradiction suffers because of the information extraction system and stronger models for the same should be able to alleviate the errors.

Modeling Improvements Our model produces competent consensus cognisant outputs. In certain cases, it does generate false information, albeit at a much lower rate than state-of-the-art methods. Instead of using the one-shot generation paradigm

Task	Performance
Entity Extraction	0.79
Relation Classification	0.69
Aggregation Operator Identification	0.78

Table 3.11: Performance of our information extraction system and its impact on content aggregation.

currently employed, an iterative re-writing mechanism where false information is eliminated can be employed.

3.7 Conclusion

While modern models produce fluent text in multi-document summarization, they struggle to capture the consensus amongst the input documents. This inadequacy – magnified in low resource domains, is addressed by our model. Our model is able to generate robust summaries which are faithful to content and cognizant of the varying consensus in the input documents. Our approach is applicable in summarization and textual updates. Extensive experiments, automatic and human evaluation underline its impact over state-of-the-art baselines.

Chapter 4

Comparative Summarization

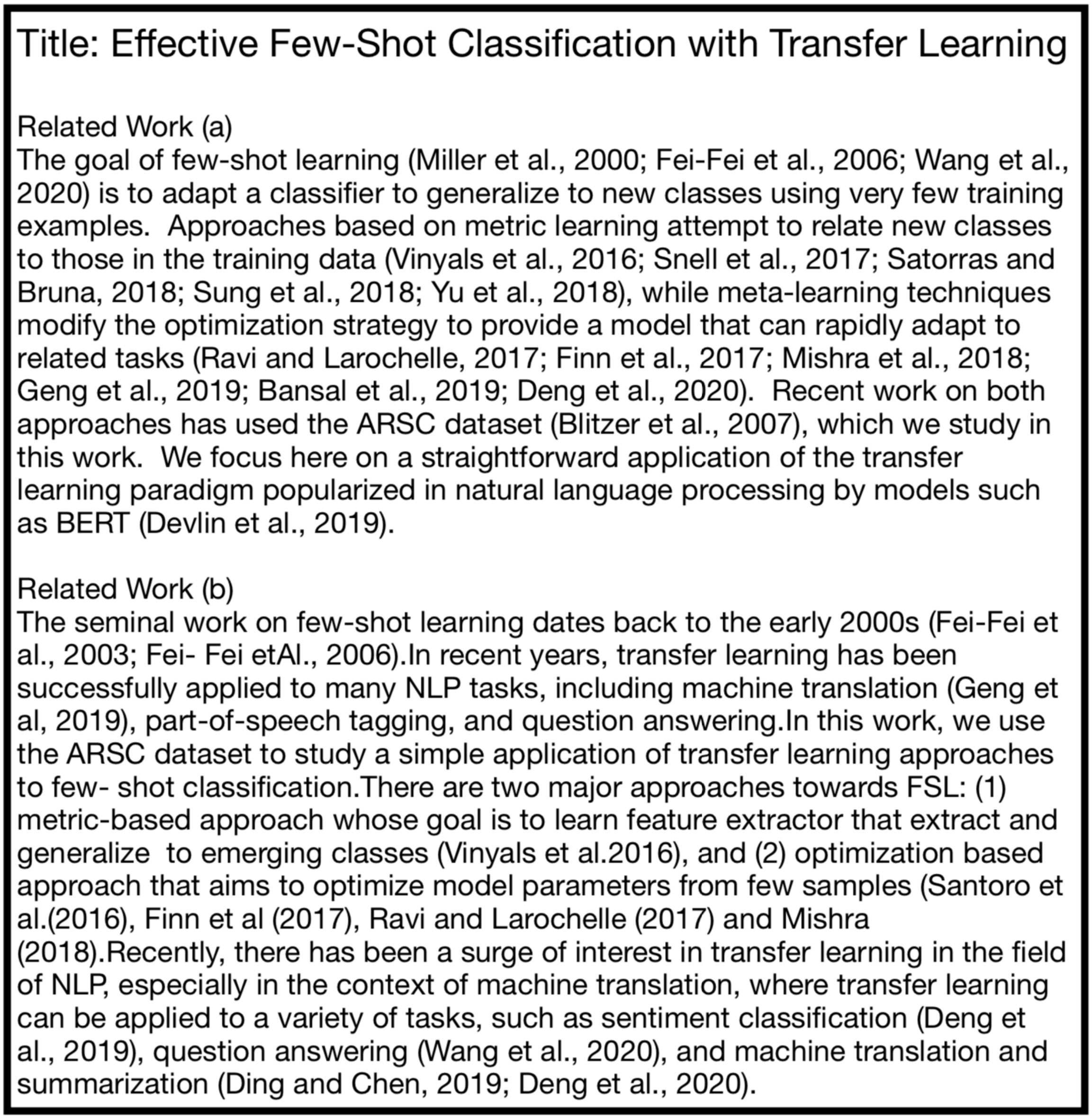
In this chapter, we study the generation of comparative summaries of past scholarly papers in context of a new research idea (query). Harnessing parallel data from an ACL Anthology based dataset and citation reason annotations, our model generates competent related work sections. Experimental results show our model outperforms state-of-the-art summarization methods in a generating realistic and motivation cognizant outputs.

4.1 Introduction

An essential component of scientific writing is positioning new research in the landscape of existing work. Commonly presented in a related work section, this comparison synthesizes information from multiple papers related to the current research. While identification of such papers can be partially automated[1], the related work section generation is yet to be automated. Here we are proposing an algorithm that can assist with the task. Figure 4-1 presents an example where our model generates a related work section for a new paper.

Writing the related work section can be viewed as a multi-document summarization task. However, most existing summarization approaches operate over input documents with significant content overlap such as news [49]. These techniques are not applicable to our task since we aim to highlight specific relations of each input

[1] `https://www.connectedpapers.com`



Title: Effective Few-Shot Classification with Transfer Learning

Related Work (a)
The goal of few-shot learning (Miller et al., 2000; Fei-Fei et al., 2006; Wang et al., 2020) is to adapt a classifier to generalize to new classes using very few training examples. Approaches based on metric learning attempt to relate new classes to those in the training data (Vinyals et al., 2016; Snell et al., 2017; Satorras and Bruna, 2018; Sung et al., 2018; Yu et al., 2018), while meta-learning techniques modify the optimization strategy to provide a model that can rapidly adapt to related tasks (Ravi and Larochelle, 2017; Finn et al., 2017; Mishra et al., 2018; Geng et al., 2019; Bansal et al., 2019; Deng et al., 2020). Recent work on both approaches has used the ARSC dataset (Blitzer et al., 2007), which we study in this work. We focus here on a straightforward application of the transfer learning paradigm popularized in natural language processing by models such as BERT (Devlin et al., 2019).

Related Work (b)
The seminal work on few-shot learning dates back to the early 2000s (Fei-Fei et al., 2003; Fei- Fei etAl., 2006).In recent years, transfer learning has been successfully applied to many NLP tasks, including machine translation (Geng et al, 2019), part-of-speech tagging, and question answering.In this work, we use the ARSC dataset to study a simple application of transfer learning approaches to few- shot classification.There are two major approaches towards FSL: (1) metric-based approach whose goal is to learn feature extractor that extract and generalize to emerging classes (Vinyals et al.2016), and (2) optimization based approach that aims to optimize model parameters from few samples (Santoro et al.(2016), Finn et al (2017), Ravi and Larochelle (2017) and Mishra (2018).Recently, there has been a surge of interest in transfer learning in the field of NLP, especially in the context of machine translation, where transfer learning can be applied to a variety of tasks, such as sentiment classification (Deng et al., 2019), question answering (Wang et al., 2020), and machine translation and summarization (Ding and Chen, 2019; Deng et al., 2020).

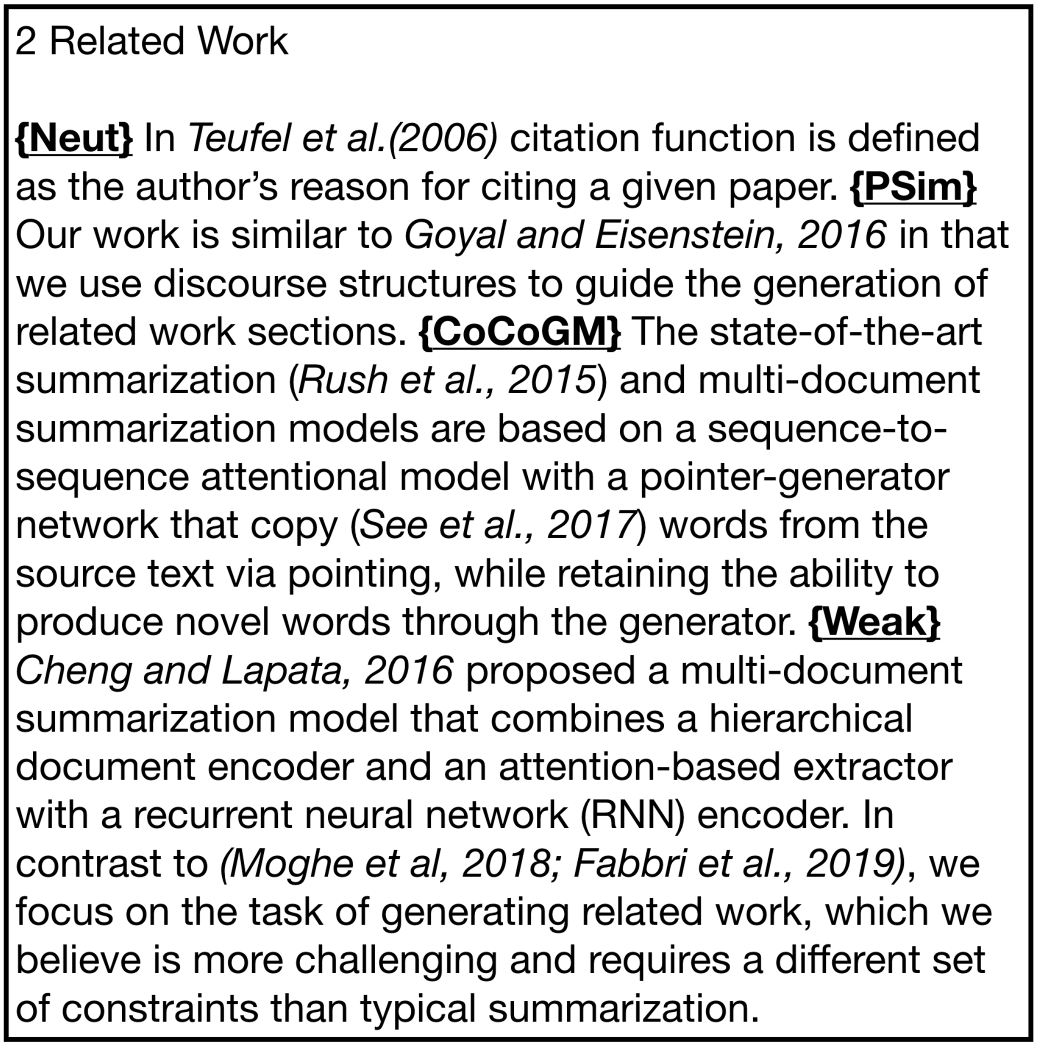
Figure 4-1: Two related work sections presented for the paper `https://www.aclweb.org/anthology/2020.coling-main.92/`.
Option (a) was produced by the authors and (b) was produced by our model.

article to the current discovery. Prior research in scientific discourse [79, 1] identified that reasons for citing papers fall into several argumentative classes such as reliance on previous results or gaps in existing solutions. Therefore, we can view the task of related work generation as predicting such reasons and conveying them in a coherent format.

We implement this approach in a traditional generation pipeline based on a content planner and surface realizer. To address the challenges of generating a long, comparative section our planner produces a skeleton to guide the subsequent text generation. Our planner takes in all available past papers and the abstract of the new paper. It generates a (depth=2) tree by predicting the individual reasons for citing past papers, and subsequently sorting selected papers and grouping them into respective branches with a combined reason for citing each branch. Our surface realization model iteratively generates a text span for every branch by fluently lexicalizing the reason behind citing the set of papers. A variety of such reasons are depicted in Figure 4-2 – similarity with past work (*PSim*), methodological comparisons (*CoCoGM*) or weakness (*Weak*) of past work are presented by our model output to describe related work.[2]

Our planning and generation strategy allows for refined control over the text spans. In particular, text segmentation annotations on related work sections, help organize the text to train our content planner and step-wise surface realizer. Furthermore, we utilize citation categorization annotations through distant supervision to learn a pairwise classification function between a pair of new and old papers – necessary for generating motivation cognisant segments. Our approach leaves enough scope for human intervention from an application perspective to modify the skeleton of cited papers or the generated related work section.

We apply our method for generating related work on an ACL Anthology dataset that we collect. Every related work section cites multiple scientific studies for varying reasons, making it extremely relevant to our task. We consider numerous studies to compare our method against state-of-the-art multi-document and query driven summarization techniques. Our approach receives the highest scores on both automatic and subjective evaluation metrics such as RougeL, BertScore, SARI and Relevance. For instance, on RougeL our approach gets an absolute 5% improvement over the best baseline. The method strongly outperforms others in an update setting, further highlighting its real world applicability in a scenario where a human written section is missing a few papers.



2 Related Work

{Neut} In *Teufel et al.(2006)* citation function is defined as the author's reason for citing a given paper. **{PSim}** Our work is similar to *Goyal and Eisenstein, 2016* in that we use discourse structures to guide the generation of related work sections. **{CoCoGM}** The state-of-the-art summarization (*Rush et al., 2015*) and multi-document summarization models are based on a sequence-to-sequence attentional model with a pointer-generator network that copy (*See et al., 2017*) words from the source text via pointing, while retaining the ability to produce novel words through the generator. **{Weak}** *Cheng and Lapata, 2016* proposed a multi-document summarization model that combines a hierarchical document encoder and an attention-based extractor with a recurrent neural network (RNN) encoder. In contrast to *(Moghe et al, 2018; Fabbri et al., 2019)*, we focus on the task of generating related work, which we believe is more challenging and requires a different set of constraints than typical summarization.

Figure 4-2: Model guided related work produced for the current paper using all ACL Anthology papers which we cite.

Our content planner's output branches [**{reason}**,(cited papers)] guide the paragraph generation by surface realizer. The **reason**s are described in Table 4.1.

4.2 Related Work

4.2.1 Multi-document Summarization

Neural sequence-to-sequence models [42, 43, 44] have become the standard for document summarization. Trained on large amounts of data, such methods have shown promise and have been successfully adapted for multi-document summarization [45, 46, 47, 48, 49]. This is achieved through manipulations in the input space, such as

[2] *PSim*, *CoCoGM*, *Weak* and *Neut* are described in Table 4.1.

concatenation through special tokens or modifications on the modeling side using hierarchical encoders [50]. These methods have improved upon traditional extractive [80, 41, 81, 82] and abstractive approaches [63, 83]. The key aspect of typical multi-document summarization solutions is to capture repetitions and similarities in the multiple input documents [84, 49]. However, in scientific writing, the goal is typically to identify and highlight differences with past work. In our past work [85], we address the consensus in inputs such as scientific studies when producing summaries. However, in related work generation, we are interested in generating a comparative section describing past works and their differences in context of the new idea.

4.2.2 Query Driven Summarization

Often a human written query forms the motivation for text generation and summarization, with the query forming the context and/or prompt for the overall writing. Such a formulation can be observed in tasks such as article writing [86], dialogue [87, 88, 89, 90], translation [91] and language modeling [92, 93]. The query can also be used to retrieve additional information to augment inputs – as observed in tasks such as question answering [94, 95], fact completion [96] and fact-checking [4].

Our work can also be considered as part of this framework, with the new work as query. However in contrast to aforementioned work, we aim to generate a coherent summary which highlights the comparative aspects of past papers.

4.2.3 Rhetorical Structure Theory

Rhetorical Structure Theory describes the structure of a document in terms of text spans that form discourse units and the relations between them [97] and is often used for summarization [98, 99]. For scientific writing, a flat structure of discourse units rather than a hierarchy has been observed [79]. Specifically, for generating related work we base our citation reasoning annotations on [1] to generate informative text. Prior work [1], defines twenty-six different reasons for citing papers, such as methodological differences, weakness in past approaches, similarity in usage or simply citing a paper for its legacy. These annotations are used, similar to recent aspect-oriented summarization approaches [100, 101], to produce motivation cognizant related work.

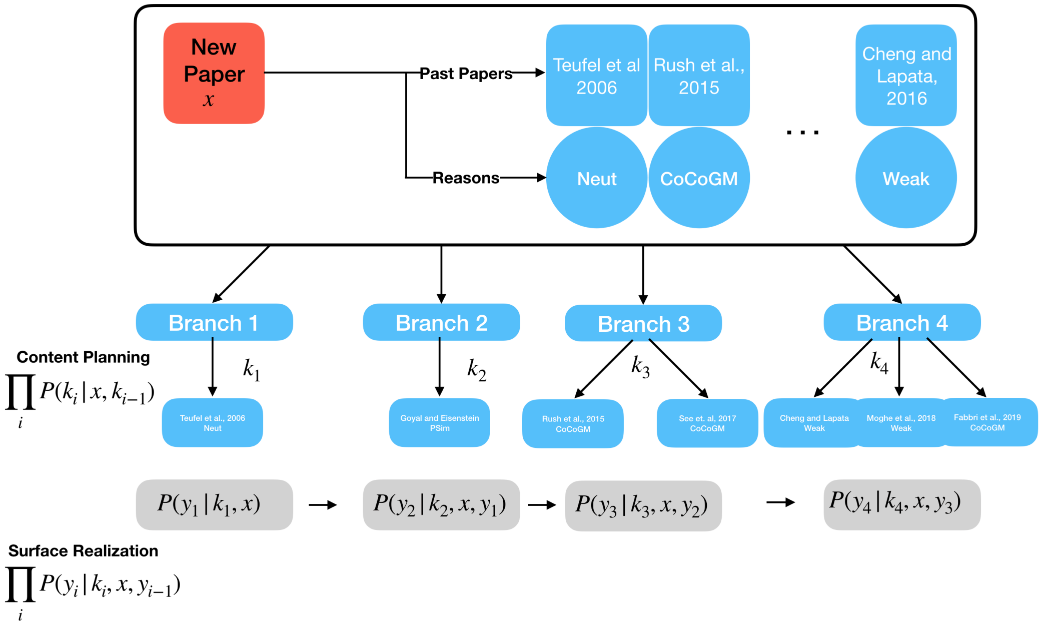


Figure 4-3: Illustrating the flow of our model. x is our paper's abstract and the past papers are the ones mentioned in Figure 4-2.
Content planning produces a tree k with four branches with respective citation reasons (*Neut*, *PSim*, *CoCoGM* and *Weak*.) Surface realization takes this tree to produce an output y.

4.3 Method

Our goal is to generate a related work section y for a new research idea (abstract) x. We organize past papers $\mathcal{C} = \{c_1, c_2, ...c_m\}$ and reasons $\mathcal{R} = \{r_1, r_2, ...r_m\}$ for citing them into a skeleton k to drive generation. The planning framework k enables the generation of citation reason aware and fluent related work sections. Our solution, illustrated in Figure 4-3, is described below.

Overview For each paper in the training corpus, we have its abstract x and the related work section to be generated y. Related work sections can be quite long, and in order to model their generation, we break y into segments $\{y_1, y_2, ..., y_n\}$ through crowd-sourced annotations. Segmentation annotations and subsequent motivation categorization of text give us our skeleton $k = \{k_1, k_2, ..., k_n\}$, composed of a grouping for cited papers and reasons for citing them.

The probability of an output summary y for a new abstract x by deriving a skeleton k is shown below. We make Markov assumptions and assuming each y_i depends on the specific k_i only.

$$P(y,k|x) = P(y|k,x)P(k|x) \tag{4.1}$$

$$P(y,k|x) = \prod_i P(y_i|k,x,y_{i-1}) \prod_j P(k_j|x,k_{j-1}) \tag{4.2}$$

$$P(y,k|x) \approx \prod_i P(y_i|k_i,x,y_{i-1}) \prod_j P(k_j|x,k_{j-1}) \tag{4.3}$$

allowing us to break the problem into two modules of content planning $\prod_j P(k_j|x,k_{j-1})$ and surface realization $\prod_i P(y_i|k_i,x,y_{i-1})$.

The model can be trained and applied in two settings: (1) The set of cited papers is known; (2) The full set of AA corpus is present and a relevant set must be selected through content planning. In addition, a trained model can be applied in an update setting.

4.3.1 Content Planning

Generating a long related work section for a new paper is aided by a detailed skeleton comprising of past papers to cite with reasons in context of the new paper. Through our content planner, we model this skeleton by generating a tree (depth=2) as depicted in the upper half of Figure 4-3. The segmentation annotations collected on the related work sections, form our supervision for this tree generation. Our content planning model takes a new paper's abstract x, a set of available papers to cite and produces a grouped ordering of papers into branches and a reason for citing each branch $k_i \in k$. For the set of papers to be cited $\mathcal{C} = \{c_1, c_2, ...c_m\}$ we produce a segmented realization $k = \{k_1, k_2, ..., k_n\}$ where a branch $k_i = (\{c^i_{a_1}, ..., c^i_{a_{n_i}}\}, r_i)$, with $\{c^i_{a_1}, ..., c^i_{a_{n_i}}\}$ the set of similar papers cited and r_i the reason for citing them. Table 4.1 mentions a subset of the reasons used in our approach, reported in [1].

At every step of our tree generation, the content planning model $\mathcal{F}$ decides whether a new branch k_{i+1} is to be created or a particular paper c_l should be added to the current branch k_i. Each paper c_l is represented by the encoding of its title and abstract $\hat{c_l} = e(c_l)$ and the new abstract x correspondingly $\hat{x} = e(x)$. We assign a default representation $\hat{c_\emptyset}$ for a new branch. Tree generation proceeds by considering all yet to be selected past papers and the current branch k_i, represented by the mean of representations of all papers selected in k_i so far : $\hat{b}$. The probability that a past paper c_l will be selected for inclusion in k_i is:

$$\pi_\theta(c_l) = \frac{\exp(\mathcal{F}(\hat{b}, \hat{x}, \hat{c_l}))}{\sum_j \exp(\mathcal{F}(\hat{b}, \hat{x}, \hat{c_j}))} \tag{4.4}$$

Tree generation continues until all papers are used or a maximum number of steps is reached. Each paper c_l has a reason r_l for citing it with respect to the new abstract x, predicted through a pair-wise classifier $r_l = f(x, c_l)$. After running the tree generation, each branch $\{c^i_{a_1}, ..., c^i_{a_{n_i}}\}$ gets a combined reason for citing based on an aggregation function $\mathcal{M}$ applied to individual reasons, $r_i = \mathcal{M}(r^i_{a_1}, ..., r^i_{a_{n_i}})$.[3]

The generated content planning tree is used by the surface planning model to write a related work section.

4.3.2 Surface Realization

After organizing and planning a skeleton of past papers, it is now critical to generate a fluent and reason aware related work section. The surface realization model $P(y|x, k)$ generates a coherent, informative and fluent summary y by taking in a new paper's abstract x and the content planning tree k from the previous step as input.

Specifically, we model the long text generation as a step-wise decoding task $P(y|x, k) = \prod_i P(y_i|k_i, x, y_{i-1})$, with a single segment y_i lexicalized every step – as depicted in the lower part of Figure 4-3. The segmentation annotations on the related work dataset create parallel data to train the realizer. The model takes the abstract of new research x, abstracts from $\{c^i_{a_1}, ..., c^i_{a_{n_i}}\}$, a token to represent the reason to cite them r_i and the text span y_{i-1} produced in the previous step. Here, $k_i = (\{c^i_{a_1}, ..., c^i_{a_{n_i}}\}, r_i)$ formulates a multi-document summarization task, controlled by a reason r_i. Furthermore, y_{i-1} from the previous step guides the generation of a continuous summary. The realizer is implemented using a Transformer based encoder-decoder model [76]. The various inputs k_i, y_{i-1} and x are separated using a special

[3] We consider $\mathcal{M}$ to be the max occurring reason from the set of individual reasons.

Category	Description
CoCoGM	Contrast in goal or method.
PSim	Author's work and cited work are similar.
Neut	Neutral description of cited work.
Weak	Weakness of cited approach.

Table 4.1: Subset (4 of 26) of reasons for citing a paper as described in [1].

token.

4.4 Dataset

In this section, we describe the dataset introduced to study our related work generation task.

AA: The ACL Anthology (AA) 2020 corpus contains papers on the study of natural language processing and computational linguistics. This corpora covers varied topic areas such as text classification, information extraction, generation, etc. We use this data dump and the corresponding text of the papers, to create our dataset of (paper, author list, title, abstract, related work section) tuples.

Paper Title and Citation: We collect the titles and corresponding lists of authors for all papers. The publication year and author list allow us to identify the acronym used to cite a paper in future works.

Abstract: We collect the abstracts for all papers, which form the description used to generate related work sections.

Related Work Section: We parse the related work section for papers in the AA corpus. These sections contain descriptions of past work, indicated explicitly through acronyms, highlighting the foundations and novelty of new papers.

Parallel Data: Considering papers with related work sections and available papers cited, we gather a reasonably sized parallel corpora of 8143 data points split into training, validation and testing sets. Papers published in or before the year 2019 are used for training and 185 and 202 papers from the year 2020 are used for validation and testing respectively.[4]

Segmentation Annotations: In order to model generating related work, we conduct crowd-sourced text segmentation annotations of the corresponding sections. Annotators are encouraged to identify atomic segments (one or more complete sentences) which convey unique information.

Citation Reason Annotations: In order to generate citation reason cognisant outputs, we leverage [1]. 26 different reasons for citing past papers are introduced in the paper (Table 4.1 explains a subset). We collect the corpora defined in this paper and train a text classification model to identify similar reasons on the current AA corpora. Using distant supervision, we use these reasons to collect a pairwise (new paper, cited paper) citation classification corpus.

4.5 Experiments

In this section, we describe the settings used to study the task, evaluation metrics, baselines for comparison and implementation details

Settings: We consider three settings to study the related work generation task.

Known Past Works: We use the gold set of cited papers to generate the related work section.

Full AA Dataset: We consider the full set of ACL Anthology papers and expect models to cite relevant ones to generate a related work section.

Related Work Update: We stimulate a related work modification, where an additional paper is to be cited in an otherwise well written related work section. This scenario is pervasive when authors miss a few references or a new paper is published during the time of writing.

[4] Data points considered for evaluation cite at least 15 past papers.

Evaluation Metrics: We evaluate our systems using the following automatic metrics. *Rouge* is an n-gram based automatic metric used to compare the model output with the gold reference [73].

BertScore is a contextualized embeddings based automatic metric used to compare the model output with the gold reference [102].

r Perplexity$^{\leftrightarrow}$ calculates the perplexity (mean of perplexity and reverse perplexity $\leftrightarrow$) of the reasoning (r) outputs inferred on the model summaries in context of those from human written related work sections in the training data.

SARI is a text update evaluation metric comparing the number of uni-grams added or kept compared to the gold update [38].

In addition to automatic evaluation, we have human annotators score our models on Relevance and Fluency for 100 data points per model.

Relevance: Indicates if the generated text shares similar information with the reference section.

Fluency: Represents if the generated text is grammatically correct and written in well-formed English.

Annotators rate relevance and fluency on a 1-5 likert scale [74]. We have 3 annotators score every data point and report the average across the scores.

Baselines: In order to demonstrate the effectiveness of our method, we compare it against several state-of-the-art multi-document summarization methods.

Copy-Gen: [44] is a summarization technique which can copy from the input or generate words and recently achieved best results on multi-document summarization [103].

Split-Encoder: [62] is a *two-encoder* decoder method for query driven summarization and citation text generation tasks [104].

MultiDocTransformer: [105] is a Transformer [76] implementation for multi-document summarization.

TransformerBART: [106] is a pre-trained state-of-the-art summarization model.

	Automatic Evaluation			Human Scores	
Model	RougeL	BertScore	r Perplexity$^{\leftrightarrow}$	Relevance	Fluency
Copy-Gen	0.17	0.60	19.8	3.47	**3.75**
Split-Encoder	0.19	0.59	18.8	3.50	3.66
MultiDocTransformer	0.14	0.52	19.8	3.54	3.66
TransformerBART	0.30	0.65	12.7	3.42	3.53
Ours	**0.32**	**0.66**	**10.5**	**3.65**	3.69

Table 4.2: Evaluation of related work generation when papers to be cited is known.

Implementation Details: Our tree generation model $\mathcal{F}$ is implemented as a feed forward neural network. We use TransformerBART for Surface Realization. We use a BERT [36] based sentence pair classifier for citation reason identification which takes in the abstracts of the new paper and the to be cited paper. We also use a BERT based related work segmentation classifier to preprocess the data. In the Full AA Setting baselines use a pretrained BERT retrieval model to select the top-n relevant papers to cite.

4.6 Results

In this section, we report the performance of our model and baselines on the gold papers known, full AA and update settings.

Gold Setting: To study our method's ability to produce realistic and citation reason driven text we run experiments when the gold set of papers to be cited is known. Table 4.2 reports the results of our model and all competing baselines in this scenario. Our model scores higher on *RougeL* and *BertScore* than the baselines, generating summaries syntactically and semantically similar to benchmark outputs. *r Perplexity$^{\leftrightarrow}$* scores capture realistic variability in reasons for citing papers. Our model receives the lowest perplexity score, highlighting its ability to plan and generate realistic (traditional perplexity) and diverse (reverse perplexity) sequences of citation reasons.

The *Copy-Gen*, *Split-Encoder*, *MultiDocTransformer* and *TransformerBART* methods focus on the repetitions amongst inputs and struggle to bring out detailed similarities and differences necessary in the related work as demonstrated in higher *r Perplexity$^{\leftrightarrow}$* scores.

Model	RougeL 10	RougeL 22
Copy-Gen	0.16	0.11
Split-Encoder	0.18	0.18
MultiDocTransformer	0.24	0.24
TransformerBART	0.24	0.24
Ours	**0.28**	**0.29**

Table 4.3: Evaluation of full AA database retrieval task.

Model	SARI	RougeL
Copy-Gen	0.28	0.24
TransformerBART	0.20	0.31
MultiDocBART	0.15	0.21
UpdateTransformerBART	0.15	0.22
Ours	**0.34**	**0.61**

Table 4.4: Evaluation on related work update task.

Model	Uni-gram	Bi-gram	Tri-gram	Four-gram
Copy-Gen	0.81	0.58	0.43	0.30
Split-Encoder	0.56	0.20	0.04	0.0
MultiDocTransformer	0.56	0.11	0.01	0.0
TransformerBART	0.76	0.44	0.30	0.24
Ours	0.78	0.39	0.21	0.15

Table 4.5: Fraction of n-grams copied.

Model	Corresponds to Reason
No-reason input	0.63
Standard	**0.65**

Table 4.6: Reason ablation.

Content Planning	Purity
K-Means	0.77
Ours	0.78

Table 4.7: Content planning clustering purity.

Human Evaluation: Table 4.2 also presents the human evaluation scores of the methods on the related work generation task. Our method is rated the highest by crowd-workers on *Relevance*, confirming the automatic evaluation metrics. Methods typically produce fluent text. The inter-annotator Kappa agreement [107] is 80% and 83% for *Relevance* and *Fluency* respectively.

Full AA Database Setting: To study the model's performance on an increasing number of cited papers we consider the setting where the model is not provided with the gold set of papers to be cited while generating the related work. Our model is trained to select, cluster and order papers for content planning. Table 4.3 reports our model performance in comparison with baselines. Our tree-segmented content planner can generate an appropriate skeleton of the papers making it easy for the surface realizer to produce summaries. Leading to a 5% RougeL improvement over the best baseline (in the setting where a maximum of 22 papers are cited). The competing baselines can not deal with the challenges of more papers and a growing input. We do not see improvements in their output quality when citing more papers – highlighting a fundamental limitation of *all-at-once* methods.

Update Setting: To study the model's ability to perform refined control we study an update setting. In this setting, an incomplete related work section is provided to the model. The model must modify the related work (with the data from the already cited works) using the missing cited paper to generate a complete related work. Table 4.4 reports the empirical evaluation of our model compared to baselines, including a setting specific *Update-Transformer* method. The flexibility of our tree-segmented content planning approach allows our model to keep most of the existing summary while updating only the requisite segments in a fluent manner to add the missing cited paper. On both SARI and RougeL our model outperforms the baselines by significant margins – absolute 6% on *SARI* and 30% on *RougeL*.

Case Study: Table 4.8 shows outputs for all the baselines and our model on the paper from Figure 4-1. *Copy-Gen* cites several papers, but doesn't give an informative summary of their contributions. Alternatively, *Split-Encoder*, while generating a longer section, does not capture the context of the new paper in terms of related work. *MultiDocTransformer* unfortunately doesn't generate a particularly relevant summary for this input. *TransformerBART* generates a paragraph which while being fluent, can't capture the entire pool of related work. In contrast, our model generates a fluent related work section, covering a lot of the relevant work. Our model also reports the task tackled in the current paper – *"In this work, we use the ARSC dataset to study a simple application of ... "*.

4.7 Analysis

We perform analysis to further study the Content Planner and Surface Realizer. We also perform subjective error analysis to explain mistakes made by our model.

Content Planning: Our content planning model allows for an organization of past papers cited in the related work generation. The tree-segmentation clusters similar papers and provides the reason for citing each cluster. Table 4.7 reports the purity score of the clusters using the tree generation method which is very comparable to a K-Means clustering method that does not provide an ordering of the segments. In addition, branching cited papers doesn't overwhelm the step-wise surface realization model with a very large number of input documents unlike the baselines.
We develop a BERT [36] sentence pair classifier to judge the reason for citing a past paper in context of a new paper. We use [1] citation reasoning annotations as distant supervision to produce training data for sentence pair classification (abstract pairs in this case), achieving an accuracy of 85% over 26 different classes.

Surface Realization: Our surface realization model, like baselines, generates a lot of new phrases as reported in Table 4.5. Our model also produces motivation cognizant outputs (Table 4.6). The 2% improvement compared to no reason input is very significant as the non *Neut* reasons are less frequent in the test set.

Text Segmentation: We develop a BERT [36] text segmentation model using MTurk annotations collected on 180 related work sections. This leads to 6000 data points for the binary classification of consecutive sentences into same or different segments. Our model performs this task at a reasonable accuracy of 71% .

Subjective Error Analysis: In order to get a detailed understanding of the kinds of errors made by our model, we perform subjective analysis on examples from our evaluation set. Specifically, we consider 50 examples which scored the lowest on the automatic RougeL metric. We categorize the errors made by our model in these cases into the following groups. Each group also has an associated number to denote the number of cases with the issue.

- *Small (20):* The generated output while relevant and close to the gold section, is small in length. This happens either because the content planning doesn't create enough branches or the surface realizer doesn't cover all the inputs.
- *Unrelated (14):* The generated output is not semantically similar to the human written section.
- *Unclear (4):* The generated output is not particularly legible.
- *Factual Error (4):* The generated output has factual errors.
- *Fine (8):* The generated outputs are semantically similar to the human written output and are fine despite low RougeL scores.

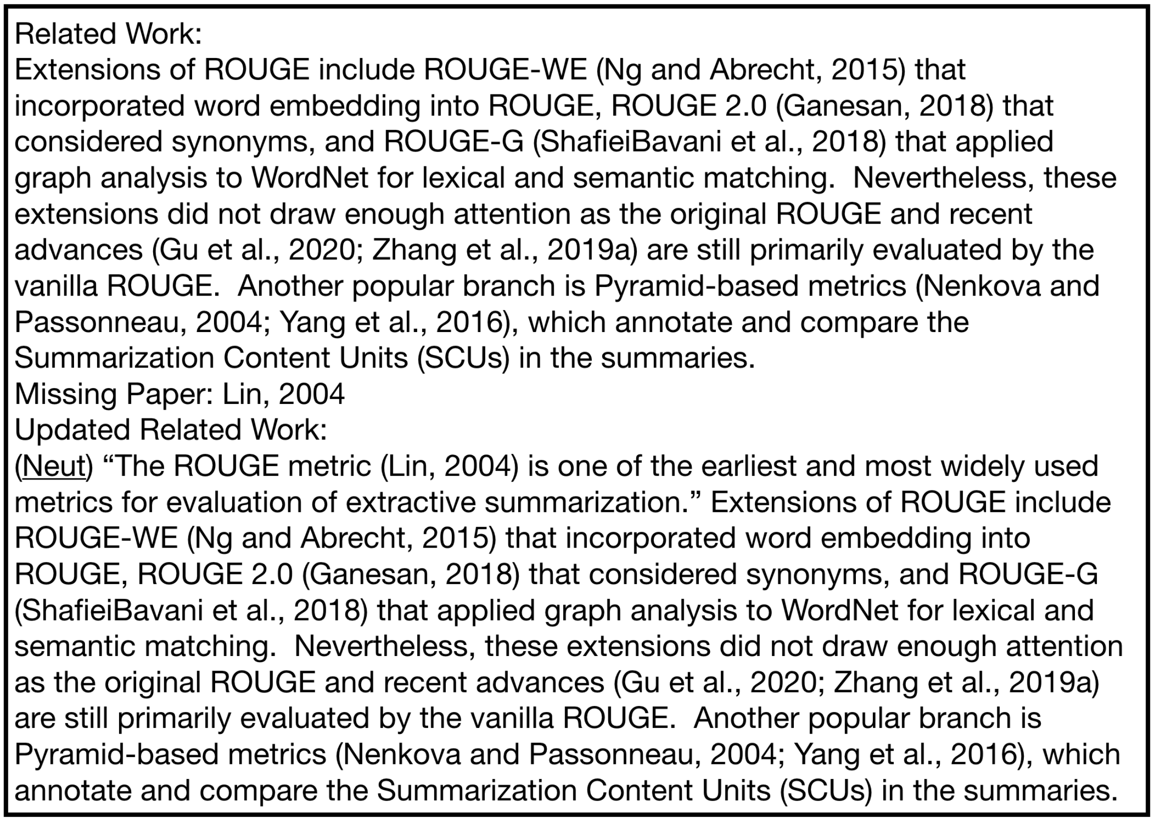
Model Improvements: Our model produces competent related work sections which outperform state-of-the-art summarization methods on both human and automatic evaluation. However, we do see scenarios where false text is generated. In order to avoid this, we recommend having a precise selection of content from reference papers while generating segments. The content planner would constrain the realizer about what phrases to generate, leading to fewer hallucinated phrases.

Update Example: We demonstrate a scenario where a human written related work missed a critical paper. When provided with the missing paper [73], our model finds an appropriate branch for the paper and the surface realization completes the writing (Figure 4-4).

Human Intervention: The related work generated by our model in Figure 4-2, we demonstrate how intervention can allow for a more pleasing content planning.

Copy-Gen There has been studied in early work such as (Fe-Fei et al., 2003; Fei-Fei et al., 2006) and more recent work (Ba et al., 2016; Santoro et al., 2016; Munkhdalai and Yu, 2017; Ravi and Larochelle, 2017; Ravi and Larochelle, 2016; Mishra et al., 2017; Sung et al., 2018; Allen et al., 2019). More recent work can be divided into two types: similarity-based methods (Vinyals et al., 2016; Snell et al., 2017; Sung et al., 2018) and optimization-based methods (Ravi and Larochelle, 2016; Munkhdalai and Yu, 2017).
Split-Encoder Approaches based on metric learning attempt to relate new classes to those in the training data (Vinyals et al., 2016; Snell et al., 2017; Ravi and Larochelle, 2017; Finn et al., 2017; Finn et al., 2017; Vinyals et al., 2016; Snell et al., 2017; Geng et al., 2018; Allen et al., 2018; Allen et al., 2018; Allen et al., 2019). Few-shot text techniques modify the optimization strategy to provide a model to few-shot classes (Vinyals et al., 2014; com/irlab/SmartCiteConBansal et al., 2016; Munkhdalai and Yu, 2017; Finn et al., 2017; Finn et al., 2017; Vinyals et al., 2016; Snell et al., 2017; Geng et al., 2018; Allen et al., 2018; Allen et al., 2018; Allen et al., 2019). The pioneering work in few-shot learning is that of few-shot text is that of informal texts such as discourse-level entailment, or the existence of the shared task (Verhagen et al., 2003; Miller et al., 2003; Wang et al., 2014). Few-shot learning techniques modify the optimization strategy to provide a model that can rapidly adapt to related tasks such as (Fe-Fei (Kim and Larochelle, 2017; Mishra et al., 2018; Yu et al., 2018), while meta-learning techniques modify the optimization (Ba and Larochelle, 2017; Miller et al., 2017).
MultiDocTransformer -based methods have been proposed for NLP tasks, such as NER (Zhang et al., 2016; Nallapineni and Nallum, 2017; NER, 2017). Recently, NER have been used for NER has been proposed, including NER and NLP (Zhou and Nenstette, 2017) and NNs (Zeng et alia and Nivreiter, 2015; Nenkova, 2016) have been applied to NER. In NER tasks, NLP applications such as machine translation (Zampieri et al.g., 2017) (Zadeh et al.(2018) have also been proposed to NMT (Zhao et al, 2017), NER.(2018), NLP community (Zhuber and Nirenburg, 2018; NLP) have shown to NNLP tasks (Zaman and Nadehong, 2018).
TransformerBART There are two major approaches towards FSL: (1) metric-based approach whose goal is to learn feature extractor that extract and generalize to emerging classes (Vinyals et al., 2016; Snell et al.(2017) and (2) optimization based approach that aims to optimize model parameters from few samples (Santoro et al, 2016; Finn etAl., 2017).In this work, we use the ARSC dataset to study a simple application of transfer learning approaches to few-shot classification.We train a single binary classifier to learn all fewshot classes jointly by prefixing class identifiers to the input text.
Our Method The seminal work on few-shot learning dates back to the early 2000s (Fei-Fei et al., 2003; Fei-Fei etAl., 2006).In recent years, transfer learning has been successfully applied to many NLP tasks, including machine translation (Geng et al, 2019), part-of-speech tagging, and question answering. In this work, we use the ARSC dataset to study a simple application of transfer learning approaches to few- shot classification. There are two major approaches towards FSL: (1) metric-based approach whose goal is to learn feature extractor that extract and generalize to emerging classes (Vinyals et al.2016), and (2) optimization based approach that aims to optimize model parameters from few samples (Santoro et al.(2016), Finn et al (2017), Ravi and Larochelle (2017) and Mishra (2018).Recently, there has been a surge of interest in transfer learning in the field of NLP, especially in the context of machine translation, where transfer learning can be applied to a variety of tasks, such as sentiment classification (Deng et al., 2019), question answering (Wang et al., 2020), and machine translation and summarization (Ding and Chen, 2019; Deng et al., 2020).

Table 4.8: Sample outputs for all models considered on the paper from Figure 4-1. While baselines produce fluent outputs, our model is most appropriate at reporting past work in context of the current problem.

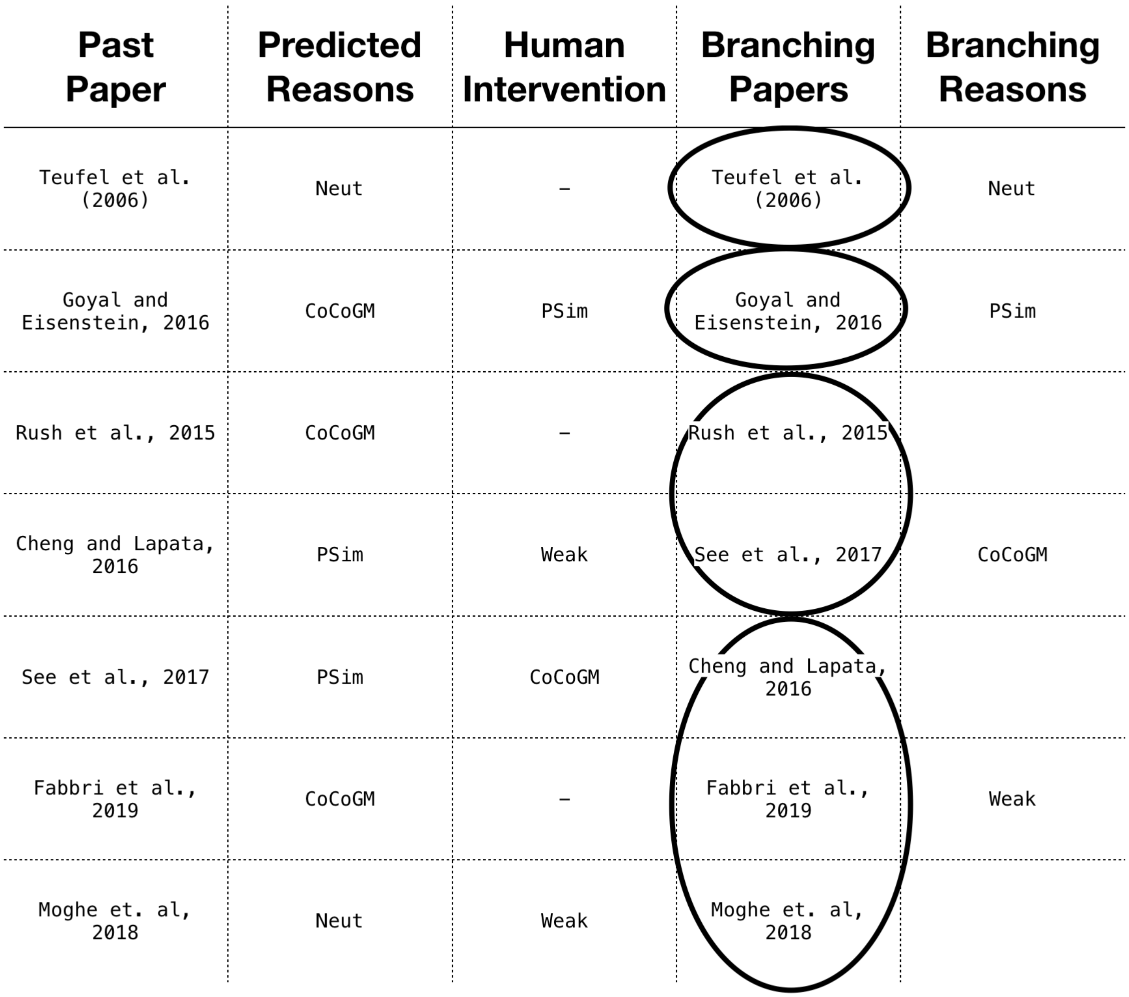


Related Work:
Extensions of ROUGE include ROUGE-WE (Ng and Abrecht, 2015) that incorporated word embedding into ROUGE, ROUGE 2.0 (Ganesan, 2018) that considered synonyms, and ROUGE-G (ShafieiBavani et al., 2018) that applied graph analysis to WordNet for lexical and semantic matching. Nevertheless, these extensions did not draw enough attention as the original ROUGE and recent advances (Gu et al., 2020; Zhang et al., 2019a) are still primarily evaluated by the vanilla ROUGE. Another popular branch is Pyramid-based metrics (Nenkova and Passonneau, 2004; Yang et al., 2016), which annotate and compare the Summarization Content Units (SCUs) in the summaries.
Missing Paper: Lin, 2004
Updated Related Work:
(Neut) “The ROUGE metric (Lin, 2004) is one of the earliest and most widely used metrics for evaluation of extractive summarization.” Extensions of ROUGE include ROUGE-WE (Ng and Abrecht, 2015) that incorporated word embedding into ROUGE, ROUGE 2.0 (Ganesan, 2018) that considered synonyms, and ROUGE-G (ShafieiBavani et al., 2018) that applied graph analysis to WordNet for lexical and semantic matching. Nevertheless, these extensions did not draw enough attention as the original ROUGE and recent advances (Gu et al., 2020; Zhang et al., 2019a) are still primarily evaluated by the vanilla ROUGE. Another popular branch is Pyramid-based metrics (Nenkova and Passonneau, 2004; Yang et al., 2016), which annotate and compare the Summarization Content Units (SCUs) in the summaries.

Figure 4-4: Model output for a summary update for an ammortized related work from the paper `https://www.aclweb.org/anthology/2020.acl-main.445/`.

Figure 4-5 shows the ease with which human intervention can be used to benefit the generation task. The initial reasons for the paper are quite valid, but the modifications allow for a better generation.

r **Perplexity**$^{\leftrightarrow}$**:** Captures the likelihood of reason sequences in model outputs being realistic. We use a trigram perplexity model to calculate perplexity. We perform a regular evaluation where a perplexity model is trained on the training (real) data and evaluated on model outputs. In order to ensure that simply trivial solutions such as ($\{Neut,Neut\}$ are down scored, we also perform reverse perplexity, where the likelihood of the training data is measured with respect to the model outputs.



Past Paper	Predicted Reasons	Human Intervention	Branching Papers	Branching Reasons
Teufel et al. (2006)	Neut	–	Teufel et al. (2006)	Neut
Goyal and Eisenstein, 2016	CoCoGM	PSim	Goyal and Eisenstein, 2016	PSim
Rush et al., 2015	CoCoGM	–	Rush et al., 2015	
Cheng and Lapata, 2016	PSim	Weak	See et al., 2017	CoCoGM
See et al., 2017	PSim	CoCoGM	Cheng and Lapata, 2016	
Fabbri et al., 2019	CoCoGM	–	Fabbri et al., 2019	Weak
Moghe et. al, 2018	Neut	Weak	Moghe et. al, 2018	

Figure 4-5: Human intervention to produce a more relevant content planning.

4.8 Discussion

While generating long texts, content control is essential. This is particularly evident in our study of generating related work which we first model as a tree generation task. This content planning model forms a strong skeleton for coherent and citation reason specific generation. Subsequent lexicalization through the surface realization model produces outputs which outperform those from state-of-the-art methods. This is confirmed by both automatic and human evaluation.

Human Association in Application: Separating the solution into planning and generation welcomes human intervention in the writing process. Our reason classification receives an accuracy of 85% and content planning grouping purity is 78%. While this performance is promising, the solution also allows for explicit guidance from humans to improve the final output (as shown in Figure 4-5). Text generation through the surface realization model achieves a state-of-the-art 32% RougeL F1 score. In a few cases, it may generate phrases which are not factually valid – prudent human validation is encouraged while using such a system.

A human written related work section, on having to accommodate missing papers, would need updating. This forms another scenario for our model to be used to augment an already strong human written summary, as shown in Figure 4-4. Our results in this scenario (Table 4.4) are promising.

Chapter 5

Conclusion

In this thesis we have developed summarization models that can address and highlight the contradictions amongst inputs. In particular, we utilize a framework of content planning and surface realization to tackle three novel settings which are rife with contrastive text.

In the scenario of factual update where an old text is updated by a guiding claim, our content planner identifies polarizing components in the original sentence and masks them. Then, using the residual sentence and the guiding claim, the surface realizer generates a new sentence which is consistent with the claim. The entire method is trained using indirect supervision from a fact-checking dataset. When applied to a Wikipedia fact update evaluation set, my method successfully generates correct Wikipedia sentences using the guiding claims. Furthermore, when used for data augmentation, our method is able to boost the performance of a biased fact-checking dataset – demonstrating the broad applicability of our system.

For the consensual multi-document summarization setting, where input documents do not have complete consensus, our planner identifies the key information to be summarized and identifies the degree of agreement amongst them. Our surface realizer takes in the key structures and aggregation consensus to generate summaries which are faithful to content and cognizant of the consensus in input documents. Compared to previous methods, which fail to capture the degree of consensus, especially in low-resource scenarios, our method is extremely competent in summarization and updating. The planner and realizer maximize the available resources by leveraging the limited parallel corpora and unlabelled text respectively.

In the query driven comparative summarization scenario, studied for generating related work sections, our planner identifies reasons for citing each input with respect to a new idea. It produces a tree of past papers to be cited along with grouped reasons. Our realizer, takes this tree to generate coherent text which is cognisant of the relation between the query and referenced articles. This in turn outperforms state-of-the-art methods which struggle to deal with increasing inputs and are unable to highlight relations of past work with the current research.

5.1 Ethical Considerations

Factual updates using guiding claims, summarizing contradicting studies and contrasting past work with new research ideas are all important task which require significant human effort. While models developed in this thesis may be used for the specific applications of encyclopedia updates, summarizing nutritional studies or writing related sections for upcoming Natural Language Processing papers, the primary goal of the thesis is to model challenging comparative generation tasks. If our models are applied in the real world, the utmost caution must be applied to ensure no facts are fabricated, no health impact is misconstrued or no past scholarly paper is misconceived. Such mistakes can have a negative impact on particularly sacrosanct tasks of summarizing nutritional studies or surveying fellow researchers' work.

5.2 Future Work

We hope the work presented in thesis can inspire more research. Our work can be extended in a number of ways, as discussed below:

Code-mixed Generation Our fact-guided sentence modification demonstrates the ability to fuse two distinct pieces of text in producing a new sentence which has the style of an input and the content of the other input. In a multi-lingual setting this could have phenomenal applications in the ability to produce code-mixed sentences from mono-lingual sentences. A content planning model can identify phrases in input sentences to be translated and a multi-lingual surface realizor can take the two inputs to generate a code-mixed sentence. Such a model can allow the generation of high-quality synthetic code-mixed data.

Augmenting Hate Speech Datasets Hate speech detection is a task which is of pressing importance in the current social and political environment. While it is beginning to get studied in the context of the Western World, hate speech detection is still under-studied for several countries and languages. In such scenarios, it is critical to produce datasets for a fair detection of hate speech on social media websites. Data augmentation, to produce synthetic data solely for the purpose of better and broader detection could be necessary. A good content planning model would be required to realize various categories of hate.

Iterative Fact-Correcting Generation Typical summarization systems and the ones explored in this thesis are one-shot in nature, where the model generation is is presented as the final output. We explore several constrains to ensure the generation systems do not hallucinate to produce false texts. An orthogonal approach is to consider a paradigm of iterative modifications to the text, by leveraging a domain specific fact-checking system to identify and correct factual mistakes in the outputs. Such an approach would entail incorporating intermediate feedback from a fact-checking model and guide this iterative writing process through a policy driven by the gold summary.

Extensive Rules Driven Generation Our consensual multi-document summarization system assumes simple rules for consensus aggregation. While this works very well in a low-resource domain, in scenarios where we have more parallel data we should explore learning and modeling complex rules for refined cases. This would increase the breadth of consensus captured and produce even more nuanced outputs, while still being interpretable.

Faithful Related Work Generation Our query driven comparative summarization method generates competent related work sections from papers. While this is extremely valuable, a next step would be to completely eliminate false text generated by the current model in a few scenarioes. This can be achieved by further constraining the data to be presented to the generation system. The relation classification between the new paper and old paper can incorporate a rationale extraction model to identify the critical spans to highlight for writing related work. The rationale model can then allow an interaction between the relation classification and generation models, further optimizing them.

Chapter 6

Appendix A

6.1 Creating a Debiased Evaluation Set for Fever

An unbiased verification dataset should exclude ‘give-away’ phrases in one of its inputs and also not allow the system to solely rely on world knowledge. The dataset should enforce models to validate the claim with respect to the retrieved evidence. Particularly, the truth of some claims might change as the evidence varies over time.

For example, the claim *“Halep failed to ever win a Wimbledon title”* was correct until July 19. A fact-checking system that retrieves information from Halep’s Wikipedia page should modify its answer to “false” after the update that includes information about her 2019 win.

Towards this goal, we create a Symmetric Test Set. For an original claim-evidence pair, we manually generate a synthetic pair that holds the same relation (i.e. Supports or Refutes) while expressing a fact that contradicts the original sentences. Combining the Original and Generated pairs, we obtain two new cross pairs that hold the inverse relations (see Figure 6-1). Examples of generated sentences are provided in Table 6.1.

This new test set completely eliminates the ability of models to rely on cues from claims. Considering the two labels of this test set[1], the probability of a label given the existence of any n-gram in the claim or in the evidence is $p(l|w) = 0.5$, by construction.

[1] Not Enough Info cases are easy to generate so we focus on the two other labels.

Source	Claim	Evidence	Label
Original	Tim Roth is an English actor.	Timothy Simon Roth (born 14 May 1961) is an English actor and director.	SUPPORTS
Generated	Tim Roth is an American actor.	Timothy Simon Roth (born 14 May 1961) is an American actor and director.	SUPPORTS
Original	Aristotle spent time in Athens.	At seventeen or eighteen years of age, he joined Plato's Academy in Athens and remained there until the age of thirty-seven (c. 347 BC).	SUPPORTS
Generated	Aristotle did not visit Athens.	At seventeen or eighteen years of age, he missed the opportunity to join Plato's Academy in Athens and never visited the place.	SUPPORTS
Original	Telemundo is a English-language television network.	Telemundo (telemundo) is an American Spanish-language terrestrial television network owned by Comcast through the NBCUniversal division NBCUniversal Telemundo Enterprises.	REFUTES
Generated	Telemundo is a Spanish-language television network.	Telemundo (telemundo) is an American English-language terrestrial television network owned by Comcast through the NBCUniversal division NBCUniversal Telemundo Enterprises.	REFUTES
Original	Magic Johnson did not play for the Lakers.	He played point guard for the Lakers for 13 seasons.	REFUTES
Generated	Magic Johnson played for the Lakers.	He played for the Giants and no other team.	REFUTES

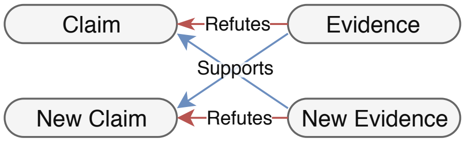
Table 6.1: Examples of pairs from the Symmetric Dataset.
Each generated claim-evidence pair holds the relation described in the right column. Crossing the generated sentences with the original ones creates two additional cases with an opposite label (see 6-1).

Also, as the example in 6-1 demonstrates, in order to perform well on this dataset, a fact verification classifier may still take advantage of world knowledge (e.g. geographical locations), but reasoning should only be with respect to the context.

6.2 Instance Re-weighting Algorithm for Debiasing

We propose an algorithmic solution to alleviate the bias introduced by 'give-away' n-grams present in the claims. We re-weight the instances in the dataset to flatten the correlation of claim n-grams with respect to the labels. Specifically, for 'give-away' phrases of a particular label, we increase the importance of claims with different labels containing those phrases.

We assign an additional (positive) balancing weight $\alpha^{(i)}$ to each training example $\{x^{(i)}, y^{(i)}\}$, determined by the words in the claim.



(A) Original pair from the FEVER dataset

Claim:
Stanley Williams stayed in Cuba his whole life.
Evidence:
Stanley [...] was part of the West Side Crips, a street gang which has its roots in South Central Los Angeles.

(B) Manually Generated pair

Claim:
Stanley Williams moved from Cuba to California when he was 15 years old.
Evidence:
Stanley [...] was born in Havana and didn't leave the country until he died.

Figure 6-1: An illustration of a Refutes claim-evidence pair from the FEVER dataset (A) that is used to generate a new pair (B).
From the combination of the Original and manually Generated pairs, we obtain a total of four pairs creating symmetry.

Bias in the Re-Weighted Dataset For each n-gram w_j in the vocabulary V of the claims, we define the bias towards class c to be of the form:

$$b_j^c = \frac{\sum_{i=1}^{n} I_{[w_j^{(i)}]}(1+\alpha^{(i)}) I_{[y^{(i)}=c]}}{\sum_{i=1}^{n} I_{[w_j^{(i)}]}(1+\alpha^{(i)})}, \tag{6.1}$$

where $I_{[w_j^{(i)}]}$ and $I_{[y^{(i)}=c]}$ are the indicators for w_j being present in the claim from $x^{(i)}$ and label $y^{(i)}$ being of class c, respectively.

Model	BASE	R.W
ESIM	55.9	59.3
BERT	58.3	**61.6**

Table 6.2: Impact of instance re-weighting algorithm on de-biasing.

Optimization of the Overall Bias Finding the α values which minimize the bias leads us to solving the following objective:

$$\min \left(\sum_{j=1}^{|V|} \max_{c}(b_j^c) + \lambda \|\vec{\alpha}\|_2 \right). \tag{6.2}$$

Re-Weighted Training Objective We calculate the α values separately from the model optimization, as a pre-processing step, by optimizing (6.5). Using these values, the training objective is re-weighted from the standard $\sum_{i=1}^{n} L(x^{(i)}, y^{(i)})$ to

$$\sum_{i=1}^{n} (1 + \alpha^{(i)}) L(x^{(i)}, y^{(i)}). \tag{6.3}$$

Evaluation The re-weighting method increases the accuracy of the ESIM and BERT models by an absolute 3.4% and 3.3% respectively (see Table 6.2). One can notice that this improvement comes at a cost in the accuracy over the FEVER Dev pairs. Again, this can be explained by the bias in the training data that translates to the development set, allowing FEVER-trained models to leverage it. Applying the regularization method, using the same training data, helps to train a more robust model that performs better on our test set, where verification in context is a key requirement.

6.3 Theoretical Perspective

The goal is to learn a classifier f on $D = (x^i, y^i)_{i=1}^{N}$, where $x^i = (x^i_{ref}, x^i_{hyp})$ and apply f to an unseen and unbiased evaluation set. We wish to minimize the following objective:

$$f_t^* = \underset{f \in H}{argmin} \sum_{(x,y) \in X \times Y} P_t(x,y) L(x,y,f)$$

Since we do not have any supervision in the target domain, we introduce the source domain:

$$\begin{aligned} f_t^* &= \underset{f \in H}{argmin} \sum_{(x,y) \in X \times Y} \frac{P_t(x,y)}{P_s(x,y)} P_s(x,y) L(x,y,f) \\ &\approx \underset{f \in H}{argmin} \sum_{(x,y) \in X \times Y} \frac{P_t(x,y)}{P_s(x,y)} \tilde{P}_s(x,y) L(x,y,f) \\ &= \underset{f \in H}{argmin} \frac{1}{N_s} \sum_{i=1}^{N_s} \frac{P_t(x^i,y^i)}{P_s(x^i,y^i)} L(x^i,y^i,f) \end{aligned}$$

Taking into account the multi-input function.

$$f_t^* = \underset{f \in H}{argmin} \frac{1}{N_s} \sum_{i=1}^{N_s} \frac{P_t(x_{ref}^i, x_{hyp}^i, y^i)}{P_s(x_{ref}^i, x_{hyp}^i, y^i)} L(x_{ref}^i, x_{hyp}^i, y^i, f)$$

For one sentence of the input and the corresponding label, the probability for the second input sentence is the same irrespective of the source or target distribution, since they represent the same function.

$$f_t^* = \underset{f \in H}{argmin} \frac{1}{N_s} \sum_{i=1}^{N_s} \frac{P_t(x_{hyp}^i, y^i)}{P_s(x_{hyp}^i, y^i)} \cancel{\frac{P_t(x_{ref}^i | x_{hyp}^i, y^i)}{P_s(x_{ref}^i | x_{hyp}^i, y^i)}} L(x_{ref}^i, x_{hyp}^i, y^i, f)$$

Now, we set $\frac{P_t(x_{hyp}^i, y^i)}{P_s(x_{hyp}^i, y^i)} = \beta^i$, to get an instance weighted formulation:

$$f_t^* = \underset{f \in H}{argmin} \sum_{i=1}^{N_s} \beta_i L(x_{ref}^i, x_{hyp}^i, y^i, f)$$

We set $\beta^i = \alpha^i + \epsilon$, with ϵ being a constant and learn α^i as described below.

6.3.1 De-biasing algorithm

We use the inductive knowledge provided to us for the target domain to estimate α^is.

We re-weight the instances in the dataset to flatten the correlation of x_{hyp} n-grams with respect to y. Specifically, for 'give-away' phrases of a particular label, we increase the importance of x_{hyp}'s with different labels containing those phrases.

We assign an additional (positive) balancing weight $\alpha^{(i)}$ to each training example $\{x^{(i)}, y^{(i)}\}$, determined by the words in x_{hyp}.

Bias in the Re-Weighted Dataset For each n-gram w_j in the vocabulary V of $x_2^i \; \forall i \in N_s$, we define the bias towards class c to be of the form:

$$b_j^c = \frac{\sum_{i=1}^{N_s} I_{[w_j^{(i)}]}(\epsilon + \alpha^{(i)}) I_{[y^{(i)}=c]}}{\sum_{i=1}^{N_s} I_{[w_j^{(i)}]}(\epsilon + \alpha^{(i)})}, \tag{6.4}$$

where $I_{[w_j^{(i)}]}$ and $I_{[y^{(i)}=c]}$ are the indicators for w_j being present in $x_{hyp}^{(i)}$ and label $y^{(i)}$ being of class c, respectively.

Optimization of the Overall Bias Finding the α values which minimize the bias leads us to solving the following objective:

$$\min \left(\sum_{j=1}^{|V|} \max_c (b_j^c) + \lambda \|\vec{\alpha}\|_2 \right). \tag{6.5}$$

6.3.2 Implication of learned weights

Having learnt the α's in the previous step, we set the β values in our loss function to find f_t^*. We now analyze β^i in context of biased x_{hyp}^i.

$$\beta^i = \frac{P_t(x_{hyp}^i, y^i)}{P_s(x_{hyp}^i, y^i)}$$

[108] proposed a kernel mean matching method to address the sample selection bias problem. To solve the following optimization problem to solve the corresponding

β's:

$$\{\beta^i\}_{i=1}^{N_s} = \underset{\beta^i}{argmin} \mid\mid \frac{1}{N_s} \sum_{i=1}^{N_s} \beta_i F_s(x_{hyp}^i, y^i) - \frac{1}{N_t} \sum_{i=1}^{N_t} F_t(x_{hyp}^i, y^i) \mid\mid$$

However, since we don't have any target domain data (we are considering an inductive setting), we solve a necessary condition i.e., assuming $F(x_{hyp}, y)$ to be represented by n-gram (bigram) features.

$$F(x_{hyp}, y) = \begin{pmatrix} I(1, b_1) & I(1, b_2) & \cdots & I(1, b_{|V|}) \\ I(2, b_1) & I(2, b_2) & \cdots & I(2, b_{|V|}) \\ \vdots & \vdots & \ddots & \vdots \\ I(\mid C \mid, b_1) & I(\mid C \mid, b_2) & \cdots & I(\mid C \mid, b_{|V|}) \end{pmatrix}$$

Where $I(c, b)$ is 1 if $b \in x_{hyp}$ and $c == y$, else 0.

We know the inductive knowledge of the class distributions in the target domain with respect to x_{hyp}'s, we optimize the β's such that each column of the feature mapping is uniform (un-biased). The algorithm is solving these necessary conditions.

Bibliography

[1] Simone Teufel, Advaith Siddharthan, and Dan Tidhar. Automatic classification of citation function. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 103–110, Sydney, Australia, July 2006. Association for Computational Linguistics.

[2] Dragomir R Radev. Generating natural language summaries from multiple online sources. 1997.

[3] Tao Lei, Regina Barzilay, and Tommi Jaakkola. Rationalizing neural predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 107–117, Austin, Texas, November 2016. Association for Computational Linguistics.

[4] James Thorne and Andreas Vlachos. Automated fact checking: Task formulations, methods and future directions. *ICCL*, pages 3346–3359, August 2018.

[5] Tao Lei, Regina Barzilay, and Tommi Jaakkola. Rationalizing neural predictions. *EMNLP*, pages 107–117, November 2016.

[6] Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. Style transfer from non-parallel text by cross-alignment. *NIPS*, pages 6830–6841, 2017.

[7] Zhirui Zhang, Shuo Ren, Shujie Liu, Jianyong Wang, Peng Chen, Mu Li, Ming Zhou, and Enhong Chen. Style transfer as unsupervised machine translation. *arXiv preprint arXiv:1808.07894*, 2018.

[8] Wei-Fan Chen, Henning Wachsmuth, Khalid Al Khatib, and Benno Stein. Learning to flip the bias of news headlines. *ICNLG*, pages 79–88, November 2018.

[9] Regina Barzilay and Kathleen R McKeown. Sentence fusion for multidocument news summarization. *Computational Linguistics*, 31(3):297–328, 2005.

[10] Shashi Narayan, Claire Gardent, Shay B. Cohen, and Anastasia Shimorina. Split and rephrase. *EMNLP*, pages 606–616, September 2017.

[11] Mor Geva, Eric Malmi, Idan Szpektor, and Jonathan Berant. DiscoFuse: A large-scale dataset for discourse-based sentence fusion. *NAACL HLT*, pages 3443–3455, June 2019.

[12] Juncen Li, Robin Jia, He He, and Percy Liang. Delete, retrieve, generate: a simple approach to sentiment and style transfer. *NAACL HLT*, pages 1865–1874, June 2018.

[13] Johannes Daxenberger and Iryna Gurevych. Automatically classifying edit categories in Wikipedia revisions. *EMNLP*, pages 578–589, October 2013.

[14] Diyi Yang, Aaron Halfaker, Robert Kraut, and Eduard Hovy. Identifying semantic edit intentions from revisions in Wikipedia. *EMNLP*, pages 2000–2010, September 2017.

[15] Manaal Faruqui, Ellie Pavlick, Ian Tenney, and Dipanjan Das. WikiAtomicEdits: A multilingual corpus of Wikipedia edits for modeling language and discourse. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 305–315, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[16] Mark Yatskar, Bo Pang, Cristian Danescu-Niculescu-Mizil, and Lillian Lee. For the sake of simplicity: Unsupervised extraction of lexical simplifications from Wikipedia. *NAACL*, pages 365–368, June 2010.

[17] Aurélien Max and Guillaume Wisniewski. Mining naturally-occurring corrections and paraphrases from Wikipedia's revision history. *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, May 2010.

[18] Aoife Cahill, Nitin Madnani, Joel Tetreault, and Diane Napolitano. Robust systems for preposition error correction using Wikipedia revisions. *NAACL HLT*, pages 507–517, June 2013.

[19] Andreas Vlachos and Sebastian Riedel. Fact checking: Task definition and dataset construction. *ACL Workshop on Language Technologies and Computational Social Science*, pages 18–22, June 2014.

[20] William Yang Wang. "liar, liar pants on fire": A new benchmark dataset for fake news detection. *ACL*, pages 422–426, July 2017.

[21] Hannah Rashkin, Eunsol Choi, Jin Yea Jang, Svitlana Volkova, and Yejin Choi. Truth of varying shades: Analyzing language in fake news and political fact-checking. *EMNLP*, pages 2931–2937, September 2017.

[22] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. FEVER: a large-scale dataset for fact extraction and VERification. *NAACL*, pages 809–819, June 2018.

[23] Tal Schuster, Darsh J Shah, Yun Jie Serene Yeo, Daniel Filizzola, Enrico Santus, and Regina Barzilay. Towards debiasing fact verification models. *EMNLP-IJCNLP*, 2019.

[24] Luis Perez and Jason Wang. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*, 2017.

[25] Sosuke Kobayashi. Contextual augmentation: Data augmentation by words with paradigmatic relations. *NAACL HLT*, pages 452–457, 2018.

[26] Xing Wu, Shangwen Lv, Liangjun Zang, Jizhong Han, and Songlin Hu. Conditional bert contextual augmentation. *arXiv preprint arXiv:1812.06705*, 2018.

[27] Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. Adversarial example generation with syntactically controlled paraphrase networks. *NAACL HLT*, pages 1875–1885, June 2018.

[28] Yuan Zhang, Jason Baldridge, and Luheng He. Paws: Paraphrase adversaries from word scrambling. *arXiv preprint arXiv:1904.01130*, 2019.

[29] Jin-Hwa Kim, Jaehyun Jun, and Byoung-Tak Zhang. Bilinear attention networks. *NIPS*, pages 1564–1574, 2018.

[30] Abigail See, Peter J. Liu, and Christopher D. Manning. Get to the point: Summarization with pointer-generator networks. *ACL*, pages 1073–1083, July 2017.

[31] Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. AllenNLP: A deep semantic natural language processing platform. *Workshop for NLP-OSS*, pages 1–6, July 2018.

[32] Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. Enhanced lstm for natural language inference. *ACL*, pages 1657–1668, 2017.

[33] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. *EMNLP*, pages 1532–1543, October 2014.

[34] Haşim Sak, Andrew Senior, and Françoise Beaufays. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. *ISCA*, 2014.

[35] Mitchell Stern, Jacob Andreas, and Dan Klein. A minimal span-based neural constituency parser. *ACL*, pages 818–827, July 2017.

[36] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[37] John Wieting and Kevin Gimpel. ParaNMT-50M: Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. *ACL*, pages 451–462, July 2018.

[38] Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. Optimizing statistical machine translation for text simplification. *TACL*, 4:401–415, December 2016.

[39] Kyle Swanson, Lili Yu, and Tao Lei. Rationalizing text matching: Learning sparse alignments via optimal transport. *arXiv preprint arXiv:2005.13111*, 2020.

[40] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners.

[41] Dragomir R. Radev and Kathleen R. McKeown. Generating natural language summaries from multiple on-line sources. *Computational Linguistics*, 24(3):469–500, 1998.

[42] Alexander M. Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[43] 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.

[44] 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.

[45] 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 2018. Association for Computational Linguistics.

[46] Logan Lebanoff, Kaiqiang Song, and Fei Liu. Adapting the neural encoder-decoder framework from single to multi-document summarization. *arXiv preprint arXiv:1808.06218*, 2018.

[47] Tal Baumel, Matan Eyal, and Michael Elhadad. Query focused abstractive summarization: Incorporating query relevance, multi-document coverage, and summary length constraints into seq2seq models. *arXiv preprint arXiv:1801.07704*, 2018.

[48] Reinald Kim Amplayo and Mirella Lapata. Informative and controllable opinion summarization. *arXiv preprint arXiv:1909.02322*, 2019.

[49] Alexander Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1074–1084, Florence, Italy, July 2019. Association for Computational Linguistics.

[50] Yang Liu and Mirella Lapata. Hierarchical transformers for multi-document summarization. *arXiv preprint arXiv:1905.13164*, 2019.

[51] Ratish Puduppully, Li Dong, and Mirella Lapata. Data-to-text generation with entity modeling. *arXiv preprint arXiv:1906.03221*, 2019.

[52] Wojciech Kryściński, Bryan McCann, Caiming Xiong, and Richard Socher. Evaluating the factual consistency of abstractive text summarization. *arXiv*, pages arXiv–1910, 2019.

[53] Matthew E. 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.

[54] Mitchell Stern, William Chan, Jamie Kiros, and Jakob Uszkoreit. Insertion transformer: Flexible sequence generation via insertion operations. *arXiv preprint arXiv:1902.03249*, 2019.

[55] Tianxiao Shen, Victor Quach, Regina Barzilay, and Tommi Jaakkola. Blank language models. *arXiv preprint arXiv:2002.03079*, 2020.

[56] Chris Donahue, Mina Lee, and Percy Liang. Enabling language models to fill in the blanks. *arXiv preprint arXiv:2005.05339*, 2020.

[57] Jonas Mueller, David Gifford, and Tommi Jaakkola. Sequence to better sequence: continuous revision of combinatorial structures. In *International Conference on Machine Learning*, pages 2536–2544, 2017.

[58] Angela Fan, David Grangier, and Michael Auli. Controllable abstractive summarization. *arXiv preprint arXiv:1711.05217*, 2017.

[59] Kelvin Guu, Tatsunori B. Hashimoto, Yonatan Oren, and Percy Liang. Generating sentences by editing prototypes. *Transactions of the Association for Computational Linguistics*, 6:437–450, 2018.

[60] Kelvin Guu, Panupong Pasupat, Evan Liu, and Percy Liang. From language to programs: Bridging reinforcement learning and maximum marginal likelihood. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1051–1062, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[61] Yanpeng Li. Learning features from co-occurrences: A theoretical analysis. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2846–2854, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics.

[62] Darsh J Shah, Tal Schuster, and Regina Barzilay. Automatic fact-guided sentence modification. *arXiv preprint arXiv:1909.13838*, 2019.

[63] Kathleen McKeown and Dragomir R Radev. Generating summaries of multiple news articles. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 74–82, 1995.

[64] Regina Barzilay, Daryl McCullough, Owen Rambow, Jonathan DeCristofaro, Tanya Korelsky, and Benoit Lavoie. A new approach to expert system explanations. In *Natural Language Generation*, 1998.

[65] Sam Wiseman, Stuart Shieber, and Alexander Rush. Learning neural templates for text generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3174–3187, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[66] 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 2019. Association for Computational Linguistics.

[67] Wang Wenbo, Gao Yang, Huang Heyan, and Zhou Yuxiang. Concept pointer network for abstractive summarization. *arXiv preprint arXiv:1910.08486*, 2019.

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

[69] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In *Advances in neural information processing systems*, pages 1693–1701, 2015.

[70] 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. *arXiv preprint arXiv:1808.08745*, 2018.

[71] Rik Koncel-Kedziorski, Dhanush Bekal, Yi Luan, Mirella Lapata, and Hannaneh Hajishirzi. Text Generation from Knowledge Graphs with Graph Transformers. 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 2284–2293, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[72] Darsh J Shah, Lili Yu, Tao Lei, and Regina Barzilay. Nutri-bullets: Summarizing health studies by composing segments. *arXiv preprint arXiv:2103.11921*, 2021.

[73] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

[74] Gerald Albaum. The likert scale revisited. *Market Research Society. Journal.*, 39(2):1–21, 1997.

[75] Andrew Hoang, Antoine Bosselut, Asli Celikyilmaz, and Yejin Choi. Efficient adaptation of pretrained transformers for abstractive summarization. *arXiv preprint arXiv:1906.00138*, 2019.

[76] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[77] Jie Yang and Yue Zhang. Ncrf++: An open-source neural sequence labeling toolkit. *arXiv preprint arXiv:1806.05626*, 2018.

[78] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

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

[80] 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, 1998.

[81] Aria Haghighi and Lucy Vanderwende. Exploring content models for multi-document summarization. In *NAACL HLT*, pages 362–370, 2009.

[82] Yue Hu and Xiaojun Wan. Automatic generation of related work sections in scientific papers: an optimization approach. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1624–1633, 2014.

[83] Kavita Ganesan, ChengXiang Zhai, and Jiawei Han. Opinosis: A graph based approach to abstractive summarization of highly redundant opinions. *Coling 2010*, 2010.

[84] Regina Barzilay, Kathleen 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, 1999.

[85] Darsh J. Shah, Lili Yu, Tao Lei, and R. Barzilay. Nutribullets hybrid: Multi-document health summarization. In *NAACL*, 2021.

[86] Peter J. Liu, Mohammad Ahmad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. Generating wikipedia by summarizing long sequences. In *arXiv preprint arXiv:1801.10198*, 2018.

[87] Nikita Moghe, Siddhartha Arora, Suman Banerjee, and Mitesh M. Khapra. Towards exploiting background knowledge for building conversation systems. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2322–2332, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[88] Jason Weston, Emily Dinan, and Alexander Miller. Retrieve and refine: Improved sequence generation models for dialogue. In *Proceedings of the 2018 EMNLP Workshop SCAI: The 2nd International Workshop on Search-Oriented Conversational AI*, pages 87–92, Brussels, Belgium, October 2018. Association for Computational Linguistics.

[89] Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. Wizard of wikipedia: Knowledge-powered conversational agents. *arXiv preprint arXiv:1811.01241*, 2018.

[90] Angela Fan, Claire Gardent, Chloe Braud, and Antoine Bordes. Augmenting transformers with knn-based composite memory for dialogue. *arXiv preprint arXiv:2004.12744*, 2020.

[91] Jiatao Gu, Yong Wang, Kyunghyun Cho, and Victor OK Li. Search engine guided neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[92] Kelvin Guu, Tatsunori B Hashimoto, Yonatan Oren, and Percy Liang. Generating sentences by editing prototypes. *Transactions of the Association for Computational Linguistics*, 6:437–450, 2018.

[93] Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through memorization: Nearest neighbor language models. *arXiv preprint arXiv:1911.00172*, 2019.

[94] Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. Eli5: Long form question answering. *arXiv preprint arXiv:1907.09190*, 2019.

[95] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.

[96] Fabio Petroni, Patrick Lewis, Aleksandra Piktus, Tim Rocktäschel, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. How context affects language models' factual predictions. *arXiv preprint arXiv:2005.04611*, 2020.

[97] William C Mann and Sandra A Thompson. Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8(3):243–281, 1988.

[98] Naman Goyal and Jacob Eisenstein. A joint model of rhetorical discourse structure and summarization. In *Proceedings of the Workshop on Structured Prediction for NLP*, pages 25–34, Austin, TX, November 2016. Association for Computational Linguistics.

[99] Amandla Mabona, Laura Rimell, Stephen Clark, and Andreas Vlachos. Neural generative rhetorical structure parsing. *arXiv preprint arXiv:1909.11049*, 2019.

[100] Florian Kunneman, Sander Wubben, Antal van den Bosch, and Emiel Krahmer. Aspect-based summarization of pros and cons in unstructured product reviews. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2219–2229, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics.

[101] Lea Frermann and Alexandre Klementiev. Inducing document structure for aspect-based summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6263–6273, Florence, Italy, July 2019. Association for Computational Linguistics.

[102] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019.

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

[104] Xinyu Xing, Xiaosheng Fan, and Xiaojun Wan. Automatic generation of citation texts in scholarly papers: A pilot study. In *Proceedings of the 58th Annual*

Meeting of the Association for Computational Linguistics, pages 6181–6190, Online, July 2020. Association for Computational Linguistics.

[105] 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 2019. Association for Computational Linguistics.

[106] 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.

[107] Mary L McHugh. Interrater reliability: the kappa statistic. *Biochemia medica*, 22(3):276–282, 2012.

[108] Jiayuan Huang, Arthur Gretton, Karsten Borgwardt, Bernhard Schölkopf, and Alex J Smola. Correcting sample selection bias by unlabeled data. In *Advances in neural information processing systems*, pages 601–608, 2007.