Adapative Summarization for Low-resource Domains and Algorithmic Fairness

Dissertation

Presented in Partial Fulfillment of the Requirements for the Degree Doctor of Philosophy in the Graduate School of The Ohio State University

By

Moniba Keymanesh

Graduate Program in Computer Science and Engineering

The Ohio State University

2022

Dissertation Committee:

Prof. Srinivasan Parthasarathy, Advisor

Prof. Micha Elsner

Prof. Tanya Berger-Wolf

Abstract

The wealth of data available at a single click often adds to the information overload problem. Summarization is an intuitive way to address this problem by constructing a condensed equivalent of the available data. However, the content of interest and the desired format or length are user-dependent. Most of the existing summarization systems yield generic summaries disconnected from users' preferences and agnostic about the salience of information in the target domain. Moreover, the neural summarization models require a large training corpus which is not available in many domains. Motivated by these limitations, we focus on controllable summarization that allows users to control different aspects of the generated summaries.

(i) To enable users to control the length of summaries, we propose a multi-level summarizer (MLS), a supervised approach to construct abstractive summaries at controllable lengths. Following an extract-then-compress paradigm, we develop the Pointer-Magnifier network– a length-aware, encoder-decoder network that constructs length-constrained summaries by shortening or expanding a prototype summary inferred from the document. The key enabler of this network is an array of semantic kernels with clearly defined human-interpretable syntactic/semantic roles in constructing the summary given a desired length. We discuss this architecture in Chapter 2.

(ii) We acknowledge that many recent advancements in summarization research, including sequence-to-sequence models, cannot be adopted in many domains due to the scarcity of training data for summarization. Legal contracts are considered a low-resource domain for the automatic

text summarization task as the available training data is limited in this domain. On the other hand, unsupervised methods rely on structural features of documents, such as lexical repetition to identify and extract important content. These heuristics showed poor empirical performance on a few low-resource domains.

In chapter 3, we propose a hybrid framework for extractive summarization of privacy policies. We show empirical results of adopting a classifier for identifying the risky data practices in the privacy policies. Given the probability distribution over the risk categories, we apply two content selection mechanisms to account for the summarization budget and minimize the information redundancy. We empirically show that our proposed pipeline outperforms domain-agnostic baselines on the summarization of privacy policies. In addition, we show negative results on pre-training and fine-tuning sequence-to-sequence networks on this domain.

(iii) Summarization can be constrained by the user's query. This requires answers to be found in the extracted summary. However, in many domains, users might not be good at articulating their questions e.g. their questions might have a very different style and language compared to the input document. As a result, summarization models fail to find the answer to user's question effectively. Moreover, existing annotated data for query-guided summarization tasks are limited. Motivated by these issues, in Chapter 4, we discuss using paraphrasing to bring the style and language of the user questions closer to the language of privacy policies. We use familiar techniques such as back-translation and lexical substitution and examine to what extent these previously unexplored techniques in the legal domain are beneficial for the privacy policy question answering task. Following query expansion, we use a content scoring module that uses the existing in-domain data to find relevant information in the policy. Our pipeline can find an answer for 87.7% of the user queries in the privacyQA dataset.

(iv) Expressing natural language summary of structured facts or relations – data-to-text summarization (D2T) – increases the accessibility of structured knowledge repositories. Previous work shows that pre-trained language models (PLMs) perform remarkably well on this task after fine-tuning on a significant amount of task-specific training data. On the other hand, while auto-regressive PLMs can generalize from a few task examples, their efficacy at D2T is largely unexplored. Furthermore, we have an incomplete understanding of the limits of PLMs on D2T. In Chapter 5, we conduct an empirical study of both fine-tuned and auto-regressive PLMs on a multi-domain D2T dataset. We consider their performance as a function of the amount of task-specific data and how the data is incorporated into the models: zero and few-shot learning, and fine-tuning of model weights. In addition, we probe the limits of PLMs by measuring performance on subsets of the evaluation data: novel predicates and abstractive test examples.

We show that the performance of fine-tuned T5 drops significantly on unseen predicates. On the other hand, the performance of few-shot GPT2-XL on unseen predicates can be enhanced even with shots containing unrelated predicates. We also notice that T5 and GPT2-XL both do well at D2T by copying the input. However, they do noticeably worse on examples where significant re-writing is needed. Adding domain knowledge (predicate descriptions) to the prompts can improve the performance of few-shot GPT2-XL on this subset by a significant amount. We also conduct a human evaluation of the generations and find that prompt tuned GPT2-XL generations can be improved by re-ranking generations by overlap with the input entity spans.

We quantitatively evaluate the performance of our proposed controllable summarization models on several domains, including news articles and Wikipedia, and low-resource domains for summarization such as social media discussion forums, privacy policies, and book chapters.

(v) Machine learning models are increasingly used to assist or replace humans in decision-making. In consequential domains such as recidivism prediction, facility inspection, and benefit assignment, individuals need to know the decision-relevant information for the model's prediction. In addition, predictions should be fair both in terms of the outcome and the justification of the outcome. In other words, decision-relevant features should provide sufficient information for the predicted outcome and should be independent of the membership of individuals in protected groups such as race and gender.

In Chapter 6, we show a novel application of text summarization for enhancing fairness in the justification of the text-based neural models. We tie the explanatory power of the model to fairness in the outcome and propose a fairness-aware summarization mechanism (`FairSum`) to detect and counteract the bias in such models. Given a potentially biased natural language explanation for a decision, we use a multi-task neural model and an attribution mechanism based on integrated gradients to extract high-utility and low-bias justifications in form of a summary. The extracted summary is then used for training a model to make decisions for individuals.

Results on the Chicago food inspection dataset and teaching evaluations written by students on ratemyprofessor.com [1] suggest that our method drastically limits the demographic leakage in the input (fairness in justification) while moderately enhancing the fairness in the outcome. Our model is also effective in detecting and counteracting several types of data poisoning attacks that synthesize race-coded reasoning or irrelevant justifications.

In Chapter Future Work, we discuss our ongoing work on automatically creating more accessible presentation forms for the privacy polices and enhancing the fairness and explainability of text-based neural models.

---

[1] https://www.ratemyprofessors.com

To my parents, mentors, and friends who encouraged me to go on every adventure,

especially this one . . .

Acknowledgments

I got to know and work with countless amazing people during my Ph.D. journey. First and foremost, I would like to express my heartfelt appreciation toward Prof. Srinivasan Parthasarathy(Srini) for giving me the freedom to explore my research interest and putting me in charge of my research projects from day one. He is a knowledgeable and visionary leader and always gave me constructive feedback on my ideas. His vision and support kept me focused on improving my work after every paper rejection. I am fortunate that I got to work with him during my time at OSU.

Next, I would like to express my deepest appreciation to Prof. Tanya Berger-Wolf. I have always admired her passion in research and her dedication in training her students and creating a collaborative and friendly lab atmosphere. She was available to give detailed feedback on my research projects even outside of regular hours. She also helped me connect with several colleagues of her in other departments, universities, and in industry. I think she is an exemplary research mentor and I feel extremely lucky to work with her. She is definitely a role model for me and several other women in the CSE graduate program.

Next, I would like thank Prof. Micha Elsner. Micha is one of the smartest and most humble people that I have worked with. He has always been very dedicated to my research progress. He always made time for our weekly meetings in his busy schedule and gave me constructive and insightful feedback on my ideas, code, experiments, scientific papers, and conference presentations.

Our discussions helped me develop some of my current and future work, and improve the quality of this dissertation.

I would also like to thank my research mentors at Bloomberg, Dr. Adrian Benton, and Prof. Mark Dredze, for interesting discussions and valuable feedback. I am incredibly grateful to my collaborators, Dr. Mark Sussman, Ritesh Sarkhel, Prof. Arnab Nandi, Dr. Bethany Boettner, Prof. Christopher Browning, Prof. Catherine Calder, Prof. Ian A. Kash, and Prof. Barbara Kowalcyk, Shubham Singh, Allison Howell, Noah Katz, and Dr. Wenna Xi.

I would next like to convey my thanks to all my peers at the Data Mining Research Laboratory and Computational Ecology, Environment, and Conservation Lab at OSU: Dr. Nikhita Vedula, Dr. Yu Wang, Goonmeet Bajaj, Dr. Mark Sussman, Pranav Maneriker, Dr. Jiankai Sun, Dr. Bortik Bandyopadhyay, Dr. Jiongqian Liang, Meghana Moorthy Bhat, Saket Gurukar, Saumya Sahai, Vedang Patel, Yuntian He, Sean Current, Namrata Barenji, Tessa Cotron, Sebrina Tesfaye Zeleke, Reshma Ramesh Babu, and Craig Fouts. Their constructive suggestions improved the quality of my ideas, research papers, talks and presentations. I enjoyed knowing these knowledgeable and hardworking researchers and hope to have them as friends for years to come.

Next, I want to express my gratitude to my friends Amineh, Saman, Mohadeseh, Shashank, Sherif, Haseeb, Mamzi, Goonmeet, Kawthar, Mounica, Jeniya, and Nathaniel. You helped me keep my sanity during this journey and were no less of a family to me in years away from home.

I owe my deepest and most sincere gratitude to my Maman and Baba for their unconditional love and support. Their courage, passion, grit, kindness, and love for science always inspired me. Finally, I would like to thank Melika, Mona, Mohammadjavad, and Nika. I hope that next time I get to celebrate my happy days with you around me.

The work done during my PhD has been supported the National Science Foundation(NSF) under grant No IIS-1939743, National Institute of Health (NIH) under Grant No NIH-1R01HD088545-01A1, and a Fellowship from the Ohio State University. Computational resources have been provided by the Ohio Supercomputer Center, the Department of Computer Science and Engineering at OSU, and Bloomberg AI. [2]

[2] Any opinions, findings, and conclusions in this dissertation material are those of the the author and her collaborators, and may not reflect the views of the respective funding agency.

Vita

Sep 2012–Sep 2016	B.S. Department of Computer Engineering and Information Technology, Amirkabir University of Technology (Tehran Poly-Technic), Tehran, Iran
June 2015–Sep 2015	Mobile Application Developer, Zoodfood, Tehran, Iran
June 2016–Sep 2016	Data Science Intern, Farabord, Tehran, Iran
Jan 2017–Aug 2017	Software Engineer, Soha Tosan, Tehran, Iran
Aug 2017–Present	Ph.D. Student, Department of Computer Science and Engineering, The Ohio State University, OH, USA
Aug 2017–Aug 2018	University Graduate Fellow, The Ohio State University, OH, USA
Aug 2018–Aug 2022	Graduate Research Associate, Department of Computer Science and Engineering, The Ohio State University, OH, USA
May 2019–Aug 2019	Natural Language Processing Research Intern, Hunter Capital, New Albany, OH, USA
June 2020–Aug 2020	Data Science Intern, Kohls Technology, San Jose, CA, USA
June 2021–Aug 2021	Data Science Intern, Kohls Technology, San Jose, CA, USA

Aug 2021 ....M.S., Department of Computer Science and Engineering, The Ohio State University, OH, USA

Aug 2021–Dec 2021 ....AI Research Intern, Bloomberg L.P. New York City, NY, USA

Publications

Research Publications

Moniba Keymanesh, Saket Gurukar, Bethany Boettner, Christopher Browning, Catherine Calder, Srinivasan Parthasarathy "Twitter watch: Leveraging social media to monitor and predict collective-efficacy of neighborhoods" *Proceedings of the 11th Conference on Complex Networks (CompleNet) 2020*, 197–211, Dec. 2020.

Ritesh Sarkhel*, **Moniba Keymanesh***, Arnab Nandi, Srinivasan Parthasarathy "Interpretable Multi-headed Attention for Abstractive Summarization at Controllable Lengths" (*equal contribution) *Proceedings of the 28th International Conference on Computational Linguistics (COLING)*, 2020.

Moniba Keymanesh, Micha Elsner, Srinivasan Parthasarathy "Toward Domain-Guided Controllable Summarization of Privacy Policies" *Proceedings of the Natural Legal Language Processing (NLLP) Workshop co-located with KDD*, 2020.

Moniba Keymanesh, Micha Elsner, Srinivasan Parthasarathy "Privacy Policy Question Answering Assistant: A Query-Guided Extractive Summarization Approach" *Proceedings of the Natural Legal Language Processing (NLLP) Workshop co-located with EMNLP*, 2021.

Moniba Keymanesh, Tanya Berger-Wolf, Micha Elsner, Srinivasan Parthasarathy "Fairness-aware summarization for justified decision-making". *arXiv preprint arXiv:2107.06243*, 2021.

Saket Gurukar, Priyesh Vijayan, Balaraman Ravindran, Aakash Srinivasan, Goonmeet Bajaj, Chen Cai, **Moniba Keymanesh**, Saravana Kumar, Pranav Maneriker, Anasua Mitra, Vedang Patel "Benchmarking and Analyzing Unsupervised Network Representation Learning and the Illusion of Progress". *Transactions on Machine Learning Research (TMLR)*, 2022

Tanya Berger-Wolf, Allison Howell, Chris Kanich, Ian A. Kash, **Moniba Keymanesh** , Barbara Kowalcyk, Gina Nicholson Kramer, Andrew Perrault, Shubham Singh "Open Problems in (Un)fairness of the Retail Food Safety Inspection Process". *ICML 2022 Workshop on Responsible Decision Making in Dynamic Environments,*, 2022.

Moniba Keymanesh, Adrian Benton, Mark Dredze "What Makes Data-to-Text Generation Hard for Pretrained Language Models?" *arXiv preprint arXiv:2205.11505*, 2022.

Fields of Study

Major Field: Computer Science and Engineering

Studies in:

Data Mining	Prof. Srinivasan Parthasarathy
Computational Linguistics	Prof. Micha Elsner
Statistics	Prof. Yuan Zhang

Table of Contents

Page

Abstract ii

Dedication vi

Acknowledgments vii

Vita x

List of Tables xvii

List of Figures xxi

1. Introduction 1

    1.1 Dissertation Statement 2
    1.2 Contribution I: Controlling the length of summaries 3
    1.3 Contribution II: Domain-guided summarization of privacy policies 4
    1.4 Contribution III: Query-Guided Summarization of Privacy Policies 5
    1.5 Contribution IV: Data-to-Text Summarization (D2T) 6
    1.6 Contribution V: Summarization for fairly-justified decision making 7
    1.7 Conclusion and Future Work 8
    1.8 Organization 9

2. Abstractive Summarization at Controllable Lengths 10

2.1 Introduction and Related Work 10

2.2 Proposed Methodology 13

2.2.1 Generating the Prototype Summary 13

2.2.2 Constructing the Length-Constrained Summary 14

2.2.3 Training the Networks 17

2.3 Experiments 18

2.3.1 Results and Discussion 20

2.4 Conclusion 22

3. Domain-guided summarization of privacy policies 23

3.1 Introduction and Related Work 23

3.2 Human-Centered Analysis of Privacy Policy Comprehension 25

3.3 Extractive Summarization of Privacy Policies 29

3.3.1 Risk Prediction 29

3.3.2 Content Selection and Redundancy Reduction 30

3.4 Dataset Extraction 31

3.5 Experiments 31

3.5.1 Hyperparameters and Training Details 32

3.5.2 Evaluation Metrics 32

3.5.3 Summarization Baselines 33

3.6 Results 33

3.6.1 Classification Results 33

3.6.2 Summarization Results 34

3.7 Conclusion 36

4. Automated Privacy Policy Question Answering Assistant 39

4.1 Introduction and Related Work 39
4.2 Proposed Pipeline 40
4.2.1 Query Expansion 41
4.2.2 Content Scoring 43
4.2.3 Answer Ranking and Selection 44
4.3 Datasets 45
4.4 Hyperparameters and Training Details 46
4.5 Experiments and Results 47
4.5.1 Query Expansion Results: 48
4.5.2 Question-Answering Results: 49
4.6 Conclusion 51

5. Low-Resource Data-to-Text Generation with Pretrained Language Models 53

5.1 Introduction 53
5.2 Background and Related Work 55
5.3 Model Adaptation 55
5.3.1 0-shot Setting 56
5.3.2 Few-shot Setting 57
5.3.3 Prompt Tuning 57
5.3.4 Domain Knowledge 58
5.3.5 Fine-tuned PLM 58
5.4 Dataset 58
5.5 Experimental Setup 59
5.6 Experiments and Results 60
5.7 Sample Model Output 64
5.8 Conclusion 65

6. Fairness aware summarization for justified decision making 71

6.1 Introduction 71
6.2 Related Work 73
6.3 Problem Formulation 74
6.4 Proposed Methodology 74
6.4.1 Model Architecture 76
6.4.2 Attribution 77
6.5 Experiments and Results 79
6.5.1 Datasets 79
6.5.2 Hyper-parameters and Training Details 81
6.5.3 Measuring Fairness of Justification and Outcome 82
6.5.4 Results and Discussion 84
6.5.5 Unfairness Attacks 86
6.5.6 Pilot user study 91
6.6 Conclusion 92

7. Conclusions, Limitations and Future Work 94

7.1 Our Key Contributions 94
7.2 Limitations and Future Work 96
7.2.1 Controllable and Low-Resource Summarization 97
7.2.2 Legal Text Summarization and Information Retrieval 97
7.2.3 Data-to-text Generation 98
7.2.4 Fairness of Natural Language Processing Models 98

List of Tables

Table **Page**

2.1 *The min, max, median, and average number of sentences in datasets D1 and D2* . . 18

2.2 *ROUGE and METEOR scores of the budgeted summaries constructed by* `MLS` *(highlighted column) and the baseline methods for the MSR-Narrative (D1) and Thinking Machines (D2) dataset* 19

2.3 *Coherence and completeness of the budgeted summaries constructed by* `MLS` *(highlighted column) and the baseline methods for MSR-Narrative (D1) and Thinking Machines (D2) dataset* 20

2.4 *Mean accuracy and completion time using* `MLS`, *A2, A3, No-content (NC) and Full-content (FC) settings* 21

3.1 The list of companies in our user study and their corresponding grade level (A-E). 25

3.2 Results of user's comprehension, preference, agreement level and task completion time for different presentation forms. 27

3.3 The min, max, median, and average number of sentences in 151 privacy contracts and their summaries. 31

3.4 Precision(P), Recall(R), Macro-F1, and Micro-F1 of the CNN classifier with two different content selection mechanisms risk-focused(RF) and coverage-focused(CF) at two different compression ratios $\frac{1}{16}$ and $\frac{1}{64}$. 32

3.5 ROUGE-1, ROUGE-2, ROUGE-l, and METEOR score of our model (highlighted in light gray) in comparison to the baselines in compression ratios $\frac{1}{64}$ and $\frac{1}{16}$. RF refers to the risk-focused content selection while CF refers to the coverage-focused content selection. The quote text of the risky sections was used to build the reference summaries. 34

3.6 Performance of our model (highlighted in light gray) in comparison to the baselines in compression ratios $\frac{1}{64}$ and $\frac{1}{16}$. RF refers to the risk-focused content selection while CF refers to the coverage-focused content selection. The plain English summaries of risky sections was used to build the reference summaries. 35

4.1 Lexical substitution rules used to bring the language of user queries closer to the legal language of privacy policy. 42

4.2 Statistics of PrivacyQA Dataset; where # denotes number of questions, policies, and out-of-scope questions. Out-of-score questions refer to questions for which no segment in the policy is annotated as relevant. We also report the average number of annotated passages/segments and the average number of relevant segments for each question. 46

4.3 Statistics of PolicyQA Dataset; where # denotes number of questions, policies, and Q&A pairs. We also report the average number of words in passages/segments, questions, and answer spans. 46

4.4 The average number of paraphrases generated by each method. The percentage of generated answerable paraphrases for non-answerable queries and the percentage of relevant segments that were answerable using at least one of the paraphrases by each method. 48

4.5 The performance of different variations of our model and the baselines in the policy question answering task. R represents the legal-Bert model fine-tuned on PrivacyQA for relevance score prediction. I shows the legal-Bert model fine-tuned on PolicyQA for informativeness prediction. Q represents the query expansion module. F@K represent percentage of queries for which at least one relevant segment was found within the top k ranked items. P@K represent precision at K, and RP presents R-precision. MRR is the mean reciprocal rank. Bold font indicates the best result for each performance metric. 50

4.6 Examples of user queries answered both our pipeline (R+A+Q) and the closest model to Ravichander et al. [195] (R), only our model or none of the models. 51

5.1 Descriptive statistics of the DART version 1.1.1 59

5.2 Model results on test set of the DART dataset. ↑: Higher is better. ↓: Lower is better. 61

5.3 Performance on the DART test set, partitioned by whether predicates are SEEN, UNSEEN, and overall. ↑: Higher is better. 61

5.4 Model results on EASY and HARD partitions of the DART test set. ↑: Higher is better. ↓: Lower is better. 63

5.5 Results of the qualitative evaluation. ↓: Lower is better. ↑: Higher is better. Inter-annotator agreement is measured by Kendall's τ rank correlation coefficient. . . . 64

6.1 The percentage of inspections for each ethnic group that received a pass, conditional pass, or a fail outcome. 80

6.2 The percentage of instructors of each gender group in each rating class for dataset D2. 80

6.3 The percentage of instructors of each gender group in each rating class for dataset D3. 81

6.4 The percentage of instructors of each gender group in each rating class for dataset D4. 81

6.5 Results on datasets D1. ”↑”: higher is better. ”↓”: lower is better. 84

6.6 Fairness metrics for datasets D1. ”↓”: lower is better. 85

6.7 Results on RMP Datasets (D2-D4). ”↓”: lower is better. 85

6.8 Summary of the 5 data poisoning attacks 90

6.9 Experimental results with 5 types of poisoning attacks. Attack success is measured by measuring false fail rate as well as demographic leakage before (indicated with Full) and after applying FairSum. ↓: lower is better. 90

6.10 Accuracy of users in rating prediction task and inter-annotator agreement (Fleiss Kappa) given different subsets of the teaching reviews. 91

List of Figures

Figure **Page**

1.1 An overview of our work on controllable text summarization. We aim to re-define and constrain the automated text summarization task based on user queries and preference in length, format, and level of neutrality (e.g. use of gender or race coded language) 3

2.1 MLS *expands the highlighted sentences in the prototype summary to the boldfaced tokens in the input text to construct a summary budgeted at half-length of the input text* 11

2.2 *An overview of* MLS *architecture. The PG-Network (left) constructs a prototype summary* s_p *from the input document. The Pointer-Magnifier network (right) constructs the length-constrained summary from* s_p *using interpretable sentence-level attention* 13

2.3 *The Encoder layer consists of 3 parallelly stacked encoder-blocks* 15

2.4 *Abstractiveness of* MLS *generated summaries* 19

2.5 *ROUGE-1 score of* MLS *and the ablative baselines MLS+ and MLS* on datasets D1 and D2* 21

3.1 Annotation for Grade and Points for Facebook on TOSDR 26

3.2 The summaries extracted by our model (CNN + RF and CNN + CF) from the privacy policy and cookie policy of Brainly at compression ratio of $\frac{1}{64}$. 37

3.3 The summaries extracted by the baselines from the privacy policy and cookie policy of Brainly at compression ratio of $\frac{1}{64}$. 38

4.1 Overview of the proposed pipeline 41

5.1 A customized 0-shot prompt for GPT 56

5.2 A customized 3-shot prompt for GPT 57

5.3 Examples from the EASY and HARD partition 66

5.4 Impact of fine-tuning data size on performance of T5. Numbers reflect average performance over 5 different data samples, with standard error of the mean indicated by bars. 67

5.5 Examples of model outputs 68

5.6 Examples of model outputs 69

5.7 Examples of model outputs 70

6.1 A fairly-justified decision should have a fair outcome and be based on fair justifications. 72

6.2 A graphical model of the proposed approach. P represents the protected attribute. X indicates the input explanations while X' indicates the farily-justified summary of X which is used to train the final model to predict outcome $\hat{Y}$. 75

6.3 An overview of the architecture: Decision learner and membership identifier are trained using decision Y and protected attribute P as supervision respectively. The attributions of each module is normalized and subtracted to obtain the inclusion scores. 78

6.4 Applying `FairSum` on teaching evaluations for a female professor (anonymized and paraphrased for privacy considerations). The pronouns and names have been removed before model training and attribution. Sentences with a positive attribution score (purple) are preserved in the summary x' while the sentences with a negative attribution score (orange) are excluded. 86

6.5 Applying `FairSum` on teaching evaluations for a male professor (anonymized and paraphrased for privacy considerations). Sentences with a positive attribution score (purple) are preserved in the summary x' while the sentences with a negative attribution score (orange) are excluded. 86

6.6 Impact of α on utility and fairness on datasets D1 (a and b) and D2 (c). 87

6.7 Impact of α on summary length on datasets D1-D4. 87

6.8 Distribution of the predicted gender in (a) positive reviews and (b) negative reviews. 92

Chapter 1: Introduction

The wealth of information available online often adds to the information overload problem [234, 221]. Assimilating new information can be cumbersome for users due to the cognitive fatigue caused by the excessive amount of information available. We can address this problem by presenting information in such a way that minimizes cognitive fatigue [57]. Automatic summarization [171] is an intuitive way to address information overload for textual data. Automatic text summarization refers to transforming a potentially long document into a condensed equivalent while preserving the key information. Broadly, there are two main lines of summarization systems: extractive and abstractive.

The extractive paradigm [43, 169] creates a summary by identifying and subsequently concatenating the most important sentences in the document. Neural approaches to extractive text summarization often formulate this task as a sentence classification or ranking task. In this setting, a neural encoder constructs text representations, and a classifier or a ranking model identifies the sentences to be included in the summary. For example, Nallapati et al. [167] adopt an encoder-based recurrent neural network for extractive text summarization while Narayan et al. [169] use a reinforcement learning-based system and globally optimize the ROUGE metric for sentence ranking. Zhou et al. [251] propose an end-to-end neural architecture for jointly learning to score and select sentences. More recently, Liu [142] and Zhang et al. [248] employ hierarchical document encoders based on the transformer architecture for sentence selection.

On the other hand, the abstractive paradigm [209, 204, 181] aims to create an abstract representation of the input text using various text rewriting operations such as paraphrasing, deletion, and reordering. Neural approaches to abstractive text summarization often formulate this task as a sequence-to-sequence problem, where an encoder maps a sequence of tokens in the input document to a sequence of continuous representations. Next, a decoder reads the encoded input sequence and generates the target summary token-by-token in an auto-regressive manner. Rush et al. [204] and Nallapati et al. [165] adopt the encoder-decoder architecture for text summarization task. See et al. [209] enhance this architecture by utilizing a pointer-generator network. In this architecture, the pointing mechanism enhances the accurate reproduction of information by copying words from the input document. In addition, the coverage mechanism discourages repetition by keeping track of what has been summarized. Celikyilmaz et al. [32] use several deep communicating agents where each agent is responsible for encoding a subsection of the input text. These encoders are connected to a single decoder. The model is trained end-to-end using reinforcement learning.

Paulus et al. [181] use a deep reinforced model for abstractive summarization to handle the coverage problem. They propose an intra-attention mechanism where the decoder attends over previously generated words. Gehrmann et al. [81] follow a bottom-up approach to improve content selection in abstractive summarization. They use a content selector as a bottom-up attention step to constrain the model to determine which phrases in the source document should be part of the summary. A copy mechanism is then applied only to pre-selected phrases during decoding. More recently, Vaswani et al. [226] and Dong et al. [59] adopt the transformer architecture for abstractive text summarization. Liu and Lapata [143] adopts an encoder-decoder architecture, combining the same pretrained BERT [56] encoder with a randomly-initialized Transformer decoder. Zhang et al. [246] propose a new self-supervised pre-training objective for abstractive summarization.

While significant progress has been made in both extractive and abstractive summarization, most existing summarization systems yield a single generic summary for an input document. Most of these models are disconnected from users' preferences and are agnostic about the salience of information in the target domain. Moreover, training neural summarization models is a resource-intensive task, while low-resource settings are common in real-world applications. This is because curating domain-specific summarization datasets for long documents and on a large scale is not feasible or cost-efficient in many domains [10].

Motivated by these limitations, we focus on controllable summarization for long documents to allow users to control different aspects of the generated summaries. We focus on constraining the summarization model by a length budget, domain information, user queries, format, or fairness objectives. We utilize advances in domain adaptation, language model pre-training, few-shot learning, data augmentation, and data synthesis to build supervised and semi-supervised models for controllable summarization subject to these control aspects. We present a formal dissertation statement in Section 1.1.

1.1 Dissertation Statement

In this dissertation, We present a comprehensive study of controllable summarization for low-resource domains. We propose supervised and semi-supervised models for controllable summarization subject to user preferences. We focus on controlling length of the summaries (§1.2), including domain knowledge in summaries (§1.3), and generating summaries in response to users' query (§1.4).

In (§1.5), we look at data-to-text summarization task. In this task, the goal is to generate a natural language description for structured records or facts. We systematically analyze the performance of two pretrained language models on this task based on the choice of adaptation mechanism: fine-tuning, prompt tuning, and few-shot learning.

In (§1.6), we look at a novel application of text summarization in enhancing fairness. In this application, the summarization is constrained by fairness objectives and is used as a preprocessing step to remove potential biases from the justifications. This preprocessing step will assist in learning predictive models that are not biased toward specific population subgroups. We show the overview of our work on controllable summarization in Figure 1.1. In this dissertation, we seek to answer the following questions:

- *Can we generate abstractive summaries at controllable lengths for domains where limited annotated data is available and the desired length is not known beforehand?* (§1.2)

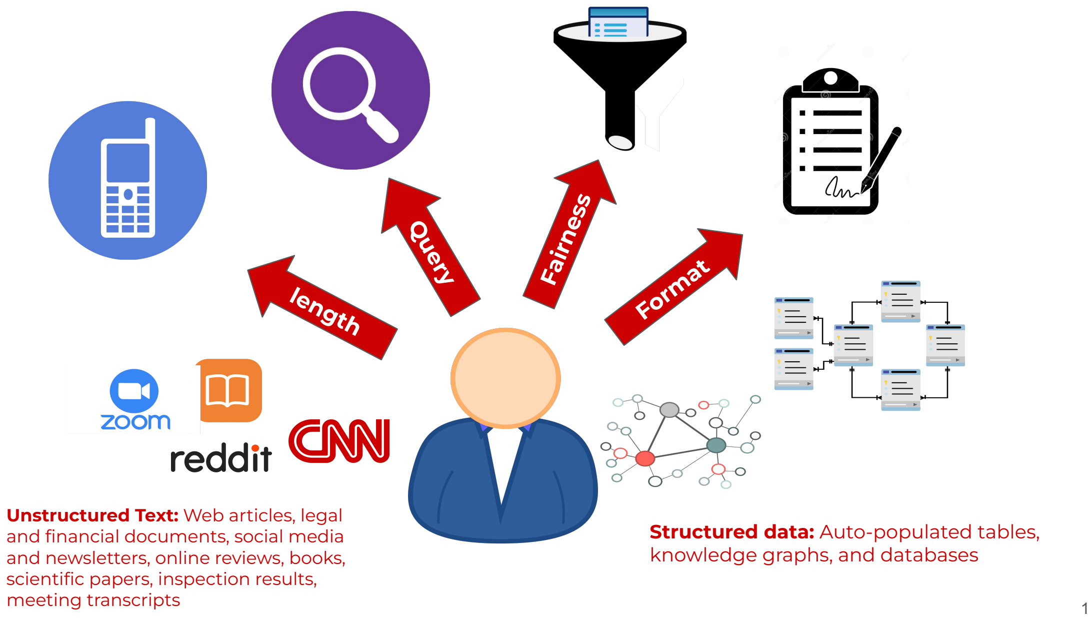


Figure 1.1: An overview of our work on controllable text summarization. We aim to re-define and constrain the automated text summarization task based on user queries and preference in length, format, and level of neutrality (e.g. use of gender or race coded language)

- *Given domain-specific knowledge about the salience of information, how can we learn to summarize with limited annotated data?*(§1.3)
- *How can we guide the summarization process with user's query in settings where limited training data is available and user queries are articulated in a very different style and wording than the target domain?*(§1.4)
- *How can we effectively adapt pre-trained language models for D2T task in domains where not enough data is available for fine-tuning? How do the adaptation mechanism and level of supervision at train time affect their performance?* (§1.5)
- *Can we incorporate fairness objectives in the summarization process to remove potential biases from the training data of text-based neural models without significantly compromising the model's utility?* (§1.6)

In the following few paragraphs, we discuss our main contribution and how we hope to address these questions.

1.2 Contribution I: Controlling the length of summaries

Being able to constrain the length of a summary while preserving its desirable properties has many real-world applications. One such application is content optimization for variable screen sizes.

Online content creators such as news portals, blogs, and advertisement agencies with audiences on multiple platforms customize their content based on the display area for the best experience.

However, there has not been much work on summarization at controllable lengths until recently. High variance in screen sizes often requires extensive human supervision to perform these modifications. As most sequence-to-sequence networks do not enforce the length of a summary [204, 165], for scenarios mentioned above, one may need to employ an ensemble of networks to cover all possible sizes. There are two major challenges in following this approach for real-world applications. First, training sequence-to-sequence networks is a resource-intensive task [218]. To train a network for generating summaries budgeted at length b, we need a parallel corpus of text documents and their gold-standard summaries at length b. Constructing a large enough corpus with summaries budgeted at $b, \forall b$ may not be possible or cost-efficient for many domains. Second, the range of possible length-budgets $\mathcal{R}(b)$ may not always be known beforehand. In many scenarios, it can be known as late as during run-time.

To address these challenges, in Chapter 2, we propose Multi-level Summarizer (`MLS`) [207], a supervised method to construct abstractive summaries of a text document at controllable lengths. The key enabler of our method is an interpretable multi-headed attention mechanism that computes attention distribution over an input document using an array of timestep independent semantic kernels. Each kernel optimizes a human-interpretable syntactic or semantic property. Exhaustive experiments on two low-resource datasets in the English language show that `MLS` outperforms strong baselines by up to 14.70% in the METEOR score. Human evaluation of the summaries also suggests that they capture the key concepts of the document at various length budgets.

1.3 Contribution II: Domain-guided summarization of privacy policies

In recent years, significant progress has been made in the abstractive summarization of text documents. Among existing works, sequence-to-sequence networks with attention [80, 138] have been one of the clear front-runners. It is worth observing that most neural summarization models have been trained and tested on the news domain where large-scale news datasets [165, 94, 125] exist while leaving out several important but low-resource domains [149, 179, 207] where the number of available training documents is limited. One important and low-resource domain for summarization is privacy policies [153].

Privacy policy and terms of service are unilateral contracts by which companies must inform users about their data collection, processing, and sharing practices. Users are required to agree to abide by the terms before they can use any service. However, many users do not read or understand this contracts [51]. Thus, they often end up consenting to terms that may not be aligned with legislation, such as the General Data Protection Regulation (GDPR)[3] [173]. This behavior is often because these contracts are too long and challenging to comprehend [156].

Summarization is an intuitive way to assist users with conscious agreement by generating a condensed equivalent of the content. However, existing summarization techniques perform poorly on contracts. Unsupervised methods [160, 85] rely on structural features of documents, such as lexical repetition, to identify and extract important content. These heuristics work poorly on the

[3]https://eugdpr.org/

legal language used in contracts [154]. Supervised methods [209, 81, 180] can learn to cope with the features of a particular domain. However, as stated earlier, training these complex neural summarization models with thousand of parameters requires a large corpus of documents and their summaries which is not currently available in this domain.

In Chapter 3, we propose a hybrid approach for extractive summarization of privacy contracts [113]. Using existing annotated resources and synthetic data, we train a classifier to predict which pieces of content are most relevant to users [51]. In particular, we identify parts of the contract that place users at risk by imposing unsafe data practices, such as selling email addresses to third parties or allowing the company to appropriate user-generated content. Next, we use this risk classifier for content selection within an extractive summarization pipeline. The classifier is substantially less expensive to train than learning to summarize directly but enables our approach to outperform a selection of domain-agnostic unsupervised summarization methods. Our model achieves the best ROUGE and METEOR results compared to domain-agnostic baselines with 49.8% improvement in ROUGE-1 and 65.6% improvement in METEOR compared to the best performing domain-agnostic baseline.

1.4 Contribution III: Query-Guided Summarization of Privacy Policies

As part of the overall goal of assisting users with understanding the content of privacy policies and conscious agreement using a better presentation, our previous work explored incorporating the risky data practices in the privacy policies in a summary [113]. However, users often care about a subset of these issues or have a personal view of what is considered risky. Thus, instead of presenting an overview or summary of privacy policies, an alternative approach is to allow them to ask questions about the issues they care about and show an answer extracted from the content of the policies [194]. This facilitates a more personal approach to privacy and enables users to review only the sections of the policy that they are most concerned about.

In this work, we take a step toward building an automotive privacy policy question-answering assistant. We propose constraining the output summary by the information need of users given in form of a question. This task is related to guided and controllable text summarization [125, 52, 115, 70, 207] as well as reading comprehension [90]. However, a few application-imposed constraints make this task more challenging than the traditional evaluation setup of reading comprehension systems.

First, users tend to pose questions to the privacy policy question-answering system that are not-relevant, out-of-scope (*‘how many data breaches did you have in the past?’*), subjective (e.g. *‘how do I know this app is legit?’*), or too specific to answer using the privacy policy (e.g. *‘does it have access to financial apps I use?’*) [194]. Moreover, even answerable user questions can be ill-phrased or have a very different style and language in comparison to the legal language used in privacy policies [195], making it difficult for the automated assistant to identify the user’s intent and find the relevant information in the document. This issue of domain shift is exacerbated due to the difficulty of annotating data for this domain. Because the existing datasets for this task are fairly small [4], the problems cannot be solved by simply training a supervised model.

In Chapter 4, we take a step toward building an automated privacy policy question-answering assistant. We focus on addressing the domain-mismatch problem and aim to bring the style, language, and specificity of the user's question closer to the language of privacy policies. To do so, we use familiar techniques such as lexical substitution and back-translation that are not previously explored in the legal domain. Next, we compute a relevance and informativeness score for each policy segment using a transformer-based language representation model fine-tuned on in-domain data. Finally, we return the top relevant segments to the user. Using existing in-domain data and techniques such as back-translation and lexical substitution, we can find an answer for 87.7% of the user queries in the PrivacyQA dataset [195].

1.5 Contribution IV: Data-to-Text Summarization (D2T)

Structured data repositories, or knowledge bases, contain a wealth of information organized to facilitate automated access and analysis. Automated data-to-text (D2T) generation systems can transform and organize this knowledge into natural language text snippets that enable broader access [79]. These systems take as input a set of relations, where each relation is a (subject, predicate, object) triple. Applications of this technology include story or dialogue generation [164], open-domain question-answering [146, 71], and text summarization [232]. Domains span journalism [128], weather [193, 158], finance, sports [186, 39, 225], and summarizing patient medical histories [188].

Historically, data-to-text systems included pipeline approaches with customized models [78]. In recent years, pretrained Transformer-based language models (PLM) [56, 144, 189] have come to dominate this task, just as they have other NLP tasks. This approach requires a PLM to be fine-tuned on a task-specific in-domain dataset [97, 210, 112]. The promising results achieved by fine-tuning PLMs belie the reality most domains and relations that one could express fail to appear in current existing datasets for this task. Furthermore, the extensive development effort behind dataset creation, underscores the challenge of creating an in-domain dataset for each task of interest.

Several methods have emerged within PLM research to address domain or task adaptation. For example, auto-regressive models, like GPT, have demonstrated improved performance on a wide range of tasks via few-shot learning from a handful of examples [42]. Other strategies, such as prompt tuning [129], can adapt PLMs to specific downstream tasks by updating only a small subset of model parameters.

While great progress has been made in utilizing PLMs for D2T summarization, the path forward is unclear, as we have an incomplete understanding of which examples they fall short on and the quantity of training resources they need to achieve acceptable performance. More specifically, it is not clear which classes of D2T examples are challenging for these models. In addition, we do not fully understand what classes of errors PLMs are prone to and how the adaptation mechanism (e.g., k-shot learning, fine-tuning) affects the prevalence of these errors.

In Chapter 5, we systematically analyze the performance of two PLMs – T5 and GPT2-XL – for D2T generation by examining performance based on the choice of adaptation mechanism: fine-tuning, prompt tuning, and few-shot learning. We focus on their performance on two classes of challenging examples: examples with novel (*unseen*) relations (*predicates*) and instances where the source and target sequences are lexically very different. We show that while fine-tuning on more data leads to better performance when no training data is available, GPT2-XL (0-shot) outperforms T5. With a small number of training examples, few-shot GPT2-XL is a more appropriate solution for D2T. We also show that the performance of few-shot GPT2-XL on unseen predicates can be enhanced even with shots containing unrelated predicates. We also observe that adding domain knowledge (predicate descriptions) to the prompts can improve the performance of few-shot GPT2-XL on the abstractive D2T examples in Dart by a significant amount.

1.6 Contribution V: Summarization for fairly-justified decision making

Widespread use of AI systems in sensitive environments and for making important and life-changing decisions has important implications for citizens in scenarios such as loan applications, automated hiring, recidivism score, credit rating, etc [157]. The data used for training the models can reflect biases that exist in our society. The models can perpetuate or even reinforce these biases without careful design and engineering.

While training models on historical decisions with unfair outcomes is detrimental, using historical training data with unfair justifications is equally harmful. For example, training a text-based neural model on unfair justifications can cause the model to associate a gender or race-coded phrase in the input to a certain outcome. This phenomena is an example of disparate impact [13, 241]. On the other hand, individuals from two or more protected groups may be treated differently (received different outcomes). But the differences can be justified and explained using multiple fair arguments and, therefore, not considered illegal [157].

Methods that do not consider the explainability aspect of discrimination will result in reverse discrimination [106]. This highlights the need to distinguish between the fairness of the outcome and fairness in the justification of the outcome. A fairly-justified decision should both have a fair outcome and be fairly justified. In other words, the justification should include enough information to explain the outcome [31] and should not be based on information about membership in protected groups.

In Chapter 6, we propose a novel application for text summarization to enhance the fairness in the justification of text-based decision-making models. We propose a text pre-processing approach called `FairSum` based on summarization that extracts the decision-relevant justifications while removing the potentially unfair ones. To measure bias, we use metrics such as demographic parity [27], equalized odds [86], and calibration [121], and by measuring the adversary's ability to identify membership in protected groups given the textual explanations. To counteract the bias, `FairSum` obfuscates the arguments that are not useful for decision making or are only useful when they correlate with the protected attribute. Finally, the extracted fairly-justified summaries are used to train a final model.

`FairSum` ensures learning a model that is both transparent and agnostic about gender-coded or race-coded arguments. In addition, our proposed approach is independent of modeling and can be integrated into the machine learning pipeline with other in-processing and post-processing fairness enhancement mechanisms.

We apply `FairSum` on input justifications of two real-world datasets–Chicago food inspections and teaching evaluations written by students in an online forum. We show that this pre-processing step does not hurt the model's utility for decision-making but significantly limits the leakage of information about protected attributes of individuals.

1.7 Conclusion and Future Work

The long-term goal of this dissertation is to advance automated text-summarization research and make it more accessible in low-resource settings. Throughout this thesis, we attempted to address challenges introduced by the lack of training resources by developing methods and pipelines that effectively utilize limited or no human annotation during training and development.

We also attempt to incorporate the existing domain knowledge in developing resource-efficient methods. Domain knowledge is independent of the task-specific labeled training data. It refers to an auxiliary source of information that can be used to conduct the summarization task in the domain of interest. For example, in Chapter 3 we use information about the potential risk policy segments pose on users to create summaries for privacy policies. In Chapter 5, we include the dictionary definition of relations in creating prompts to enhance the data-to-text generation model's ability to generate descriptions for unseen and abstractive relations.

Another important objective of this dissertation is to redefine and constrain the automated text summarization task based on users' preferences in length, format, and focus of the extracted summary. The challenge posed by lack of task-specific labeled data and the need for control aspects in summarization are crucial in many real-world applications in fields such as law, finance, and medicine.

Constraining the automatic summarization task by a control aspect such as user query or fairness objectives requires going beyond traditional evaluation metrics to measure the quality of the output summaries. Throughout this dissertation, in addition to standard evaluation metrics such as ROUGE [133] and METEOR [124] we evaluated summaries in terms of other qualities depending on the control aspect of interest.

We evaluated summaries in terms of faithfulness to the input document. We used measures such as sentiment, topic, keyword-use and coherence divergence(in Chapter 2). In Chapter 3, in addition to evaluating summaries based on overlap with human-written summaries, we evaluated them based on accuracy in identifying risky policy sections in the output summaries. In Chapter 4 we used IR-based evaluation metrics [152] to measure usefulness of output summaries in answering users' questions. To measure bias in the output summaries, in Chapter 6, we measured demographic leakage and fairness metrics such as equality of odds [86] to understand how information summarization impacts the fairness of automated decision-making.

In addition, one of our objectives is to highlight the importance of the qualitative evaluation of generated summaries. In Chapter 2, we measured information coverage in length-constraint summaries by measuring users' performance in a question-answering task given the summaries. In Chapter 5, we evaluate the generation quality by conducting a human study, asking users to evaluate summaries in terms of hallucination, missing information, and fluency of the generated summaries. In Chapter 6, we evaluate the neutrality of extracted summaries in revealing the protected attribute by asking human subjects to guess the gender of instructors given their summarized teaching reviews.

1.8 Organization

The rest of the dissertation is organized as follows. We discuss a supervised method to construct abstractive summaries of a text document at controllable lengths for low-resource domains in Chapter 2. Chapter 3 describes our proposed domain-guided extractive summarization pipeline for privacy policies. In Chapter 4, we discuss our proposed pipeline for query expansion and question answering of privacy policies. We then look at our work on controlling the format of the output summary in low-resource domains; we systematically analyze the performance of pre-trained language models on the data-to-text generation task in Chapter 5. In Chapter 6, we introduce `FairSum`; an extractive summarization model with fairness objectives and present our empirical analysis of using this method to remove biases from justifications. We conclude this dissertation and present some interesting and important directions of future research in Chapter 7.

Chapter 2: Abstractive Summarization at Controllable Lengths

Recent advances in abstractive summarization based on the encoder-decoder architecture only generate a single summary for a given input document. However, the ability to control the length of the summaries is important in many practical applications. In many of such applications, the desired length of the summary is not known beforehand. Moreover, in many domains, only limited training data is available. This makes the length-controllable abstractive summarization specifically a challenging task in low-resource domains. Meanwhile, when it comes to trusting machine-generated summaries, explaining how a summary was constructed in human-understandable terms may be critical. We propose Multi-level Summarizer (`MLS`), a supervised method for constructing abstractive summaries at controllable lengths. The key enabler of our method is an interpretable multi-headed attention mechanism that computes attention distribution over an input document using an array of timestep independent semantic kernels. Each kernel optimizes a human-interpretable syntactic or semantic property. Exhaustive experiments on two low-resource datasets in English language show that `MLS` outperforms strong baselines by up to 14.70% in the METEOR score. Human evaluation of the summaries also suggests that they capture the key concepts of the document at various length-budgets[4].

2.1 Introduction and Related Work

Sequence-to-sequence networks with attention have been extensively applied to abstractive summarization [204, 44, 165, 209, 180, 80, 138]. Most of these methods generate only a single generic summary for an input document. However, controlling the aspects of the generated summary such as length is important in many real-world applications. For example, Online content creators should be able to control the length of the generated summary so that it fits the device that displays it. For scenarios as mentioned above, one may need to employ an ensemble of networks to cover all possible lengths. There are two major challenges in following this approach for real-world applications. First, training sequence-to-sequence networks is a resource-intensive task [218]. To train a network for generating summaries budgeted at length b, we need a parallel corpus of text documents and their gold-standard summaries at length b. Constructing a large enough corpus with summaries budgeted at $b, \forall b$ may not be possible and/or cost-efficient for a number of domains. Most existing works on abstractive summarization train and test their model on large-scale news corpus datasets [165, 94], leaving out several important but low-resource domains [149, 179, 113, 154] where the number of available training documents is limited. Second, the range of possible length-budgets $\mathcal{R}(b)$ may

[4]This is a collaborative work with PhD student Ritesh Sarkhel. Both authors contributed equally to this work.

Input text

police are hunting a man aged between 50 and 60 suspected of robbing a bank in broad daylight and running off with £3,000 in cash. the robbery took place at 12.30pm at a lloyds bank branch in fairwater, cardiff, police said. **detectives have issued cctv images of the suspect, who is 50 to 60, 5ft 9in to 6ft and was wearing black clothing. the white male suspect, who has greying black hair and wore glasses, was captured on camera inside the bank.** detectives said no one was injured during the robbery and they were 'confident' the public would be able to identify the suspect. detective sergeant andy miles, from fairwater cid, said: 'inquiries are continuing to identify the culprit. the cctv is clear and i am confident that members of the public will know his identity...'. (`truncated`)

Summary at compression budget = $\frac{1}{2}$

police are hunting a man aged between 50 and 60 suspected of robbing a bank in broad daylight and running off with £3,000 in cash. the robbery took place at 12.30pm at a lloyds bank branch in fairwater, cardiff, police said. the white male suspect, who has greying black hair and wore glasses, was captured on camera inside the bank. detectives have issued cctv images of the suspect, who is 50 to 60, 5ft 9in to 6ft and was wearing black clothing. detective sergeant andy miles, from fairwater cid, said: 'inquiries are continuing to identify the culprit.

Prototype Summary

robbery took place at 12.30pm at a lloyds bank branch in fairwater , cardiff. detectives have issued cctv images of the suspect , who is 50 to 60. detective sergeant andy miles , from fairwater cid , said : ' inquiries are continuing to identify the culprit.

Figure 2.1: `MLS` *expands the highlighted sentences in the prototype summary to the boldfaced tokens in the input text to construct a summary budgeted at half-length of the input text*

not always be known beforehand. In many scenarios, it can be known as late as during run-time. Therefore, we formalize the summarization task addressed in this paper as follows.

Problem Definition: Given a document S of length N (tokens) and a maximum token budget of b, we aim to construct an abstractive summary s_b that satisfies the following conditions, **`C1:`** information redundancy is minimized in s_b; **`C2:`** coverage of the major topics of S is maximized in s_b; **`C3:`** length of s_b is maximal within the specified budget b without adversely affecting the conditions `C1` and `C2` i.e., $|s_b| \leq b$ & $\nexists s_c$ such that $|s_b| < |s_c| \leq b$. `C1` and `C2` ensure that the properties of a high-quality summary is preserved in s_b, whereas `C3` ensures that s_b is the largest possible summary that can be constructed within budget b without compromising its quality. Note that `C1` and `C2` are seemingly contradictory to each other as the length of the summary increases. Our goal is to find the optimal tradeoff.

Early works on incremental summarization [25, 236] leveraged structural tags supported by document markup languages to generate summaries at various lengths. This constraint makes these methods only applicable for a few type of document formats (e.g. XML, HTML). Incremental sampling of sentences based on a salience score [175, 29] can partially solve this problem by constructing extractive summaries of the input document. We show in Section 2.3 that these sampling-based methods often fail to preserve the desirable properties of a high-quality summary. Among recent works, [118] were the first to propose a supervised method for length-controllable abstractive summarization. Their work was later extended by [70] who introduced the length of a summary as an input to the network. However, instead of exact input, they approximate the length to a predefined value-range, often failing to adhere to the allocated budget in a number of cases. [145] address this issue by proposing a convolutional encoder-decoder network, introducing the desired summary length as an input to the initial state of the decoder. We compare and report its performance on two datasets in our experimental setup in Section 2.3.

In this work, we replace the self-attention mechanism with a lightweight, interpretable alternative to be able to train our network in settings where limited training data is available. Briefly, the main goal of attention mechanism [226] in an encoder-decoder network is to assign a softmax score to every encoder hidden state (based on its relevance to the token being decoded) and amplify those that are assigned high scores through a weighted average. Source-target attention [165] relies on another sequence for computing these scores, whereas self-attention [226, 181] operates over the elements in the current input sequence. A multi-headed attention mechanism allows a neural model to speed up training by enabling parallelization across timesteps. The number of operations in the computation of self-attention, however, scales quadratically with input length, making it a computationally expensive operation for long input sequences. Training such a network for a summarization task would require a large parallel corpus of input documents and their corresponding gold-standard summaries budgeted at b. The role of some of the attention-heads during abstractive summarization is also not transparent [8]. To address these, we replace self-attention with a lightweight, interpretable alternative. Instead of projecting each input sequence multiple times[5] at every timestep, we encode an input sequence only once, using a timestep-independent kernel ($\vec{Q}$) learned in an unsupervised or distantly supervised way from the input document. Each kernel has a human-interpretable syntactic/semantic role. Every attention-head in this multi-headed mechanism computes an attention distribution over the input sequence using a unique kernel $\vec{Q}_i$, recycling it at every timestep. Compared to self-attention, our proposed attention mechanism scales linearly with the input sequence length and leverages a significantly less number of trainable parameters. As we will show in Section 2.3, this allows us to train our network on limited training samples in low-resource datasets.

We propose `MLS` – a supervised method to generate abstractive summaries at arbitrary lengths in this paper. It computes a length-constrained summary s_b budgeted at length b by soft-switching between a copy and expand operation over a prototype summary s_p constructed from the document. The key enabler in this process is an interpretable, multi-headed attention mechanism. We develop a length-aware encoder-decoder network, called the *Pointer-Magnifier* network that leverages this attention mechanism to construct summaries within a specified length. We train our network on limited training samples from two cross-domain datasets: the MSR-Narrative [176] and Thinking Machines dataset [24]. Exhaustive evaluation on a range of success metrics shows that `MLS` performs competitively or better against strong baseline methods. Subsequent human evaluation of summaries generated by `MLS` suggests that they accurately capture the main concepts of the input document. To summarize, some of the major contributions of this work are as follows:

- We propose `MLS`, a supervised approach to generate abstractive summaries of a text document at controllable lengths.
- We develop a length-aware encoder-decoder network that leverages an interpretable, multi-headed attention mechanism to construct length-constrained summaries.

[5] one time each to compute the query, key and value matrix [226] from the input sequence

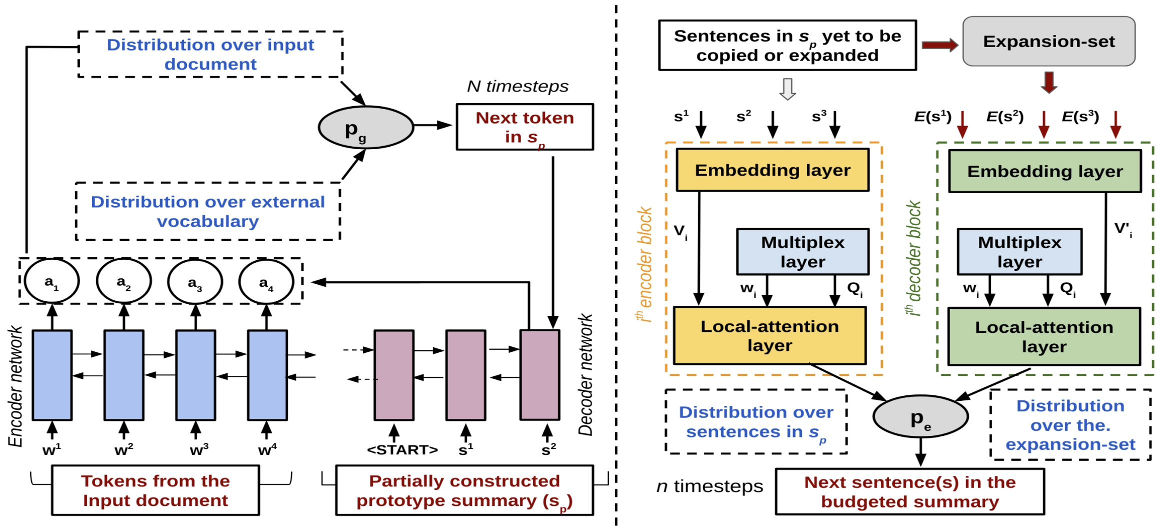


Figure 2.2: *An overview of* MLS *architecture. The PG-Network (left) constructs a prototype summary s_p from the input document. The Pointer-Magnifier network (right) constructs the length-constrained summary from s_p using interpretable sentence-level attention*

- Experimental results on two cross-domain datasets show that trained on limited training samples, MLS was able to generate summaries that are coherent and captured the key concepts of a document.

2.2 Proposed Methodology

MLS constructs a length-constrained summary of a document in two steps. First, it derives a prototype summary s_p from the document, covering its major concepts. Then, it expands or shortens it, depending on the length-budget to create the final summary. We employ a pair of encoder-decoder networks at both steps. For the first step, we extend the PG-network [209]. We develop a length-aware encoder-decoder network for the second step. We describe both steps in greater detail in the following sections.

2.2.1 Generating the Prototype Summary

We extend PG-Network by [209] to construct the prototype summary s_p of a document. We tokenize the document and feed it to the encoder network sequentially. As the encoder hidden states are updated, the decoder network constructs the prototype summary one token at a time by soft-selecting between tokens in the input document and an external vocabulary. The decoding process is guided by an attention distribution[6] computed over the input document and the external vocabulary. An overview of this network is shown in Fig 2.2. We point the readers to the work by See et al.

[6] we closely followed the official implementation at: https://github.com/abisee/pointer-generator

for more background on this network. An example prototype summary is shown in Figure 2.1. Contrary to existing prototype-text guided summarization methods [137, 205], we do not specify the length of the prototype summary as an input of the network, rather infer it by outputting tokens until the EOS token is produced. We discuss the training and parameter settings of the network used in our experiments in Section 2.2.3. It is worth mentioning here that one of the main reasons to select the PG-Network as our architecture of choice for this step is due to its capability to construct a summary by looking up a learned language model. Other networks with similar capabilities can also be used, as this step has a transitive effect on the next phase of our approach.

2.2.2 Constructing the Length-Constrained Summary

To construct a summary within length-budget b, we develop the *Pointer-Magnifier* network: a length-aware, interpretable, encoder-decoder network. An overview of the network is shown in Fig. 2.2. It consists of a multiplex layer, an encoder (yellow rectangles) layer and a decoder (green rectangles) layer. The encoder layer takes the prototype summary constructed in the previous step as input. The decoder layer outputs the final summary. We describe each layer in detail below.

A. The Multiplex Layer and Interpretable Kernels: In an effort to build a transparent network, we embody three qualitative properties that are associated with a high-quality summary in our network. A high-quality summary, (1) maximizes the coverage of the *major topics* (Φ_1) and (2) *keywords* (Φ_2) appearing in the input document, while (3) minimizing the amount of *redundant information* (Φ_3). We encode each property using a semantic kernel ($\vec{Q_i}$), learned using an unsupervised or distantly supervised way from the input document itself. Every kernel plays a unique, human-interpretable syntactic/semantic role in constructing the final summary. One of the key components in this process is the multiplex layer $\mathbb{M}$. Physically, it is a nested matrix of dimensions 3×3 shared between the encoder and decoder layer. Each row in $\mathbb{M}$ contains the following information: (a) a distance-metric ($dist_i$), (b) a scalar value (w_i), and (c) a semantic kernel ($\vec{Q_i}$), where $-1 \leq w_i \leq 1, \forall i$ & $\Sigma_i^3 w_i = 1$. During inference, each of these kernels measures the contribution of every sentence in the prototype summary towards optimizing one of the properties Φ_i, $1 \leq i \leq 3$, mentioned above. w_i represents the relative weight assigned to the property Φ_i in constructing the final summary. We compute the kernels as a preprocessing step.

Defining the Kernels: To encode the property ϕ_1, we define $\vec{Q_1}$ as a matrix of dimensions 3×300, where each row of $\vec{Q_1}$ represents one of the three most dominant topic vectors of the input document as a 300-dimensional vector. We use an unsupervised LDA-based model [21] to derive these topic vectors. Symmetric KL-divergence is used as the distance metric ($dist_1$). Similarly, we encode the property ϕ_2 as a single dimensional vector $\vec{Q_2}$ of length 50, where each vector component represents the relative frequency of one of the 50 most frequent keywords in the input document. We use RAKE [200], a publicly available library to identify the keywords of a document. Symmetric KL-divergence is used as the distance metric ($dist_2$). Finally, we encode ϕ_3 as a matrix $\vec{Q_3}$ of dimensions $p \times 300$, where the ith row of $\vec{Q_3}$ represents an embedding of the ith sentence in the input document. We compute the embedding vector of each sentence using a pretrained model [126] on English Wikipedia corpus. Cosine similarity is used as the distance metric ($dist_3$). Our choice of unsupervised/distantly supervised kernels reflects our motivation (see Section 2.1) to leverage a limited number of training samples from the experimental dataset to construct the final summary. We discuss the role played by each semantic kernel ($\vec{Q_i}$), distance metric ($dist_i$), and weight (w_i)

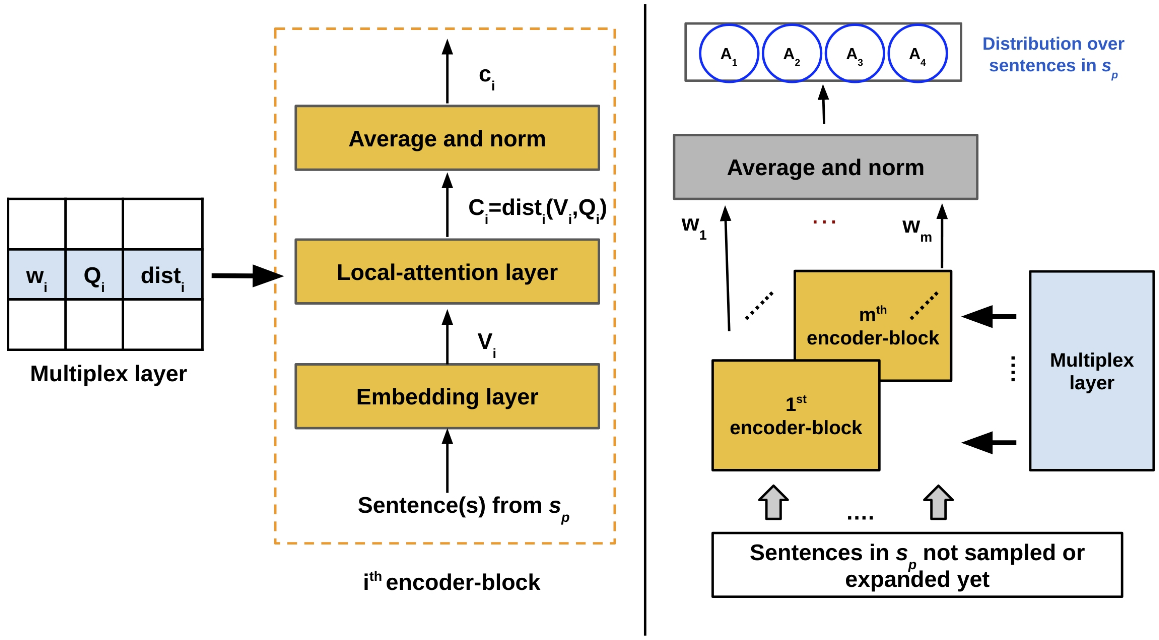


Figure 2.3: *The Encoder layer consists of 3 parallelly stacked encoder-blocks*

in constructing the final summary from s_p in the following section. **B. The Encoder Layer:** The encoder layer consists of 3 parallelly stacked encoder-blocks. Each encoder-block (see Fig. 2.3) contains an *embedding layer* and a *local-attention layer*. At every timestep t, a sentence from s_p is fed into the embedding layer of each of the three encoder-blocks. It computes a fixed-length embedding ($\vec{V_i}$) of the sentence and propagates it to the local-attention layer. Each encoder-block in our network is mapped to a unique triplet ($\vec{Q_i}$, $dist_i$, w_i) in the multiplex layer. To compute local-attention (c_i) attributed to a sentence in s_p by the ith encoder-block, we embed it in the same semantic space as $\vec{Q_i}$ and compute its distance from $\vec{Q_i}$ in that encoding space (Eq. 1).

$$\vec{C_{t,i}} = \frac{1}{r}\Sigma_{j=1}^{r} dist_i(\vec{V_i}, \vec{Q_i}^{\,T}[j]) \tag{2.1}$$

$$c_i = \frac{1}{n_i}\Sigma_{j=1}^{n_i}(\vec{C_{t,i}}[j]) \tag{2.2}$$

In Eq. 1, $\vec{Q_i}$ represents a kernel of dimensions $r \times n_i$ and $\vec{V_i}$ represents an embedding vector of length n_i. The embedding layer represents each sentence in s_p in the same encoding space as the kernel $\vec{Q_i}$ associated with that block. We compute the local-attention c_i by taking a column-wise average of the distance-matrix $\vec{C_{t,i}}$ (Eq. 2). The kernel $\vec{Q_i}$ is reused for all the sentences fed to the ith encoder-block. The distribution $[c_1, c_2...]$ obtained this way is then normalized to derive the local-attention distribution $\vec{C_i}$ over s_p. The final attention distribution ($\vec{A}$) over s_p at timestep t is computed by normalizing the weighted average (Eq. 3) of local-attention distributions computed by each attention-head.

$$\vec{A}^{*} = \texttt{norm}(\frac{1}{m}\Sigma_i^m\,(\vec{C}_i \cdot w_i)) \tag{2.3}$$

It is worth noting here that attributing each encoder-block with a distinct attention-head ensures that there is a dedicated pathway to compute local attentions for every encoder-block. This allows us to parallelize the network and speed-up the decoding process when constructing the final summary.

C. The Decoder Layer: Similar to the encoder, the decoder layer also consists of 3 parallelly stacked decoder-blocks. Each decoder-block contains an *embedding layer* and a *local-attention layer*. Parameters of the i-th encoder-block and i-th decoder-block are shared. We construct a length-constrained summary s_b of the input document by processing each sentence in s_p sequentially. Depending on the remaining length-budget at each timestep, the final summary is constructed by soft-switching between a *copy* and *expand* operation. This process is guided by a sentence-level attention distribution (Eq. 3) computed over s_p. If the copy operation is selected, a sentence from s_p is copied into the final summary, whereas the expand operation replaces a sentence with similar content from the input document in s_b. The original ordering of sentences is preserved.

The Copy Operation: The probability of copying a sentence s from the prototype summary that has not been included in the final summary (s_b) till timestep t into s_b is defined as follows: $P_c(s) = \vec{A}^t[s]$, where $\vec{A}^t$ represents a sentence-level attention distribution over s_p at timestep t. Initialized as $\vec{A}^{*}$ (Eq. 3), we update the attention distribution at each timestep after a copy or expand operation. If $s^* = argmax(P_c(s))$ represents the sentence copied into s_b at timestep t, we update the attention distribution by zeroing out the probability of s^* in $\vec{A}^t$ and renormalizing the resulting distribution.

The Expand Operation: If the length of our prototype summary (s_p) is less than the length-budget b, `MLS` can choose to expand a set of sentences from s_p. For each sentence $s \in s_p$, we define its *expansion-set* $E(s)$ as the sentence n-gram that is most similar to s in the input document. We determine the expansion-set $E(s)$ of a sentence s by using beam-search over all n-grams in the input document that are yet to be included in the final summary. Our search objective being maximizing $score(E) = sim(s, E) \times overlap(s, E)$. The first term in $score(E)$ denotes the average pairwise cosine similarity between s and the sentences in $E(s)$, whereas the second term denotes the fraction of tokens in s that appear in $E(s)$. To minimize across-sentence repetitions in the summary, top 4 candidates identified from the search process are re-ranked [41] based on the number of repeated word bigrams and trigrams if the expansion-set is included in the final summary. We obtained best performance by initializing n with 3 and changing it to 2 at later iterations of the decoding process. If $\vec{v}_i^k$ denotes the embedding-vector of the k-th sentence in $E(s)$ computed by the embedding-layer of the i-th decoder-block, we define the probability of expanding a sentence s from the prototype summary to $E(s)$ in the final summary as follows.

$$\vec{C}^e_{i,k} = \frac{1}{r}\Sigma_{j=1}^r\, dist_i(\vec{v_i}^{\,k},\, \vec{Q_i}^{\,T}[j]) \tag{2.4}$$

$$c^e_{i,k} = \frac{1}{n_i}\Sigma_{j=1}^{n_i}(\vec{C}^e_{i,k}[j]) \tag{2.5}$$

$$\vec{A}^{\,e} = \frac{1}{m}\Sigma_{i=1}^m\,(\vec{c}^{\,e}_i \cdot w_i) \tag{2.6}$$

In Eq. 4, $\vec{Q}_i$ denotes the semantic kernel shared between the i-th encoder-block and decoder-block. We compute the probability of including the kth sentence of $E(s)$ into the final summary by computing its contribution ($c^e_{i,k}$) towards optimizing the qualitative property Φ_i encoded by $\vec{Q}_i$ first (Eq. 5). Repeating this process for all the sentences in $E(s)$, followed by normalization provides us with the distribution $\vec{c}^e_i = (c^e_{i,1}, c^e_{i,2}, ...)$. Here, $\vec{c}^e_i$ represents the probability distribution over $E(s)$. To obtain the expansion probability of a sentence in $E(s)$, we repeat this process for all 3 attention-heads and average them (Eq. 6). The probability $P_e(s)$ of expanding a sentence s from the prototype summary is obtained by averaging the expansion probability of all sentences in $E(s)$. Once a sentence s has been expanded into the final summary, we update the attention distribution by zeroing out the probability at s and renormalizing the resulting distribution.

Soft-Selection between Copy and Expansion: We define the probability $p_o(s)$ of selecting between the copy and expand operation for a sentence s in the prototype summary as follows.

$$p_o(s) = \alpha \times P_e(s) + (1 - \alpha) \times P_c(s) \tag{2.7}$$

$$\alpha = \begin{cases} 0 & if\ b \leq len(s_b{}^*) \\ \mathtt{max}(P_e(s), P_c(s)) & if\ b > len(s_b{}^*) \end{cases} \tag{2.8}$$

In Eq. 8, $s_b{}^*$ denotes the partially constructed summary till timestep t. If the length-budget b is smaller than the length of the prototype summary s_p, the probability of including a sentence from s_p into the final summary depends on the attention distribution $\vec{A}^t$ over sentences in s_p that are not included in the final summary till timestep t. In all other scenarios, α acts as a soft-switch between copying or expanding a sentence in s_p. A sentence can be expanded only if doing so does not exceed the length-budget. Once the probability of each sentence (and/or its expansion set) has been computed, the decoder attends to the position with the highest probability and copies/expands it into the final summary. Generation stops once $len(s_b{}^*)$ reaches b. We observed that the probability of expanding a sentence from the prototype summary (instead of copying it) increases with the allocated length-budget.

2.2.3 Training the Networks

We trained PG-Network and the Pointer-Magnifier network separately on a NVIDIA Titan-XP GPU with a batch size of 16. We pretrained the PG-Network on the CNN-DailyMail dataset [165] and then fine-tuned it on training samples of our experimental datasets. Using the evaluation script provided by [165], we obtained a training set of 287,226 pairs and validation set of 13,368 pairs for this dataset. All encoder-decoder weights were allowed to be updated during fine-tuning stage, following a L1-transfer [177] of weights from the pretrained network. The external vocabulary used in both pretraining and fine-tuning stage consisted of 80K most frequent tokens in the training samples of the CNN-DailyMail dataset, our experimental dataset or both. Learning-rate and initial accumulator values were set to 0.15 and 0.1 respectively. We used Adagrad [62] to train the network. The encoder was fed a maximum 400 tokens and the decoder generated 100 tokens during pretraining. These values were increased to 500 and 200 respectively during fine-tuning. To prevent overfitting, we stopped training after 3000 iterations during the fine-tuning stage. With respect to the Pointer-Magnifier network, we learn the optimal values of $w_i, 1 \leq i \leq 3$ associated with each attention-head by grid-searching over the interval [-1,1] with the learning objective of maximizing ROUGE-1 score on the validation set. The optimal weights assigned to the attention-head corresponding to topic-coverage (ϕ_1) and keyword-coverage (ϕ_2) were positive, whereas information redundancy (ϕ_3) was assigned a negative weight for both of our datasets.

Index	Dataset	Size	Max	Median	Mean
D1	MSR Narrative	476	130	15	18.65
D2	Thinking Machines	186	82	33	33.23

Table 2.1: *The min, max, median, and average number of sentences in datasets D1 and D2*

2.3 Experiments

We seek to answer three key questions in our experiments. Given a length-constrained summary s_b, (a) how similar is s_b to a gold-standard summary?, (b) is it coherent and representative of the input document? and (c) how abstractive is s_b? We answer the first two questions by evaluating the summaries generated by `MLS` over a range of success metrics on datasets belonging to two low-resource domains. We also conduct a user study to measure how representative are the summaries with respect to the input documents. A representative summary covers the main topics of the document. We answer the third question by computing the percentage of n-grams in s_b that do not appear in the input document and/or generated from the external vocabulary.

A. Datasets: We evaluate `MLS` on two publicly available datasets from two low-resource domains: the MSR-Narrative [176] (D1) dataset and the Thinking-Machines [24] (D2) dataset. The MSR-Narrative dataset contain personal stories shared by users on a social networking website. The Thinking-Machines dataset, on other hand, contains position papers on a popular scientific topic published in an educational website. Each document in both datasets is paired with a gold-standard summary. We randomly selected 25% document-pairs to construct the training set and 10% document-pairs to construct a validation set for both datasets. The rest comprised the test corpus. We present an overview of some of the important properties of both datasets in Table 2.1.

B. Metrics: We compare the summaries constructed by `MLS` against gold-standard summaries using METEOR [11] and ROUGE [133] scores[7]. The average F_1 score of ROUGE-1, ROUGE-2 and ROUGE-L metrics obtained for both datasets are shown in Table 2.2. To measure the *representativeness* of a summary, we compute the average KL-divergence score between the top-3 topic vectors of a summary and its input document. Following [215], we measure the coherence of a summary by computing the average cosine similarity between consecutive sentences. We report the absolute difference between the coherence score computed for a summary and its input document in Table 2.3. We also report the KL-divergence score between sentiment vectors of a summary and the input document to check for potential biases in its polarity distribution. We used a publicly

[7]We used py-rouge [8] and the NLTK library to compute the ROUGE and METEOR score respectively.

Dataset	Metric	Budget = 1/32				Budget = 1/16				Budget = 1/8				Budget = 1/4				Budget = 1/2			
		MLS	**A1**	**A2**	**A3**	**MLS**	**A1**	**A2**	**A3**	**MLS**	**A1**	**A2**	**A3**	**MLS**	**A1**	**A2**	**A3**	**MLS**	**A1**	**A2**	**A3**
D1	**ROUGE-1**	**45.99**	23.44	37.46	41.65	**45.99**	30.50	37.68	43.07	**45.99**	31.27	38.05	43.50	**46.11**	41.86	43.95	44.10	**45.67**	40.67	41.13	45.50
	ROUGE-2	**35.97**	14.79	22.59	30.65	**35.97**	20.77	25.50	30.65	**35.98**	22.95	29.14	33.50	**35.60**	27.57	32.36	34.50	**36.70**	29.38	31.02	35.02
	ROUGE-L	**40.89**	21.35	32.38	37.65	**42.50**	27.9	33.07	38.92	43.01	36.25	37.62	**43.50**	**42.83**	38.83	40.95	41.07	40.18	39.60	40.74	**41.50**
	METEOR	**47.12**	18.91	24.22	45.51	**47.12**	13.07	25.02	45.60	**46.50**	20.89	30.86	43.88	**46.61**	27.26	33.05	44.65	**45.71**	27.84	32.95	45.39
D2	**ROUGE-1**	**40.25**	16.20	21.06	35.60	**40.0**	17.08	22.0	36.0	**40.25**	22.59	28.10	39.72	**41.01**	23.55	27.83	38.50	**44.36**	29.53	32.75	44.06
	ROUGE-2	**33.25**	11.25	17.22	26.50	**34.50**	12.0	16.75	30.05	**35.67**	14.60	19.01	31.80	**36.0**	17.90	20.06	31.0	**38.70**	20.67	23.46	36.44
	ROUGE-L	**37.17**	14.50	19.06	33.67	**37.0**	15.60	20.55	35.70	**37.05**	21.65	20.26	34.33	**37.96**	21.87	22.60	32.77	**41.50**	26.04	27.17	39.75
	METEOR	**40.22**	12.68	24.33	35.05	**44.82**	15.17	23.22	42.90	**44.82**	11.96	30.79	42.0	**42.88**	24.20	21.83	38.05	44.79	28.08	25.82	**45.70**

Table 2.2: *ROUGE and METEOR scores of the budgeted summaries constructed by* `MLS` *(highlighted column) and the baseline methods for the MSR-Narrative (D1) and Thinking Machines (D2) dataset*

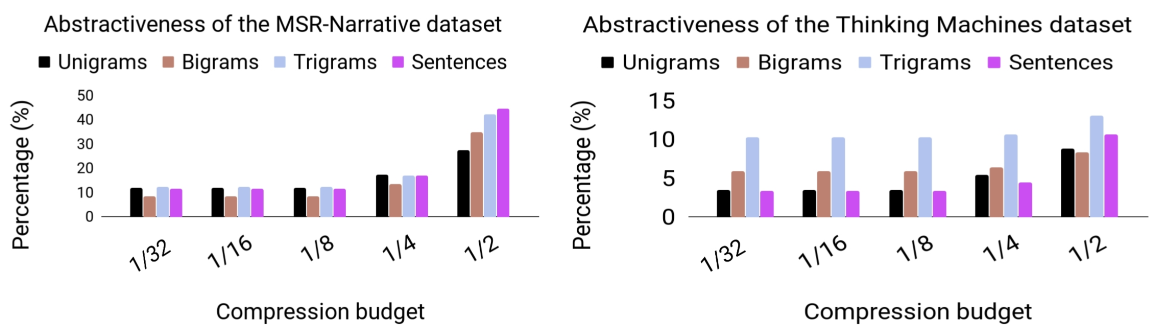


Figure 2.4: *Abstractiveness of* `MLS` *generated summaries*

available library [98] to derive the sentiment vectors.. Note that, lower values of $\Delta Coherence$ and KL-divergence score are desirable for a high-quality summary.

C. Baselines: We compare `MLS` against three baseline methods. Two of them follow a sampling based approach, while our final baseline method employs a convolutional network to construct length budgeted summaries. Our first baseline (**A1**) follows a systematic sampling based approach to construct length-controlled summaries. Initialized with a randomly selected sentence from the first k-1 sentences of the input document, it constructs the final summary by including the k-th sentence from the last sampled position. We set k = 3 in all of our experiments for both datasets. Sampling terminates when the budget limit is exceeded or the end of document is reached. Our second baseline method (**A2**) follows a weighted graph-based sampling strategy to construct budgeted summaries. It represents each sentence in the input document as a node in an undirected, complete, weighted graph. The weight assigned to an edge in this graph is equal to the pairwise cosine similarity between the connecting nodes. To construct the budgeted summary, we sample the top-K nodes of this graph using a weighted PageRank algorithm [160]. Sampling stops when the budget is reached. Our third and final baseline method (**A3**) is a convolutional approach proposed in [145]. It is a sequence-to-sequence network with Gated Linear Units [53] that takes the desired length of a summary as an additional input to the initial state of the decoder network. Similar to our training protocol, we pretrain this network on the CNN-DailyMail dataset first and fine-tune it on the training samples from both of our experimental datasets. We allowed all weights to be updated during the fine-tuning phase.

Dataset	Metric	Budget = 1/32				Budget = 1/16				Budget = 1/8				Budget = 1/4				Budget = 1/2			
		MLS	A1	A2	A3	MLS	A1	A2	A3	MLS	A1	A2	A3	MLS	A1	A2	A3	MLS	A1	A2	A3
D1	**Topic**	**0.12**	0.28	0.29	0.21	**0.12**	0.27	0.27	0.20	**0.12**	0.26	0.23	0.15	**0.13**	0.21	0.19	0.18	**0.13**	0.21	0.21	0.18
	Sentiment	**0.09**	0.22	0.19	0.11	**0.09**	0.23	0.15	0.13	**0.09**	0.19	0.15	0.12	**0.1**	0.14	0.12	**0.1**	0.16	**0.07**	0.17	0.13
	ΔCoherence	**0.08**	0.3	0.20	0.11	**0.08**	0.26	0.18	0.09	0.08	0.21	0.11	**0.07**	**0.09**	0.13	0.10	0.12	0.1	**0.06**	0.09	0.1
D2	**Topic**	**0.05**	0.27	0.24	0.15	**0.05**	0.27	0.25	0.16	**0.05**	0.17	0.2	0.12	**0.05**	0.08	0.08	0.11	0.03	0.03	**0.02**	0.10
	Sentiment	**0.03**	0.24	0.16	0.10	**0.03**	0.21	0.13	0.07	**0.03**	0.12	0.15	0.04	**0.03**	0.06	0.08	0.05	0.04	**0.02**	0.03	0.03
	ΔCoherence	**0.03**	0.27	0.20	0.05	**0.03**	0.18	0.12	0.10	**0.03**	0.09	0.09	0.05	**0.03**	0.05	0.05	0.06	0.04	**0.03**	**0.03**	0.04

Table 2.3: *Coherence and completeness of the budgeted summaries constructed by* `MLS` *(highlighted column) and the baseline methods for MSR-Narrative (D1) and Thinking Machines (D2) dataset*

2.3.1 Results and Discussion

We report the performance of all competing methods at five length-budgets. We specify the length-budget to construct a summary as a product of the number of tokens in the input document and a compression-budget $c \in \{\frac{1}{32}, \frac{1}{16}, \frac{1}{8}, \frac{1}{4}, \frac{1}{2}\}$. Results from our experiments are presented in Tables 2.2 and 2.3. The best performance achieved for each metric is boldfaced. We highlight some of our key findings below.

Qualitative Evaluation at five Compression Budgets: In general, the abstractive methods (`MLS` and A3) outperform sampling-based approaches (see Table 2.2) on both datasets. `MLS` performs consistently well on all budgets, although performance is relatively better on smaller budgets. We obtain an absolute improvement of 4.34% and 4.65% in ROUGE-1 score & 1.61% and 5.17% in METEOR score over the convolutional baseline (A3) for datasets D1 and D2 at compression budget = $\frac{1}{32}$. At higher budgets, our performance was comparable with A3. In terms of coherence, `MLS` performs comparably or better than A3 (see Table 2.3). Smaller $\Delta Coherence$ score than A1 and A2 suggests that `MLS` generated more coherent summaries than these two baseline methods. Small KL-divergence between the topic distribution of a budgeted summary and input document shows that `MLS` generated summaries are representative of the document for both datasets. In fact, topic-coverage in summaries generated by `MLS` is at least 75% better than the convolutional baseline (A3) [145], although performance becomes comparable at larger budgets as more sentences from the prototype summary are expanded to make the final summary. `MLS` outperfoms A1 and A2 in terms of staying true to the sentiment distribution of the input document. This can be seen from the small KL-divergence scores obtained for the sentiment distribution achieved by `MLS` in Table 2.3.

`MLS` generated summaries were more abstractive at higher budgets (Fig. 2.4). At compression budget = $\frac{1}{2}$, 27.35% tokens in the summaries constructed for dataset D1 and 8.75% tokens for dataset D2 were contributed by the external vocabulary.

Ablative Analysis: To investigate the effects of pretraining on end-to-end results, we compare the ROUGE-1 score of summaries constructed by `MLS` against an ablative baseline MLS*. It is identical to `MLS` except that the PG-Network was not pretrained. In our second experiment, we compare `MLS` against MLS+, an ablative baseline that constructs the prototype summary following a greedy heuristics [175] instead of the PG-network. `MLS` outperforms both baselines (Fig. 2.5) on both datasets, thereby establishing that using PG-Network in our framework and pretraining it on the CNN-DailyMail dataset improved the quality of our final summaries. Finally, to investigate the effects of the semantic kernels introduced in the Pointer-Magnifier network, we iteratively replaced

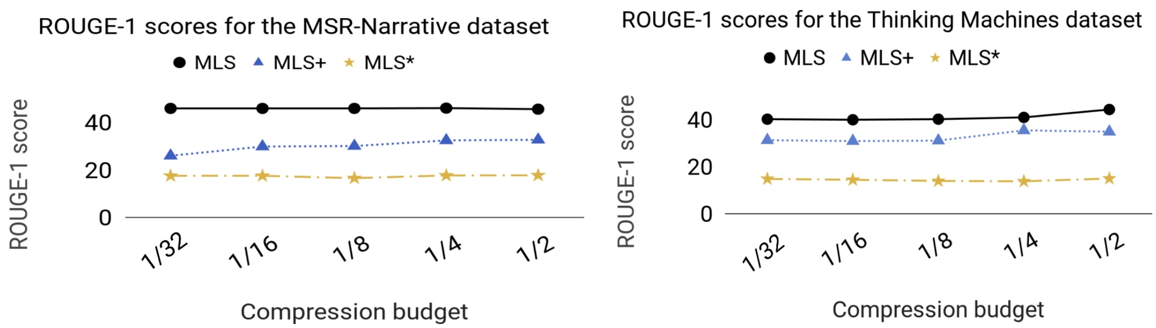


Figure 2.5: *ROUGE-1 score of* `MLS` *and the ablative baselines MLS+ and MLS* on datasets D1 and D2*

each of the three semantic kernels (Section 2.2.2) with a randomized kernel by shuffling its rows and columns.

We observed an absolute decrease of up to 4.30% in ROUGE-1 score and 3.75% in METEOR score for $\vec{Q}_3$, with bigger impacts in performance at higher length-budgets. Replacing $\vec{Q}_2$ with a randomized kernel, on other hand, decreased the average $\Delta Coherence$ score by approximately 45% for dataset D1 and 30% for D2 for summaries constructed at compression budget = $\frac{1}{2}$, i.e. half-length of the input document.

Human Evaluation of Length-Controlled Summaries: We conducted a study to evaluate the completeness of the summaries constructed by `MLS`. More specifically, we considered a scenario where the user needs to complete a fact checking task. We chose three documents from both datasets randomly and asked each participant to verify the presence of some key facts of the document in the summaries constructed by `MLS` and/or a baseline method.

Index	Dataset	MLS	A2	A3	NC	FC
D1	**Accuracy**	**0.88**	0.55	0.55	0.0	**0.88**
	Duration (s)	**36.7**	43.69	69.08	12.0	75.6
D2	**Accuracy**	0.55	0.44	**0.66**	0.0	**0.88**
	Duration (s)	70.24	**68.9**	96.47	20.95	132.86

Table 2.4: *Mean accuracy and completion time using* `MLS`*, A2, A3, No-content (NC) and Full-content (FC) settings*

Each participant was instructed to complete the task solely based on the content of the summary and not depending on any previous knowledge. For example, the question "*Does the story tell us why the narrator was fired?*" was paired with the following summary– "*I tried to return a lost wallet to a customer who accused me of stealing it and then grabbed my hair. We got in a physical fight and I was fired from my job*". The participants had to chose between 'Yes', 'No", and "More information required". If a participant selected the third option, a longer summary was shown with the same question. The task was terminated otherwise. In addition to `MLS`, A2 (the stronger extractive baseline in our experimental setup) and A3, we add two extreme settings: (a) the full-content setting in which the original document was shown, and (b) the no-content setting where no textual content (other than the question itself) was shown to a participant. The full-content setting ensured that the question could indeed be answered from the article, whereas the no-content setting ensured whether the questions contained any hint about the answer.

The task started by showing each participant a summary generated at compression budget = $1/32$. If they opted for more information to be shown, we provided a summary generated by the same method by doubling the compression budget each time until the user responded with a 'Yes' or 'No' or we reached the budget of $1/2$. The key intuition here is that if users are given a complete and representative summary, they should be able to answer the questions accurately, as a good summarization model would pick up the key concepts of the document even at shorter length-budgets, without requiring for it to be expanded further. With this in mind, we recorded task completion time and user response for each treatment. All budgeted summaries were constructed beforehand. We invited 22 graduate students to participate in the study. Each participant was shown summaries generated by at most two different methods in random order. No information on the method used was revealed to a participant at any stage. To prevent information retention, each participant was shown a summary generated from the same document only once. Using a balanced, incomplete block design [6], each of the 10 settings (5 methods $\times$ 2 datasets) was assigned to 3 subjects. The average accuracy and task completion time recorded for each treatment is shown in Table 2.4. The accuracy of the no-content setting is 0 for both datasets, indicating that the questions did not contain any hint to the correct answer, whereas the full-content setting shows that overall the questions could have been answered from the original documents. When using summaries generated by `MLS`, the participants responded as accurately as the Full-content setting on dataset D1, while being more than two times faster, outperforming A2 and A3. For dataset D2, participants were more accurate using summaries constructed by `MLS` than A2. MLS performed better than A3 on one document, comparable on one and worse on one document, with an average accuracy of 0.55.

2.4 Conclusion

We have proposed `MLS`, a supervised approach to construct abstractive summaries at controllable lengths. Following an extract-then-compress paradigm, we develop the Pointer-Magnifier network – a length-aware, encoder-decoder network that constructs length-constrained summaries by shortening or expanding a prototype summary inferred from the document. The key enabler of this network is an array of semantic kernels with clearly defined human-interpretable syntactic/semantic roles in constructing the summary given a budget-length. We train our network on limited training samples from two cross-domain datasets. Experiments show that the summaries constructed by `MLS` are coherent and reflectively capture the main concepts of the document. Our human evaluation study also suggest the same.

Chapter 3: Domain-guided summarization of privacy policies

Companies' privacy policies are often skipped by the users as they are too long, verbose, and difficult to comprehend. Identifying the key privacy and security risk factors mentioned in these unilateral contracts and effectively incorporating them in a summary can assist users in making a more informed decision when asked to agree to the terms and conditions. However, existing summarization methods fail to integrate domain knowledge into their framework or rely on a large corpus of annotated training data.

In this chapter, we first conduct a user study to examine users' comprehension of the policies. We investigate whether the presentation format of the policies impacts their accuracy in a policy comprehension task. We also examine the impact of presentation format on the user's agreement level. Our results show that users are more accurate with a highlighted presentation form where riskier content are color-coded. They also trust this presentation form more than the grade format and short plain English summary.

Inspired by this finding, and to further address the information overload problem in this domain, we propose a hybrid approach to identify sections of privacy policies with a high privacy risk factor. We incorporate these sections into summaries by selecting the riskiest content from different privacy topics. Our approach enables users to select the content to be summarized within a controllable length. Users can view a summary that captures different privacy factors or a summary that covers the riskiest content. Our approach outperforms the domain-agnostic baselines by up to 27% in ROUGE-1 score and 50% in METEOR score using plain English reference summaries while relying on significantly less training data in comparison to abstractive approaches.

3.1 Introduction and Related Work

Privacy policy and terms of service are unilateral contracts by which companies are required to inform users about their data collection, processing, and sharing practices. Users are required to agree to abide by the terms before they can use any service. However, many users do not read or understand these contracts [51]. Thus, they often end up consenting to terms that may not be aligned with legislation such as the General Data Protection Regulation (GDPR)[9] [173]. This behavior is often because these contracts are too long and difficult to comprehend [156]. Summarization is an intuitive way to assist users with conscious agreement by generating a condensed equivalent of the content. Broadly, there are two main lines of summarization systems: *abstractive* and *extractive*.

[9]https://eugdpr.org/

The abstractive paradigm [204, 165, 40, 209, 222, 180, 207] aims to create an abstract representation of the input text and involves various text rewriting operations such as paraphrasing, deletion, and reordering. The extractive paradigm [167, 239] on the other hand, creates a summary by identifying and subsequently concatenating the most important sentences in the document. The abstractive systems are more flexible while the extractive models enjoy better factuality [30]. However, existing summarization techniques perform poorly on contracts. Unsupervised methods [160, 85] rely on structural features of documents, such as lexical repetition, to identify and extract important content. These heuristics work poorly on the legal language used in contracts [154]. Supervised methods [209, 81, 180] can learn to cope with the features of a particular domain. However, training these complex neural summarization models with thousand of parameters requires a large corpus of documents and their summaries. Currently existing corpora in the legal domain are not large enough to train such models. We propose a hybrid approach for extractive summarization of privacy contracts: using existing annotated resources, we train a classifier to predict which pieces of content are most relevant to users [51]. In particular, we identify parts of the contract which place users at risk by imposing unsafe data practices on them, such as selling email addresses to third parties or allowing the company to appropriate user-generated content. Next, we use this risk classifier for content selection within an extractive summarization pipeline. The classifier is substantially less expensive than learning to summarize directly but enables our approach to outperform a selection of domain-agnostic unsupervised summarization methods.

Prior computational work on privacy policies has used information extraction and natural language processing methods to classify segments of these documents into different data practice categories [139, 231, 252]. Another trajectory of work has focused on presenting a graphical "at-a-glance" description of the privacy policies to the user. For example, PrivacyGuide [223] and PrivacyCheck [240] define a few privacy factors and map each factor to a risk level using a data mining model. Relying on these "at-a-glance" description methods raises several concerns. First, there is no way for the user to check the factuality of the predicted risk classes or interpret the reasoning behind them. Moreover, users tend to have an easier time comprehending the content when provided in natural language. Researchers also have focused on assigning a risk factor–green, yellow, or red–to each segment of the privacy policies [170, 87]. However, summarizing the text may benefit users more than directly presenting the classifier output.

In this Chapter, we first, conduct a comprehensive user study to compare some of these presentation formats. We examine how the presentation format impacts users' comprehension of the policy and their agreement level. Essentially, we expose them to 4 different presentation forms (i) grade overview, (ii) plain English summary (abstractive), (iii) highlighted policy and the (iv) full text. We next, ask a few privacy-related questions from users and measure their accuracy in responding to these questions.

Inspired by our findings, we propose a hybrid approach for identifying and summarizing sections of the privacy policies with a high privacy risk factor. The first module of our framework extends prior work [170, 87] to highlight segments of privacy policies that have a higher risk. We employ a pre-trained encoder and convolutional neural network to classify sentences of the contracts into different risk levels. To address the limitations of previous work, we incorporate the domain information predicted by the classifier in the form of a summary by comparing a risk-focused and a coverage-focused content selection mechanism. The coverage-focused selection mechanism aims to reduce information redundancy by covering the riskiest sentence from each privacy topic.

Service	Grade
DuckDuckGo	A
Telegram	B
Wikipedia	B
New York Times	C
Foursquare	C
Quora	D
Paypal	E
Amazon	E
Instagram	E
Uber	E

Table 3.1: The list of companies in our user study and their corresponding grade level (A-E).

We evaluate the effectiveness of employing a classifier in identifying the domain knowledge for summarization. We also evaluate the quality of summaries extracted by our two content selection criteria. Using our approach users can view a summary that captures different privacy factors or a summary that covers the riskiest content. We release our dataset of 151 privacy policies annotated with risk labels to assist future research.

We present our user study in Section 3.2 and our hybrid pipeline for extractive summarization of the policies in Section 3.3.

3.2 Human-Centered Analysis of Privacy Policy Comprehension

To understand the impact of presentation format on user's comprehension of privacy policies and their preference for presentation forms we conducted a user study.

Service Selection: We selected 10 popular services to include in our user study from TOSDR [10]. TOSDR is a website dedicated to rating and explaining the privacy policy of companies in plain English. A list of the selected services and their risk level (A-E) on TOSDR is shared in Table 3.1. Grade "A" companies are the safest class of services in TOSDR classification while grade "E" companies raise serious privacy concerns [11].

Presentation Format: For each service, we collect the annotation for their grade, and points written by staff on TOSDR. Each TOSDR point is a statement about data sharing and practices of the company in plain English. For example, case ID 10493 for Facebook states that *"They store data on you even if you did not interact with the service"*. Each case is linked to a certain segment of the terms of service or privacy policy document of service. An overview of information crawled from TOSDR and how it is used to create different presentation forms is shared in Figure 3.1. Next, we explain how we create each presentation format.

[10]TOSDR https://tosdr.org

[11]See more details about TOSDR risk classification in https://tosdr.org/classification

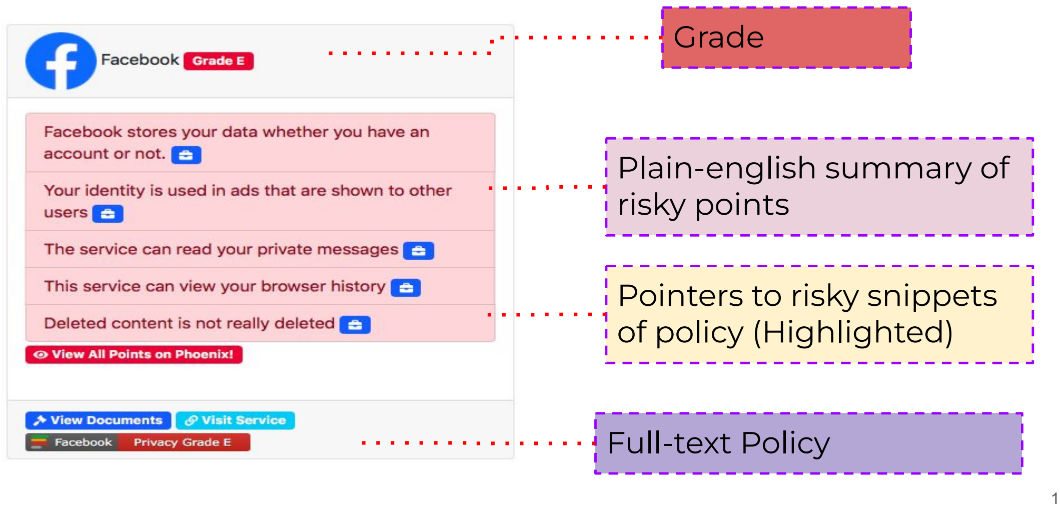


Figure 3.1: Annotation for Grade and Points for Facebook on TOSDR

- To create the *Grade* presentation format, we simply present the service name and grade and a description of different risk grades to the user. The motivation behind this presentation form is to test users' assumptions about a particular service. In addition, it gives users a comparative measure of where companies stand in terms of their privacy measures.

- To create the *Summary* presentation format, we simply present the points written for each company to the user. We use no color coding for this presentation format.

- To create the *Full-text* presentation form we simply concatenate the privacy policy, data policy, terms of service, and cookie policy of the services. We use the same font, size, and spacing format used in the website of the corresponding service.

- To create the *Highlighted* presentation format, we highlight the segments of the Full-text presentation format that are linked to a point with yellow. We inform the users that highlighted segments are marked as "risky" or "important" by other users.

In this study, we recruited 365 workers to participate in our study. Workers are recruited through MTurk. MTurk is a platform that connects "requesters" (people who need tasks completed) with "workers" (people willing to complete the tasks for monetary compensation). The study was posted to MTurk via CloudResearch [136] [12]. CloudResearch is an online platform linked to MTurk that provides additional data collection and filtering features to ensure high data quality.

Workers are based in the U.S. and have a high-school degree or above. In addition, English is their first language. They were paid 2.5$ for their participation.

[12]cloudreasearch.com

Presentation Format	F-score	Need More Info	Preference	Agreement	Time (Min)
Grade	42.22	22.53	**2.88**	2.87	**6.23**
Summary	49.13	14.71	2.59	2.60	10.72
Highlighted	**59.13**	**4.39**	2.60	**3.47**	17.28
Full-Text	59.06	6.51	2.73	3.27	16.79

Table 3.2: Results of user's comprehension, preference, agreement level and task completion time for different presentation forms.

Task Description: Users were first shown information about the terms of service of one of the 10 prominent companies in our study. They were informed that they may not return to this page later on and they will be asked a few questions about the content they see. The information was presented in one of the 4 formats discussed above (full-text, highlighted, plain English summary, or grade). Then users were instructed to answer 5 privacy questions about the terms [13]. All questions had multiple correct answers and users were instructed to choose all that apply. Users were also instructed not to use any external resources for answering the questions. Users had a maximum of 30 minutes to complete the task. The questions are shared below:

- Q1: What type of information does the site collect?
- Q2: What is the purpose of collecting your data?
- Q3: Who can access your information?
- Q4: What are your rights?
- Q5: How long does the company keep/use your personal info?

To create the answer keys for these questions, two of the authors and a legal expert read the full-text policy and answered the questions independently. Next, the answers were discussed and annotators resolved any potential disagreements.

Users also were asked a few questions about their preferences and trust. They were asked to indicate how much they prefer the format they saw and how likely they are to agree to the terms after going through the presented information. They had to answer on a scale of 1-5.

Privacy Comprehension: To measure users' comprehension, we measured the precision and recall of users in the reading comprehension task. We report the F-score which is the harmonic mean of the precision and recall in Table 3.2(averaged across 5 privacy questions). In addition, for each presentation format, we report the average percentage of users that indicated they need more information to be able to answer the questions. We report the average value for 5 questions. This value is presented in the *Need more info* column in Table 3.2.

As it can be seen in the Table, the comprehension of the policies is in general low. The highest comprehension level is achieved when users looked at the highlighted and full-text format and

[13]These questions are inspired by GDPR rights

are only 59.13 and 59.06 respectively. Users' accuracy when looking at the grade and summary formats is lower. This is expected for the grade format as in this case, users are not provided any information about the service and only answer the questions based on their assumptions. For the summary format, the F-score is slightly higher (49.13). We conjecture, that this is mostly because the presented summary is generic and is compiled by combining points written by staff at TOSDR but are not directly related to questions we asked in the end. We suspect that a summary guided by questions about GDPR rights would increase users' comprehension of the policies. We present our pipeline for query-guided summarization of the policies in Chapter 4.

Users had a better comprehension of the highlighted and the full-text format. These results are statistically significant (one-way Anova, P-value < 0.001). This observation contradicts what previous study by Obar and Oeldorf-Hirsch [173] reports. In their study, they noticed that 97% of users do not read or understand privacy policies. This can be because, in our study, users were mildly motivated to read the policies as they were notified about the quiz at the end. This makes our setting different than the real-world interaction with policies. The values for needing more information follow the same trend.

Users' Preference in Format: The results for whether they prefer the presentation form they interacted with are shown in column *preference*. Values range from 1-5. While user comprehension of the grade format was low, they preferred this format more than other ones on average. However, the results were not statistically significant. In addition, we also observed that users have different preferences for format depending on the underlying service, in other words, they did not consistently prefer one format over the other.

Users' Trust and Agreement: The average value for whether users agree to abide by the policy after this interaction, is shared under *Agreement* in Table 3.2. Users feel less comfortable agreeing to the terms when they are given too little amount of information (grade and summary). They were more likely to agree to the terms when presented in the highlighted or full-text format. These results are statistically significant (one-way Anova, P-value < 0.001). However, note that in a real-world interaction with a service users are more motivated to agree to the terms as they want to use the service immediately afterward. However, this is not the case with this study and users are less motivated to agree. Thus, we expect the values to be higher if users were to use the service after their interaction with the policy.

Task Completion Time: Not surprisingly it takes users longer to read the longer formats. However, as mentioned before, users were mildly motivated to read the longer formats due to the quiz at the end. Interestingly highlighting riskier content in the policy increased users interaction time with the policy. However, this observation is not statistically significant.

As discussed earlier, the highlighted presentation format leads to a better understanding of the policy. Motivated by this finding, and to further address the information overload problem in this domain, in the next section we present a hybrid approach for extractive summarization of privacy policies.

3.3 Extractive Summarization of Privacy Policies

Given a privacy policy document D consisting of a sequence of n sentences $\{s_1, s_2, ...s_n\}$ and a sentence budget m such that $m < n$ our summarization model extracts a risk-aware summary with m sentences. For each sentence $s_i \in D$ we predict a binary label y_i (where a value of 1 means s_i is included in the summary). We achieve this by computing an inclusion probability $p(y_i|s_i, D, \theta)$ for each sentence s_i. θ are the model's parameters. We aim to maximize the inclusion probability for risky sections of the privacy policies and minimize it for non-risky sections. We also would like to cover different privacy factors within the sentence budget m by reducing the redundancy. The main intuition behind our proposed approach is that users when going through the privacy policies are most interested in knowing how their information can potentially be abused [51]. Thus, a condensed equivalent of the terms should include such risky sections. Next, we explain the architecture of our risk prediction model and our content selection mechanisms.

3.3.1 Risk Prediction

Given the content of privacy policies, the first step in our framework is to identify the associated risk class with each sentence of the contract. We rely on a crowd-sourcing project called TOS;DR to automatically annotate 151 privacy contracts. TOSDR has annotated several snippets of privacy contracts based on the average Internet user's perception of risk. We explain our dataset extraction in Section 3.4. We use this dataset to train our risk classifier.

Prior research has exploited word embeddings and Convolutional Neural Networks (CNN) for sentence classification [48, 119, 104, 249]. These simple architectures achieve strong empirical performance over a range of text classification tasks. Our model is a slight variant of the CNN architecture proposed in [48].

Model architecture: Let $s_j = \{t_1, t_2, ...t_n\}$ be the j-th sentence in the contract D and $v_i \in R^d$ be the d-dimentional vector representation of token t_i in this sequence. Word representations are output of a pretrained encoder [185] and will be discussed in the next Section. We build the sentence matrix $A \in R^{n \times d}$ by concatenating the word vectors v_1 to v_n:

$$A_{1:n} = v_1 \oplus v_2 \oplus ...v_n$$

Following [48] we apply convolution filters to this matrix to produce new features. The length of the filters is equal to the dimensionality of the word vectors d. The height or region size of the filter is denoted by h and is the number of rows (word vectors) that are considered jointly when applying the convolution filter. The feature map $c \in R^{n-h+1}$ of the convolution operation is then obtained by repeatedly applying the convolution filter w to a window of tokens $t_{i:i+h-1}$. Each element c_i in feature map $c = [c_1, c_2, ...c_{n-h+1}]$ is then obtained from:

$$c_i = f(w \; . \; A[i : i + h - 1] + b)$$

where $A[i : j]$ is the sub-matrix of A from row i to j corresponding to a window of tokens t_i to t_j and "." represents the dot product between the filter w and the sub-matrices. $b \in R$ represents the bias term and f is an activation function such as a rectified linear unit. We use multiple kinds of filters by using various region sizes. This extracts various types of features from bigrams, trigrams, and so on. The dimensionality of the feature map c generated by each convolution filter is different for sentences with various lengths and filters with different heights. We apply a max-over-time [48] pooling operation to downsample each feature map c by taking the maximum value over the window defined by a pool size p. The max-pooling operation naturally deals with variable sentence lengths. The outputs generated from each filter map are concatenated to build a fixed-length feature vector for the penultimate layer. This feature vector is then fed to a fully connected softmax layer that predicts a probability distribution over the risk level categories. We apply dropout [95] as a means of regularization in the softmax layer. Our objective is to minimize the binary cross-entropy. The trainable model parameters include the weight vectors w of the filters, the bias term b in the activation function, and the weight vector of the softmax function. We minimize the loss using *Stochastic gradient descent* and back-propagation [203].

Pretrained Word Vectors: Prior research indicates that better word representations can improve performance in a variety of natural language understanding (NLU) tasks [184]. We use ELMo [185]-a deep contextualized word representation model-to map each token t_i in sentence s_i in contract D to its corresponding contextual embedding v_i with length 1024 [14]. ELMo uses a bi-directional LSTM [208] for language modeling and considers the context of the words when embedding them[15].

3.3.2 Content Selection and Redundancy Reduction

Given the probability distributions over the risk categories, we apply two content selection mechanisms to account for the summarization budget m and minimize the information redundancy. The first mechanism focuses on including the most "risky" sections while the second mechanism focuses on covering diverse privacy factors. Next, we explain these two variations of our model.

Risk-Focused Content Selection: Given a privacy policy contract D with sentences $\{s_1, ...s_n\}$, a summarization budget m, and risk score $p(y_i = 1|s_i, D, \theta)$ predicted for s_i by the risk classifier, the risk-focused selection mechanism assembles a summary by extracting the top m sentences that have the highest risk score.

Coverage-Focused Content Selection: Given a privacy policy contract D with sentences $\{s_1, ...s_n\}$, a summarization budget m, and risk scores $p(y_i = 1|s_i, D, \theta)$, the coverage-focused selection method finds m privacy factors by clustering sentences for which the risk score is larger than a predefined value of α. Next, the riskiest sentence from each privacy factor cluster is selected to be included in the summary. Note that if less than m sentences have a risk score greater than α the summary will have less than m sentences. To find privacy topics of a contract, we apply k-means [110] to sentence representations. Sentence representations are obtained through concatenating the word vectors. Number of clusters is set to $min(m, |r|)$ where $r = \{s_i \mid p(y_i = 1) > \alpha\}$.

---

[14]Model was trained on the One billion word benchmark [38] and was obtained from https://github.com/allenai/allennlp

[15]BERT [56] as the current state-of-the-art for language model pretraining has achieved amazing results in many NLU tasks with minimal fine-tuning. However, our preliminary results of fine-tuning bert did not outperform our results from Elmo word vectors and task-specific architecture explained in Section 3.3.1.

3.4 Dataset Extraction

In this section, we explain the dataset that we compiled from the TOS;DR website and privacy contracts of 151 companies. TOS;DR is a website dedicated to rating and explaining privacy policy of companies in plain English. Members of the website's community classify specific sections of privacy policies into ”bad”, ”good”, ”blocker”, and ”neutral” categories and provide summaries for them. We collected the user agreement contracts of 151 services that were annotated on TOS;DR from the companies' websites. Some companies have several such contracts e.g. privacy policy, terms of service, and cookie policy. In this case, all the contracts were merged into a single document. Next, we compared each sentence of the contract with specific snippets that were annotated on TOS;DR. If the corresponding sentence or a very similar sentence was annotated by the TOS;DR contributors, the same label was used. Otherwise, it was annotated as ”neutral”. The assumption behind our annotation schema is that, if a section was not annotated by the contributors, it most likely does not include a privacy risk and thus, is considered neutral. NLTK was used to segment the contracts into sentences. Jaccard similarity of the vocabulary was used to measure the similarity of the sentences. Two sentences from the same contract were considered similar if the Jaccard similarity of their tokens was more than 50%. We combined the ”bad” and ”blocker” sections to build the ”risky” class. The ”good” and ”neutral” classes were also combined to build the ”non-risky” class. This dataset is highly imbalanced with 61674 non-risky sentences and only 719 risky sentences. To build the ground truth risk-aware summary of each privacy policy we concatenate the plain English summaries of the snippets that have a ”risky” label. The dataset statistics of the 151 privacy policies and their corresponding summaries are presented in Table 3.3. Our dataset is available online [16].

Dataset	**Min**	**Max**	**Median**	**Mean**
Privacy Policies	61	1707	350	411.6
Plain English Summaries	1	53	1	3.5

Table 3.3: The min, max, median, and average number of sentences in 151 privacy contracts and their summaries.

3.5 Experiments

In this section, we discuss our data augmentation mechanism to reduce the data imbalance problem, our hyper parameter choice for designing the risk classifier, and the training details. We discuss our evaluation criteria in Section 3.5.2.

[16] www.github.com/senjed/Summarization-of-Privacy-Policies

	Compression Ratio = 1/64				Compression Ratio = 1/16			
	P	**R**	**Macro-F1**	**Micro-F1**	**P**	**R**	**Macro-F1**	**Micro-F1**
CNN + RF	22.40	28.13	61.94	98.01	9.86	59.74	56.65	93.10
CNN + CF	19.64	24.06	60.26	97.95	12.19	52.65	58.51	94.94

Table 3.4: Precision(P), Recall(R), Macro-F1, and Micro-F1 of the CNN classifier with two different content selection mechanisms risk-focused(RF) and coverage-focused(CF) at two different compression ratios $\frac{1}{16}$ and $\frac{1}{64}$.

3.5.1 Hyperparameters and Training Details

For the CNN model, we use two filter region sizes 3 and 4 each of which has 50 output filters. We use rectified linear unit as the activation function of the convolution layer. The pool size in the max pooling operation is set to 50. We apply dropout with a rate of 20%. We optimize the binary cross-entropy loss using stochastic gradient descent with a learning rate of 0.01. To account for the class imbalance problem, we randomly under-sampled the majority class (non-risky) with a rate of 10%. We also apply *SMOTE* over sampling [37] on the minority class (risky) with rate of 50%. We train our model on this resampled dataset for 20 epochs and weight the loss function inversely proportional to class frequencies in the input data. To set the value of risk threshold α in the content selection module, we used the ROC curve of the validation set of each fold. We set α for each fold to the threshold value that achieves 80% true positive rate.

3.5.2 Evaluation Metrics

In our experiments, we seek to answer the following questions:

- How well does our model identify the risky sentences in the contracts?
- What content selection method leads to more "human-like" summaries?

To answer the first question we report the Macro-F1 and Micro-F1 score of our classifier. To answer the second question, we evaluate the quality of the extracted summaries by our model by computing the average F1-score for ROUGE-1, ROUGE-2, and ROUGE-L [134] metrics (which respectively measure the unigram-overlap, bigram-overlap, and longest common sequence between the reference summary and the summary to be evaluated). ROUGE metrics fail to capture semantic similarity beyond n-grams [238]. Thus, we also report the METEOR score [55] which goes beyond the surface matches and accounts for stems and synonyms while finding the matches.[17] We evaluate our model using 5-fold cross-validation. In each fold, contracts of 96 companies are used for training, 24 contracts are used for validation, and the rest is used for testing. We explain our baselines in Section 3.5.3 and our experimental results in Section 3.6.

[17] We use pyrouge and NLTK python packages for computing ROUGE and METEOR values respectively.

3.5.3 Summarization Baselines

We compare the performance of our domain-aware extractive summarization model with the following unsupervised baselines. Unlike the evaluation setup in [154], we run the models on the entire contract. For methods that require a word limit as the budget, a compression ratio r is multiplied by the average number of tokens in all contracts (10488.7) to compute the word limit. Similarly, the compression ratio of r is multiplied by the average number of sentences in all contracts (413.1) to build a sentence limit.

- **TextRank**: An algorithm introduced in [160] that uses page rank to compute an importance score for each sentence. Sentences with the highest importance score are then extracted to build a summary until a word limit is satisfied.
- **KLSum**: Introduced in [85], KLSum aims to minimize the Kullback-Lieber (KL) divergence between the input document and proposed summary by greedily selecting sentences.
- **Lead-K**: A common baseline in news summarization that extracts the first k sentences of the document until a word limit is reached.
- **Random**: This baseline picks random sentences of the document until a word limit is satisfied. For this baseline, we report the average results over 10 runs.
- **Upper Bound Baseline**: This baseline picks all the sentences in a contract with ground truth label ”risky”. This baseline indicates the performance upper bound of an extractive method on our dataset.

3.6 Results

In this section, we discuss our experiments conducted using 5-fold cross-validation. We shared our training details in Section 3.5.1. As an example, summaries extracted by our model and the baselines from privacy policy of Brainly [18] is displayed in Figures 3.2 and 3.3. It can be seen that both of the summaries generated by our method indicate that third party advertising companies will be able to collect information about use of Brainly. KLSum misses this information and the traditional lead-k heuristic which is very effective for news performs poorly on the contracts. This indicates the advantage of injecting domain-specific knowledge into content selection.

3.6.1 Classification Results

In this section, we evaluate the performance of our model discussed in Section 3.3.1 and study the effect of different content selection mechanism on the risk prediction task. We evaluate our summaries at two compression ratios of $\frac{1}{64}$ and $\frac{1}{16}$. The summarization budget m at each compression ratio r is achieved by multiplying r in the average number of sentences(or words) in the contracts. Thus, at the compression ratio of $\frac{1}{64}$, summaries are restricted to the maximum length of 6

[18] https://Brainly.com

	Compression Ratio = 1/64				Compression Ratio = 1/16			
	ROUGE-1	**ROUGE-2**	**ROUGE-L**	**METEOR**	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**	**METEOR**
CNN + RF	**43.09**	**31.21**	**36.80**	**41.98**	34.0	24.96	24.83	40.03
CNN + CF	40.45	28.69	34.01	41.55	**37.93**	**28.82**	**29.23**	**43.91**
Textrank	28	13.89	22.06	22.4	33.78	22.12	26.85	35.49
KLSum	28.75	13.14	23.53	25.34	24.74	11.36	18.86	26.95
Lead-k	25.57	9.09	20.25	19.54	25.67	11.33	19.77	26.85
Random	24.26	6.45	18.78	18.11	24.43	9.85	18.08	27.01

Table 3.5: ROUGE-1, ROUGE-2, ROUGE-l, and METEOR score of our model (highlighted in light gray) in comparison to the baselines in compression ratios $\frac{1}{64}$ and $\frac{1}{16}$. RF refers to the risk-focused content selection while CF refers to the coverage-focused content selection. The quote text of the risky sections was used to build the reference summaries.

sentences or 164 words. Similarly, at the compression ratio of $\frac{1}{16}$, summaries are limited to the maximum length of 29 sentences or 656 words. We report the precision, recall, Micro-F1, and Macro-F1 of our risk classifier with two different content selection mechanisms namely risk-focused (RF) and coverage-focused (CF) in Table 3.4. As can be seen in the table, the Micro-F1 scores of both content selection methods are quite high. However, the best Macro-F1 value is achieved by the risk-focused approach and is 61.94. The large gap between the two values is due to the high level of class imbalance in our dataset (1 positive sample for every 100 negative samples). At $\frac{1}{64}$ compression ratio, risk-focused performs more than two times better in terms of recall. When the compression ratio is $\frac{1}{16}$, the risk-focused method captures many more risky sections and achieves a recall of 59.74. However, with this increase in recall, the false positive rate also increases. On the other hand, the coverage-focused method is better at preserving the precision at higher budgets (only 7.45 drop in precision with a 28.59 points increase in recall). This observation is caused by extracting sentences with a risk score greater than α in coverage-focused content selection. This naturally puts an upper bound on the false positive rate. We conclude that both mechanisms are moderately successful at identifying the risky sections of contracts. We also conclude that at higher compression ratios, the risk-focused mechanism can be used where recall is more essential while the coverage-focused mechanism can be used when precision is more of interest. In the next section, we examine whether the domain information given by the risk classifier can improve the quality of summaries in comparison to domain-agnostic extractive summarization baselines.

3.6.2 Summarization Results

In this section, we evaluate the quality of the summaries extracted by our model and the baselines. We introduced our evaluation metrics in Section 3.5.2 and our baselines in Section 3.5.3. We compare the summaries against two type of reference summaries. The first type of summary is built by assembling all the sentences that have ground truth ”risky” label. These sentences are derived directly from text of the contract. We will refer to this reference summary as ”quote text” reference. The second type of summary is derived by assembling the plain English summary of the ”risky” sections written by the TOS;DR contributors. The summarization results using the quote text

	Compression Ratio = 1/64				Compression Ratio = 1/16			
	ROUGE-1	**ROUGE-2**	**ROUGE-L**	**METEOR**	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**	**METEOR**
Upper Bound	22.45	13.7	18.27	22.32	22.56	13.95	18.49	23.03
CNN + RF	**13.97**	**6.08**	**9.83**	**16.58**	9.07	3.94	5.53	12.07
CNN + CF	12.39	4.81	8.51	14.93	**10.18**	**4.54**	**6.58**	**13.16**
Textrank	10.94	2.78	7.51	11.2	10.08	3.37	6.37	12.47
KLSum	10.96	2.43	7.34	12.54	8.37	1.92	5.26	11.06
Lead-k	11.21	1.9	7.9	11.04	9.33	2.44	5.96	11.87
Random	11.44	1.87	8.03	12.02	9.13	2.32	5.73	12.45

Table 3.6: Performance of our model (highlighted in light gray) in comparison to the baselines in compression ratios $\frac{1}{64}$ and $\frac{1}{16}$. RF refers to the risk-focused content selection while CF refers to the coverage-focused content selection. The plain English summaries of risky sections was used to build the reference summaries.

summaries is presented in Table 3.5. The summarization results using the plain English reference summaries is presented in Table 3.6.

Extracting the risky content: As it can be seen in Table 3.5, at both compression ratios, both variation of our model outperform the baselines. At compression ratio of $\frac{1}{64}$, the CNN + RF, achieves the best ROUGE and METEOR results with 49.8% improvement in ROUGE-1, 124.6% improvement in ROUGE-2, 56.3% improvement in ROUGE-L, and 65.6% improvement in METEOR in comparison to the best performing domain-agnostic baseline for each metric. At compression ratio of $\frac{1}{16}$ the CNN + CF achieves the best ROUGE results by improving ROUGE-1 by 12.2%, ROUGE-2 by 30.2%, ROUGE-L by 8.8%, and METEOR by 23.7% in comparison the the best performing baseline for each metric. The improvement in METEOR score is found to be statistically significant using Wilcoxon signed ranked test [230] with p-value < 0.01 (Bonferroni corrected [63] to account for multiple testing). Similar to our observation in classification task, we find that the risk-focused content selection achieves more recall and thus, achieves a better METEOR score in comparison to the coverage-focused mechanism. On the other hand, by increasing the summarization budget, the ROUGE values for this method slightly drop. This is because, in most of the contracts, the number of risky sentences is smaller than the budget at ratio of $\frac{1}{16}$ (29 sentences).

Building Human-like summaries: We present our summarization results using the plain English summaries as reference summaries in Table 3.6. At compression ratio of $\frac{1}{64}$, both variations of our model outperform the baselines. Our CNN + RF model, increases the METEOR score by 32.2% over KLSum and 48% over textrank. This improvement is found to be statistically significant (with p-value < 0.01). The CNN + CF outperforms the baselines over all evaluation metrics. However, the improvement is not statistically significant. At compression ratio of $\frac{1}{16}$, CNN + RF outperforms all domain-agnostic baselines. This improvement however, is not statistically significant. At this compression ratio, CNN + RF achieves comparable result with textrank. We conclude from our experiments that our domain-aware extractive model does moderately better than the baselines at lower compression ratios, however, due to high level of abstraction in plain English summaries of TOS;DR [153], a fully-extractive approach cannot mimic the human-like qualities in the plain English summaries. This can also be seen by looking at the performance of the upper bound baseline.

3.7 Conclusion

In this chapter, we first presented the results of a user study on users comprehension of the policies and their preferences. Our findings suggested that users are the most accurate with the color-coded policy. To further extend the previous work on color-coding policies based on risk factor and also addressing the information overload problem in this domain, in this chapter we proposed a domain-aware extractive model for summarizing the privacy contracts.

Our model, employs a convolutional neural network to identify risky sections of the contracts. We build summaries by using a risk-focused and a coverage-focused content selection mechanism. Our approach enables users to select the content to be summarized within a controllable length while relying on substantially less training data in comparison to the existing supervised summarization methods. Our two different content selection mechanisms enable users to build budgeted summaries of contracts based on their preference of coverage vs risk. In spite of the moderate success in classification of our realistically imbalanced dataset, we observed a noticeable improvement in ROUGE and METEOR metrics in comparison to domain agnostic baselines. We also release a dataset of 151 privacy policies for which each segment is annotated with a risk class to facilitate future research in this domain.

Plain English Summary: The Privacy Policy states, ”We and our third party partners may also use cookies and tracking technologies for advertising purposes.”. In the Privacy Policy, it states that, ”Although we do our best to honor the privacy preferences of our users, we are unable to respond to Do Not Track signals set by your browser at this time.” The Privacy Poilicy says Brainly can track usage information and personal information ”through a variety of tracking technologies, including cookies, web beacons, Locally Stored Objects (LSOs such as Flash or HTML5), log files, and similar technology (collectively, “tracking technologies”).” If Brainly aims to ”preserve all content posted on the site,” then we can conclude that such personal data is still necessary for the purpose of the site. There are places on the site where answers without usernames or profile pictures are visible. The Cookie Policy states, ”Service operator [sic] informs that restricting the use of cookies may affect some of the functionalities available on the Website.” For users not in europe, brainly reserves the right, in its sole discretion, to immediately modify, suspend or terminate your account, the brainly services, your brainly subscription, and/or any products, services, functionality, information, content or other material. *<truncated>*

CNN + RF: We participate in interest-based advertising and use third party advertising companies to serve you targeted advertisements based on your online browsing history and your interests. We permit third party online advertising networks, social media companies and other third party services, to collect, information about your use of our service over time so that they may play or display ads on our service, on other websites, apps or services you may use, and on other devices you may use. We may share a common account identifier (such as an email address or user id) or hashed data with our third party advertising partners to help identify you across devices. Brainly reserves the right to moderate the Brainly services and to remove, screen, or edit your content from the Brainly services at our sole discretion, at any time, and for any reason or for no reason, with no notice to you. Brainly reserves the right, in its sole discretion, to immediately modify, suspend or terminate your account, the Brainly services, your Brainly subscription, and/or any products, services, functionality, information, content or other materials available on, through or in connection with the Brainly services and/or your Brainly subscription, including, but not limited to, the mobile software, and/or your access to some or all of them without cause and without notice. In the event that Brainly suspends or terminates your account, the Brainly services or your Brainly subscription, you acknowledge and agree that you shall receive no refund or exchange for any unused time on a Brainly subscription or any subscription fees or anything else.

CNN + CF: We participate in interest-based advertising and use third party advertising companies to serve you targeted advertisements based on your online browsing history and your interests. We permit third party online advertising networks, social media companies and other third party services, to collect, information about your use of our service over time so that they may play or display ads on our service, on other websites, apps or services you may use, and on other devices you may use. We may share a common account identifier (such as an email address or user id) or hashed data with our third party advertising partners to help identify you across devices. To the fullest extent permitted by applicable law, no arbitration or claim under these terms shall be joined to any other arbitration or claim, including any arbitration or claim involving any other current or former user of the Brainly services or a Brainly subscription, and no class arbitration proceedings shall be permitted. We may modify or update this privacy policy from time to time to reflect the changes in our business and practices, and so you should review this page periodically. If you object to any changes, you may close your account. Continuing to use our service after we publish changes to this privacy policy means that you are consenting to the changes.

Figure 3.2: The summaries extracted by our model (CNN + RF and CNN + CF) from the privacy policy and cookie policy of Brainly at compression ratio of $\frac{1}{64}$.

Lead-K: Welcome to Brainly!. Brainly operates a group of social learning networks for students and educators. Brainly inspires students to share and explore knowledge in a collaborative community and engage in peer-to-peer educational assistance, which is made available on www.Brainly.com and any www.Brainly.com sub-domains(the "website") as well as the Brainly.com mobile application (the "app") (the "website" and the "app" are collectively the "Brainly services". We have two sets of terms and conditions: part(a) sets out the terms that apply to our users unless you are based in Europe and part (b) sets out the terms that apply to our users in Europe. It is important that you read and understand the terms that apply to you when you use the Brainly services before using the Brainly services. Part (a): terms and conditions applicable to users unless you are based in Europe. This part and the documents referred to within it set out the terms and conditions that apply to your use of Brainly services if you access Brainly services from within the united states or other countries except Europe. The Cookie Policy states, "Service oparator [sic] informs that restricting the use of cookies may affect some of the functionalities available on the Website."

KLSum: Brainly reserves the right, in its sole discretion, to immediately modify, suspend or terminate your account, the Brainly services, your Brainly subscription, and/or any products, services, functionality, information, content or other materials available on, through or in connection with the Brainly services and/or your Brainly subscription, including, but not limited to, the mobile software, and/or your access to some or all of them without cause and without notice. Brainly makes no warranty that the Brainly services and/or any products, services, functionality, information, content or other materials available on, through or in connection with the Brainly services or your Brainly subscription, including, but not limited to, the mobile software, will meet your requirements, or that the Brainly services or Brainly subscriptions will operate uninterrupted or in a timely, secure, or error-free manner, or as to the accuracy or completeness of any information or content accessible from or provided in connection with the Brainly services or Brainly subscriptions, regardless of whether any information or content is marked as "verified". You must not: use Brainly services other than for its intended purpose as set out in the terms of use; *<truncated for presentation purpose. Rest of the summary includes examples of misuse of the Brainly services.>*

TextRank: You must not: use Brainly services other than for its intended purpose as set out in the terms of use; copy any portion of Brainly services; give or sell or otherwise make available any portion of Brainly services to anybody else; change Brainly services in any way; interfere with any security related features of Brainly services or features that prevent or restrict use or copying of the content accessible via Brainly services; give any information or permit another person to use Brainly services under your name or on your behalf; fake your identity or give the impression they are linked to us or to Brainly, if this is not the case; use Brainly services other than for its intended purpose as set out in the terms of use; use Brainly services if we have suspended your access to it, or have otherwise banned you from using it; modify, interfere, intercept, disrupt or hack Brainly services or collect any data from Brainly services other than in accordance with the terms of use; misuse Brainly services by knowingly introducing viruses, trojans, worms, logic bombs or other material which would harm the Brainly services or the equipment of any user of Brainly services.

Figure 3.3: The summaries extracted by the baselines from the privacy policy and cookie policy of Brainly at compression ratio of $\frac{1}{64}$.

Chapter 4: Automated Privacy Policy Question Answering Assistant

Existing work on making privacy policies accessible has explored new presentation forms such as color-coding based on the risk factors or summarization to assist users with conscious agreement. To facilitate a more personalized interaction with the policies, in this chapter, we propose an automated question answering assistant that extracts relevant segments from the policy in response to an input user query. This is a challenging task because users articulate their privacy-related questions in a very different language than the legal language of the policy, making it difficult for the system to understand their inquiry. Moreover, existing annotated data in this domain are limited. We address these problems by paraphrasing to bring the style and language of the user questions closer to the language of privacy policies. We use familiar techniques such as back-translation and lexical substitution and examine to what extent these previously unexplored techniques in the legal domain are beneficial for the privacy policy question answering task. Following query expansion, we use a content scoring module that uses the existing in-domain data to find relevant information in the policy. Our pipeline can find an answer for 87.7% of the user queries in the privacyQA dataset. Our analysis shows that the unanswered questions are mostly ambiguous, subjective, or too specific.

4.1 Introduction and Related Work

As mentioned in the previous chapter, online users often do not read or understand privacy policies due to the length and complexity of these unilateral contracts [51]. This problem can be addressed by utilizing a presentation form that does not result in cognitive fatigue [57, 234] and satisfies the information need of users. To assist users with understanding the content of privacy policies and conscious agreement, previous computational work on privacy policies has explored using information extraction and natural language processing to create better presentation forms [68]. For example, PrivacyGuide [223] and PrivacyCheck [240] present an at-a-glance description of a privacy policy by defining a set of privacy topics and assigning a risk level to each topic. Harkous et al. [87] and Nejad et al. [170] used information extraction and text classification to create a structured and color-coded view of the risk factors in the privacy policy. In our previous work we explored incorporating the risky data practices in the privacy policies in form of a natural language summary [113]. While great progress has been made to create more user-friendly presentation forms for the policies, users often only care about a subset of these issues or have a personal view of what is considered risky. Instead of presenting an overview or summary of privacy policies, an alternative approach is to allow them to ask questions about the issues that they care about and present an answer extracted from the content of the policies [194]. This facilitates a more personal approach to privacy and enables users to review only the sections of the policy that they are most concerned about.

This task is related to guided and controllable text summarization [125, 52, 115, 70, 207] as well as reading comprehension [90]. However, a few application-imposed constraints make this task more challenging than the traditional evaluation setup of reading comprehension systems. First, users tend to pose questions to the privacy policy question-answering system that are not-relevant, out-of-scope (*'how many data breaches did you have in the past?'*), subjective (e.g. *'how do I know this app is legit?'*), or too specific to answer using the privacy policy (e.g. *'does it have access to financial apps I use?'*) [194]. Moreover, even answerable user questions can be ill-phrased or have a very different style and language in comparison to the legal language used in privacy policies [195], making it difficult for the automated assistant to identify the user's intent and find the relevant information in the document. This issue of domain shift is exacerbated due to the difficulty of annotating data for this domain. Because the existing datasets for this task are fairly small [4], the problems cannot be solved by simply training a supervised model.

In this chapter, we take another step toward making privacy policies more accessible by creating an automated question-answering assistant. We focus on addressing the domain-mismatch problem and aim to bring the style, language, and specificity of the user's question closer to the language of privacy policies. To do so, we use familiar techniques such as lexical substitution and back-translation that are not previously explored in the legal domain. Next, we compute a relevance and informativeness score for each segment of the policy using a transformer-based language representation model fine-tuned on in-domain data. Finally, we return the top relevant segments to the user. To summarize we make the following contributions:

- We propose a pipeline for automated question answering of privacy policies
- We explore the adaptation of existing techniques for query expansion to bring the language of user queries closer to the legal language of the privacy policies.
- We show that using the existing in-domain data and techniques such as back-translation and lexical substitution we can find an answer for 87.7% of the user queries in the PrivacyQA dataset.
- We discuss future directions for further improving the automated privacy policy questions answering models and making privacy policies more accessible to lay users.

We discuss our proposed hybrid question-answering pipeline in §4.2. We introduce the datasets we use for training and testing different modules of our model in §4.3. Finally, we present our experiments and results in §4.5.

4.2 Proposed Pipeline

Our query-guided extractive summarization pipeline includes three main components. Given a privacy policy document and a user query, the first component - the query expansion module, processes the user query and generates a set of paraphrases that have a more similar language, style,

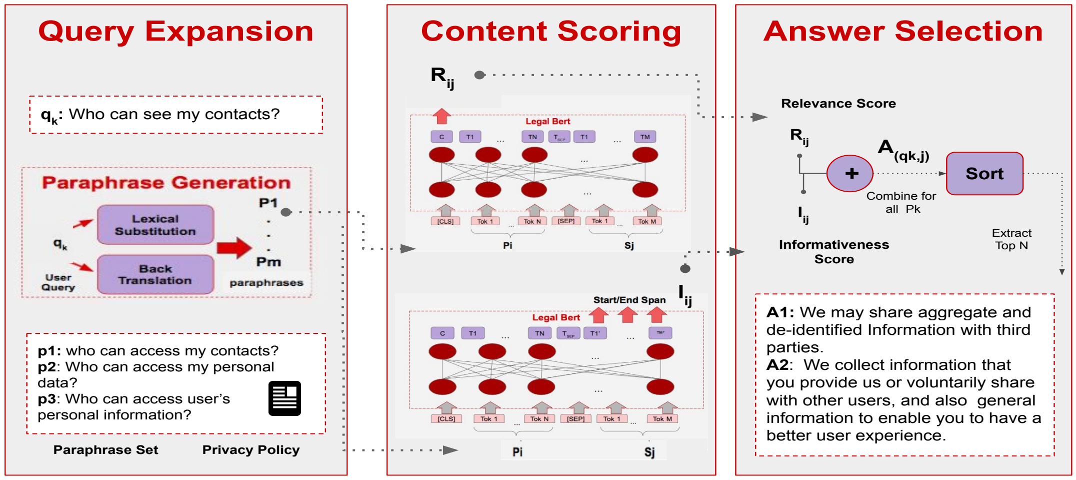


Figure 4.1: Overview of the proposed pipeline

and specificity to the privacy policy. Next, given the query and the paraphrase set, the content scoring module computes two scores—relevance and informativeness—for each segment of the policy. Lastly, the two scores are combined for the expanded query set to obtain the final answerability score for each segment. Segments are then ranked based on the answerability score and top-ranked items are shown to the user. The overview of the pipeline as well as an input-output example is shown in Figure 4.1. Next, we discuss each component of our pipeline in more detail.

4.2.1 Query Expansion

Question-answering systems are very sensitive to many different ways the same information need can be articulated [58]. As a result, small variations in semantically similar queries can yield different answers. This is especially a challenge in building a question-answering assistant for privacy policies. Often, users are not very good at articulating their privacy-related inquiries and use a style and language that is very different from the legal language used in the privacy policies [194].

Query expansion by paraphrasing has been used in the past to improve the performance of the QA-based information retrieval [253, 199, 7]. We employ several methods from the literature, testing their applicability to this domain and in particular to the issues caused by mismatch between external training resources, user queries and the privacy policies themselves. To increase the diversity and coverage of the generated paraphrases, we employ methods based on lexical substitution [155, 103] and neural machine translation (NMT) [150, 220, 16, 15]. Note that the paraphrase generation module is independent of the neural-based content scoring module and thus, any method can be used to generate paraphrases. Below, we discuss the three methods used for generating query paraphrases.

4.2.1.1 Lexical Substitution

Lexical substitution can be done by simply replacing a word with an appropriate synonym or paraphrase in a way that the meaning is not changed. For example, the sentence *'what information is collected about me?'* can be written as *'what information is collected about the user?'*. For generating paraphrases we employ two lexical substitution methods: (i) replacement with similar words based on Word2Vec representations [126] and (ii) a collection of hand-crafted lexical replacements rules aimed to bring the style and language of user queries closer to the legal language in the privacy policies.[19]

Word2Vec: We fine-tune pre-trained Word2Vec representations on a corpus of 150 privacy policies. This dataset was introduced in Section 3.4. Training details can be found in Section 4.4. To create paraphrases, we substitute nouns and verbs in user queries with the top 5 most similar words in the embedding space that have the same part of speech.

Lexical Substitution Rules		
my → user's	ads → advertisement	search → browse
i → user	others → third parties	view → access
keep → retain	3rd → third	looking up → search
sell → share	upload → share	businesses → third parties
information → data	images → presonal data	keep a record → collect
sold → shared	messages → personal data	report → share
anyone → third parties	mine → user's	private → personal
stored → retained	selling → sharing	kept → retained
store → collect	records → collects	sent → shared
save → collect	monitoring → accessing	send → share
me → the user	monitored → accessed	see → access
record → collect	tell → notify	phone → device
recorded → collected	log → collect	photos → personal data
keep record of → collect	invade → access	monitor → access
held → retained	warning → notification	track → collect
safe-guarded → protected	safegaurd → protect	saved → collected
read → access	info → information	know → has access
3rd → third	tracked → collected	secured → protected
upload → share	mine → user's	records → collects
images → presonal data	selling → sharing	monitoring → accessing
messages → personal data	monitored → accessed	tell → notify

Table 4.1: Lexical substitution rules used to bring the language of user queries closer to the legal language of privacy policy.

[19] We also tried using WordNet [162] for lexical substitution. However, our preliminary experiments suggest that majority of paraphrases generated using this method are not meaningful and thus a more rigorous filtering mechanism should be used to identify useful paraphrases. Thus, we have decided not to use WordNet.

Lexical replacement rules: to bring the language and style of user queries closer to the language of privacy policies, we manually create a collection of 50+ lexical substitution rules. For example, we replace the word ”my” with ”user’s” and ”phone” with ”device”. We test two variations of this approach: (i) single-replacement, in which we only apply a single replacement rule to generate a paraphrase and (ii) all-replacement, in which we apply all possible lexical substitution rules to generate a paraphrase. These replacement rules are shared in Table 4.1

4.2.1.2 NMT-based Paraphrase Generation

One of the well-known approaches for paraphrase generation is bilingual pivoting [12, 102, 103, 58]. In this approach, a bilingual parallel corpus is used for learning paraphrases based on techniques from paraphrase-based statistical machine translation [122]. Intuitively, two sentences in a source language that translates to the same sentence in a target language can be assumed to have the same meaning. Mallinson et al. [150] show how the bilingual pivoting method can be ported into NMT and present a paraphrasing method purely based on neural networks. In our work, we use German as our pivot language following Mallinson et al. [150], who suggest that it outperforms other languages in several paraphrasing experiments. We employ a simple back translation method to automatically create paraphrases for user queries using Google Translate[20] which is a mature and publicly available online service to translate user queries from English to German and back from German to English.

4.2.2 Content Scoring

Given the segment set $S = \{s_1, s_2, ...s_n\}$ of the privacy policy, a user query q_k and paraphrase set $P_k = \{p_1, p_2, ..., p_m\}$ obtained for q_k, we aim to extract the most relevant segments $s_j \in S$ of the privacy policy that fully or partially answer q_k. To do so, for each pair of paraphrase-segment pair (p_i, s_j), we compute two scores that we call the relevancy score R_{ij} and informativeness score I_{ij}. Both scores are computed using BERT [56], but we employ different problem formulations, discussed below. We combine these two scores to get the final answerability score A_{ij} for paraphrase-segment pair (p_i, s_j). Finally we compute the average answerability score of the paraphrase set P_k to get answerability score of query q_k. We represent the answerability score of segment j for query k with $A_{(q_k,j)}$. Next, we discuss how these scores are computed.

Relevance Score: To compute the relevance score R_{ij} for a paraphrase-segment pair (p_i, s_j), we formulate this as a sentence-pair classification task. In this task, given a question p_i and segment $s_j \in S$, the goal is to predict whether s_j is relevant to p_i. To compute the relevance score, we rely on a transformer-based language representation model [56] pretrained on legal contracts called *legal-bert* [36].[21]. We fine-tune this model for sentence pair classification task on the train set of the PrivacyQA dataset proposed in [194] for 3 epochs. PrivacyQA is a corpus of privacy policy segments annotated as *”relevant”* or *”irrelevant”* for a set of user queries. We introduce this dataset in §4.3. During fine-tuning, we pass the question-segment pairs separated with the special token $[SEP]$ with question and segment using different segment embeddings. We also add a special

[20]https://translate.google.com

[21]Pretrained model is obtained from https://huggingface.co/nlpaueb/legal-bert-base-uncased

token $[CLS]$ in the beginning and a token $[SEP]$ at the end of the sequence. We use weighted binary cross-entropy as our loss function and update the encoder weights during the fine-tuning. The final hidden vector for the first input token $[CLS]$ is fed to the output layer for the relevance prediction task. We use the fine-tuned model to get the posterior probability of relevancy for each paraphrase-segment sequence. More details on our fine-tuning process can be found in Section 4.4.

Informativeness Score: Even if a segment of the policy is relevant to a question, it might not fully answer it. To account for this, we also train a span-detection QA system similar to those used for SQUAD question answering [192]. In preliminary experiments, we do not find that this system always extracts spans which are legible enough on their own for presentation to the user; this is partly due to the complex, contextually-sensitive language used in the contracts. However, we do find that the system's ability to find a promising span provides another indication that the text segment contains a potential answer. Thus, for each question-segment pair (p_i, s_j) we compute an informativeness score which measures how informative is s_j in answering p_i. To compute this score, we fine-tune the legal-bert [36] for question-answering task on the train set of the PolicyQA dataset [4]. This dataset contains reading comprehension style question and answer pairs from a corpus of privacy policies and will be introduced in §4.3. More details on our fine-tuning process can be found in Section 4.4. We refer to the legal-bert fine-tuned for question-answering as the *"answer-detector"* module. During fine-tuning, we feed a query and segment of the policy as a packed sequence separated by the special token $[SEP]$ with the question and the segment using different segment embeddings. In addition, a start vector S and an end vector E are introduced during the fine-tuning process. For each token in the sequence two probabilities are computed; the probability of word k being the start of the answer span and the probability of word k being the end of the answer span. To compute the start-of-answer-span probability we compute the dot-product of the token vector T_k and the start vector S followed by a softmax over all tokens in the segment. A similar formula is used to compute the end-of-answer-span probability for each token. The training objective during fine-tuning is to maximize the sum of the log-likelihoods of the correct start and end positions. The informativenss score of the span from position a to position b is defined as $S \cdot T_a + E \cdot T_b$ where $a \leq b$. We represent the informativeness score of the segment s_j with respect to the paraphrase p_i with I_{ij} and compute it by taking the maximum score of the spans within the segment:

$$I_{ij} = max(S \cdot T_a + E \cdot T_b)$$

Where $0 \leq a \leq b \leq len(s_i)$.

4.2.3 Answer Ranking and Selection

Finally, to compute the answerability score A_{ij} for each paraphrase-segment pair (p_i, s_j), we simply sum up the relevance score R_{ij} and informativeness score I_{ij}:[22]

$$A_{ij} = R_{ij} + I_{ij}$$

[22]We also tried training a regression model using R_{ij} and I_{ij} as inputs and the relevance labels from PrivacyQA as the target variable. However, reusing PrivacyQA labels seems to result-in over-fitting. Thus, we decided to combine the scores by simply summing them up.

A_{ij} is computed for all paraphrase-segment pairs (p_i, s_j). Finally the answerability score of the segment s_j for query q_k is computed by averaging the answerability score of the paraphrase set:

$$A_{(q_k,j)} = \frac{\Sigma_{i=1}^{m} A_{ij}}{m}$$

Where m represents the number of the paraphrases generated for query q_k. Finally, we rank segments $s_j \in S$ based on their answerability score $A_{(q_k,j)}$ with respect to the input user query q_k. The top ranked segments are then shown to the user. In our experiments in §4.5 we show the results of retrieving the top 5 and top 10 ranked segments. Our question-answering pipeline is shown in Figure 4.1. In the next section, we introduce the datasets used for fine-tuning and testing our proposed pipeline.

4.3 Datasets

We rely on three publicly available data sets for training and testing different modules in our pipeline. As mentioned earlier, we train the word2vec model used for lexical substitution on the set of 150 privacy policies collected in our previous work [113]. More details about this dataset is shared in Section 3.4. In addition, we employ two datasets called PrivacyQA [194] and PolicyQA [4] for fine-tuning the legal bert model for sentence pair classification and question-answering tasks respectively. PrivacyQA is a sentence-selection style question-answering dataset where each question is answered with a list of sentences. On the other hand, PolicyQA is a reading-comprehension style question-answering dataset in which a question is answered with a sequence of words. Next, we introduce these datasets in more details.

PrivacyQA: Ravichander et al. [194] asked each crowd worker in their study to formulate 5 privacy questions about privacy policies of a set of 35 mobile applications. The crowd workers were only exposed to the public information about each company. In addition, they were not required to read the privacy policies to formulate their questions. Thus, this dataset presents a more realistic view of what type of questions are likely to be posed to an automotive privacy policy question-answering assistant. Given the questions formulated by Mechanical Turkers, four experts with legal training annotated paragraphs of the privacy policy as *relevant* or *irrelevant* considering each query. We consider a segment of the privacy policy as relevant if at least one of the annotators marked it as relevant. We use the train set for fine-tuning the legal bert model for the sentence pair classification task and computing the relevance score. We use the test of the PrivacyQA dataset which contains 400 user queries to evaluate our proposed pipeline. We share our experimental results in §4.5. Relevant statistics of this datset is shared in Table 4.2.

PolicyQA: This dataset is curated by Ahmad et al. [4] and contains 25,017 reading-comprehension style question and answer-span pairs extracted from a corpus of 115 privacy policies. The train portion of this dataset contains 693 human-written questions with an average answer length of 13.3 words. To curate this dataset, two domain experts used the triple annotations {Practice, Attribute, Value} from the OPP-115 dataset [231] to come up with the questions. For instance, given the triple annotation {First Party Collection/Use, Personal Information Type, Contact} and the corresponding answer span "name, address, telephone number, email address" the annotators formulated questions such as, *"What type of contact information does the company collect?"* and *"Will you use my*

	#Questions	#Policies	#out-of-scope questions	Avg. passages per question	Avg. relevant passage per question
Train	1350	27	425	137.1	5.2
Test	400	8	34	155.3	15.5

Table 4.2: Statistics of PrivacyQA Dataset; where # denotes number of questions, policies, and out-of-scope questions. Out-of-score questions refer to questions for which no segment in the policy is annotated as relevant. We also report the average number of annotated passages/segments and the average number of relevant segments for each question.

contact information?". Note that during the annotation process, the domain experts were asked to formulate questions given the content of the privacy policy. Therefore, PolicyQA questions are less diverse than PrivacyQA and do not fully reflect a real-world user-interaction with a privacy policy question-answering assistant. Thus, we only use this dataset for fine-tuning the legal bert model for question-answering task and computing the informativeness score. We do not use this dataset for evaluation. Relevant statistics of this datset is shared in Table 4.3. Next, we will share details of tuning word embeddings for lexical substitution and details of fine-tuning Bert for relevance score and informativess score computation.

	Questions	#Policies	#Q&A pairs	Avg. question length	Avg. passage length	Avg. answer length
Train	693	75	17,056	11.2	106.0	13.3
Valid	568	20	3,809	11.2	96.6	12.8
Test	600	20	4,152	11.2	119.1	14.1

Table 4.3: Statistics of PolicyQA Dataset; where # denotes number of questions, policies, and Q&A pairs. We also report the average number of words in passages/segments, questions, and answer spans.

4.4 Hyperparameters and Training Details

In this section, we report our choice of hyper-parameters and training details of different modules in our pipeline.

Training Word2Vec for Query Expansion: We use the Gensim Python library [196][23] for training word vectors. We initialize the word vectors using the pre-trained representations [24]. These word vectors are 300 dimensional and are pre-trained on a corpus of Google News dataset (about 100 billion words). We continue training the word vectors on a corpus of 150 privacy polices discussed

[23]https://github.com/RaRe-Technologies/gensim

[24]Pre-trained word vectors are obtained from https://code.google.com/archive/p/word2vec/

in §3.4 for 5 epochs. We tokenize the privacy policy text using NLTK tokenizer [20][25] and remove punctuation and convert all strings to lower case before feeding the data to the Word2Vec model. During training we ignore all the words that occur less than 5 times in the corpus. We use context window size of 5, and use continuous bag of words training algorithm with non-zero, negative sampling for training.

Fine-tuning Legal-Bert for Relevancy Prediction: We obtained the pre-trained model named *bert-base-uncased-contracts* along with the respective tokenizer from Hugging Face [26]. This model is pre-trained on more than 76k US contracts. We fine-tune this model for sentence pair classification task on PrivacyQA dataset for 5 epochs with learning rate of $1e^{-5}$. We use weighted binary cross entropy as our loss function. During tuning, We use batch size of 32 and maximum input length of 70 tokens (95 percentile of the question + segment length in PrivacyQA dataset). We update both weights of the encoder as well as the classification layer.

Fine-tuning Legal-Bert for Informativeness Prediction: We start with the same pre-trained model for informativeness prediction. We fine-tune this model for answer span-detection task on PolicyQA dataset for 5 epochs. We set the learning rate to $1e^{-5}$. and the batch size to 32. The training objective is the sum of the log-probabilities of the correct start and end positions.

4.5 Experiments and Results

In this section, we present our experimental results. As stated earlier, given a query and a privacy policy, the first module of our framework- the query expansion module- brings the style and language of the user-queries closer to the language of the privacy policy by paraphrasing. Next, given the paraphrases for the input query, the content scoring and answer selection modules retrieve the most relevant snippets of the privacy policy.

In our experiments, we aim to answer the following questions:

- Does the query expansion module generate paraphrases that have a closer language than the input query to the privacy policy?
- If so, what proportion of the generated paraphrases are more answerable than the input user query?
- Does the proposed pipeline succeed in retrieving the relevant sections of the privacy policy in answer to user queries?
- Which modules in our pipeline are essential for finding relevant answers to user queries?

Our experiments presented in §4.5.1 answer question one and two. Experiments in §4.5.2 answer question three and four. For all our fine-tuning experiments we use a single NVIDIA V100 GPU with 32GB of memory. Please find more details on our training setup and hyperparameter choice in Section 4.4. For our evaluation, we rely on the test set of the PrivacyQA dataset as it

[25] https://github.com/nltk/nltk

[26] https://huggingface.co/nlpaueb/legal-bert-base-uncased

Method	Rule-based (one)	Rule-based (all)	Back-translation	Word2Vec	All
Average #paraphrases	1.4	0.4	0.9	2.9	5.7
%Retrieved relevant segments	34.5	19.8	25.4	54.0	67.5
%Answerable paraphrases	24.2	31.3	25.4	28.9	27.3

Table 4.4: The average number of paraphrases generated by each method. The percentage of generated answerable paraphrases for non-answerable queries and the percentage of relevant segments that were answerable using at least one of the paraphrases by each method.

presents a more realistic user interaction with a privacy policy assistant. This dataset is introduced in §4.3.

4.5.1 Query Expansion Results:

To answer our first and second questions regarding the quality and answerability of the generated paraphrases, we use lexical substitution methods and back-translation for expanding user queries in the test set of the PrivacyQA dataset. These approaches are introduced in §4.2.1. The average number of paraphrases generated by each approach is presented in Table 4.4. On average, using these methods, we can generate 5.7 paraphrases for each query. The two variations of the rule-based approach, the single-replacement and all-replacement generate 1.4 and 0.4 paraphrases on average. Note that for some queries only one substitution rule is applicable and thus, the all-replacement variation does not generate any new paraphrases. The back-translation method creates 0.9 paraphrases on average; using this method may not always generate novel text.[27] Word2Vec generates more paraphrases than other methods, generating 2.9 paraphrases on average.

To measure the language similarity between paraphrase p_i and segment s_j, we conduct the following experiment. We hypothesize that the answer-detector model introduced in §4.2.3, can successfully detect the answer span within the relevant segment of the policy if the query has a similar language and style to the privacy policy text. Note that in this problem *Recall* is more important than *Precision*. Meaning that being able to find a relevant answer from the policy is more crucial than falsely retrieving a few irrelevant sentences. Our experimental design reflects this domain-imposed requirement. We pass the paraphrase, segment pair $< p_i, s_j >$ that are annotated as ”relevant”, to answer-detector model and save the extracted answer span [28]. In cases that the paraphrase p_i and segment s_j do not have a similar language the model typically returns no answer.[29] In our experiments, we observe that for 342 of initial user queries and relevant segment of the policy (around 5.5% of all pairs), the answer-detector model can't find the answer span. We interpret this as user queries having a different style and language from the policy text. To answer our first question, we measure the percentage of cases for which the expansion method generated an answerable paraphrase. This is shown in Table 4.4 as the *percentage retrieved relevant segments*. The rule-based approach (one-replacement) and Word2Vec are able to generate at least one answerable paraphrase for 34.5% and 54% of the previously non-answerable cases. Note that these two methods on average

[27] In this work we only use the NMT architecture used by Google translate and German language. Using more architectures or more target languages can expand the pool of generated paraphrases.

[28] We exclude 34 queries for which there was no relevant information in the policy (out-of-scope questions).

[29] This includes an empty sequence or special token $[CLS]$

generate more paraphrases for each query while back-translation cannot change the language and style of the user query in some cases. We also observe that 67.5% of all the non-answerable cases could be answered by at least one of the query expansion approaches. Thus, for better recall, we include all the expansion methods in our pipeline.

To measure the quality of generated paraphrases, we report the percentage of all generated paraphrases by each method that were answerable in Table 4.4. Note that different expansion methods generate different number of paraphrases. As presented in the Table, 27.3% of all the paraphrases were answerable. The all-replacement variation of the rule-based method and Word2Vec generate better quality paraphrases in comparison to back-translation (31.3 and 28.9 in comparison to 25.4). We conjecture that this is due to domain mismatch. The NMT architecture generates high-quality paraphrases, but it is trained on out-of-domain data and therefore has no bias to restate the query in a way that makes it match the privacy policies better. On the other hand, the domain-guided rule-based model and word vectors may be less fluent in paraphrasing, but use in-domain data, which allows them to generate better matches.

4.5.2 Question-Answering Results:

To answer our third and forth question, we evaluate the performance of our pipeline in privacy policy question answering task. Essentially, following the query-expansion, we use the content scoring module to generate the relevance and informativeness scores for each paraphrase-segment pair. This process is discussed in §4.2.2. Finally, the answer ranking module combines these scores for the entire paraphrase set and ranks the segments based on their final answerability score.

We evaluate the ability of our proposed pipeline in retrieving the relevant segments of the policy within the top k ranked items. We compare the performance of our model with a simple lexical matching method called the *Word Count baseline*[237]. The Word Count baseline retrieves the top k sentences that have the highest word overlap with the input query. For evaluation, we rely on metrics used for the evaluation of IR systems [151]. We report the average precision at k (P@K), average R-precision (RP), and the mean reciprocal rank (MRR) of the retrieved relevant segments over 3 runs in Table 4.5. P@k indicates the fraction of the relevant segments in the top k ranked items. To measure RP, the number of ground truth relevant segments is used as the cut-off point k for each query. RP measures the fraction of relevant items within the retrieved items. For each query RP is equal to recall at the kth position. We report the average value across queries in the test set. Mean reciprocal rank indicates the multiplicative inverse of the position of the first relevant segment in the resulted ranking. A perfect ranking system achieves a MRR of 1 by always ranking a relevant segment in the first position. We also report the percentage of queries in our test set for which at least one relevant snippet was listed at the top k retrieved items. This is shown as F@k in the table.

We observe that using our proposed model we are able to retrieve a relevant answer for 80% of the queries within the top 5 results and 87.7% of the queries within the top 10 results.[30] We also observe that on average, 39.1% of the top 5 ranked passages are actually relevant. Increasing the output budget to include top 10 retrieved passages decreases the precision to 32. The MRR of the full pipeline is 0.69 indicating that on average the first relevant item appears in the second or first

[30]For these experiments we only use the queries that had at least one relevant answer segment in the policy.

Model	F@5	F@10	P@5	P@10	RP	MRR
R + I + Q (full pipeline)	**80.0**	**87.7**	39.1	32.0	32.5	0.60
R + I	78.9	87.1	39.0	32.2	32.9	0.60
R	78.0	86.8	**40.6**	**33.6**	**34.5**	**0.62**
Word Count	54.7	63.0	20.9	16.7	15.5	0.37

Table 4.5: The performance of different variations of our model and the baselines in the policy question answering task. R represents the legal-Bert model fine-tuned on PrivacyQA for relevance score prediction. I shows the legal-Bert model fine-tuned on PolicyQA for informativeness prediction. Q represents the query expansion module. F@K represent percentage of queries for which at least one relevant segment was found within the top k ranked items. P@K represent precision at K, and RP presents R-precision. MRR is the mean reciprocal rank. Bold font indicates the best result for each performance metric.

position in the ranking. Overall our model easily beats the Word Count baseline by a large margin (+25.3 in F@5, +18.2 in P@5, +17 in RP).

To evaluate the contribution of each module in our pipeline we conduct an ablation study. To test the effect of the query-expansion Q, we use the pipeline without this module for finding answers to user queries. We observe that the percentage of queries answered within the top 5 and top 10 ranked items slightly decreases (-1.1% , -0.6% respectively). We conclude that the query expansion module slightly boosts the performance of the model. However, for most queries in the test set the model is already able to find at least one relevant passage without paraphrasing the queries.

In our next experiment, in addition to the query-expansion module Q, we also remove the answer-detector module I from our pipeline. In this version of our model, passages are only ranked based on the relevance score R. This variation is the closest to the model discussed in [195].[31] Note that we use legal Bert which is pretrianed on contracts whereas Ravichander et al use Bert pretrainaed on books and Wikipedia. We suspect that our choice of using in-domain pretrianing data for Bert creates a stronger baseline than their model. We notice that this baseline achieves a lower F@5 and F@10 that both variations of our pipeline. However, the P@K, RP, and MRR are slightly higher. We conclude that both the query expansion and answer-detector components are effective in the ability of our pipeline in finding ”at least one” relevant answer to user queries. However, the improvement is modest. Using familiar techniques for query expansion enables our pipeline to find a relevant answer for several questions rendered as unanswerable by Ravichander et al. [195]'s model. For example, *”Can other people see your real name?”*, *”Does it need to use the micro-phone at all?”*, and *”Do you use my data to do medical research?”* . Find more examples of such queries in Table 4.6.

Several user queries still cannot be answered by our pipeline nor by the Ravichander et al. [195]'s model. The majority of these questions are either vague, subjective, too specific, too broad, or ill-formed. Examples of these questions are shared in Table 4.6. We conjecture that while using

[31] Our experimental setup is different from Ravichander et al. [195] as they did not constrain the output to include maximum of k sentences which is a necessity for solving information overload problem in privacy policy comprehension. Therefore, we use IR-based metrics for evaluation and not sentence-level F1 [192] as they did

Both Models	Only full pipeline	None
• Who can see my information?	• Who can see the jobs that I post?	• If its free does Viber profit off its users in some way?
• How do i restrict its access?	• Can other people see your real name?	• Who is allowed to use it?
• Do you sell my data?	• Why do you need so many unrelated permissions?	• Is there any way for a freelancer to contact a customer outside of Fiverr?
• Is my personal information anonymous?	• Can it access my other social media accounts?	• Can my call log be subpoenaed?
• Can people see my workout log?	• Does it need to use the microphone at all?	• Where is the privacy statement?
• Will my information be saved after i use groupon once?	• Is any information recorded?	• What does each permission for the app mean?
• Will my location be tracked?	• Can you guarantee my privacy while playing your game?	• Will the application make money off of the info i enter in the app?
• What permissions does the app require in order to work?	• What information do collaborators have access to?	• How do they keep track of how many people are playing the game?
• Does the wordscapes app collect any personally identifiable information like my name or email?	• Do you use my data to do medical research?	• Does this app owner theft any personal details in my mobile (like photos videos) possibilities are there?
• Who all has access to my data?	• How can i be sure that my saliva samples are being delivered correctly?	• Are there any in game purchases in the wordscapes app that i should be concerned about?

Table 4.6: Examples of user queries answered both our pipeline (R+A+Q) and the closest model to Ravichander et al. [195] (R), only our model or none of the models.

existing in-domain data and familiar techniques such as back-translation trained on out-of-domain data can increase the coverage of user queries, more in-domain data is needed to build paraphrase techniques to adapt the specificity and style of the user queries to privacy policies.

4.6 Conclusion

In this chapter, we took a step toward building an automated privacy policy question-answering assistant. This presentation form provides a more personalized interaction with privacy policies in comparison to the previous approaches. We address two main challenges in this domain: (i) the difference between the language and style of user queries and the legal language of the privacy policies and (ii) low training resources. To so do, we use familiar methods such as lexical substitution and back-translation to bring the language and style of the user queries closer to the language of the policies. Next, we use a language representation model fine-tuned on existing in-domain data to compute a relevancy and informativeness score for each segment in the policy regarding the user query. Finally, the top-ranked passages are presented to the user. Our proposed pipeline can successfully find the relevant information in the privacy policy for 87.7% of the queries in the privacyQA dataset. We observed that using a domain-inspired rule-based approach and training word-vectors on in-domain data is more effective than an out-of-domain NMT-based paraphrase generation approach for bringing the language and style of user queries closer to the language of the privacy policy. However, more in-domain data is needed to build paraphrase techniques to adapt the specificity and style of the user queries to privacy policies. In addition, we observed that relying on existing in-domain resources for building a question-answering assistant provides a sufficiently high-recall retrieval system. However, more resources are required for increasing the precision of the ranking system.

Chapter 5: Low-Resource Data-to-Text Generation with Pretrained Language Models

Expressing natural language descriptions of structured facts or relations – data-to-text generation (D2T) – increases the accessibility of structured knowledge repositories. Previous work [168] shows that pre-trained language models (PLMs) perform remarkably well on this task after fine-tuning on a significant amount of task-specific training data. On the other hand, while auto-regressive PLMs can generalize from a few task examples, their efficacy at D2T is largely unexplored. Furthermore, we have an incomplete understanding of the limits of PLMs on D2T.

In this chapter, we conduct an empirical study of both fine-tuned and auto-regressive PLMs on the DART multi-domain D2T dataset. We consider their performance as a function of the amount of task-specific data and how the data is incorporated into the models: zero and few-shot learning, and fine-tuning of model weights. In addition, we probe the limits of PLMs by measuring performance on subsets of the evaluation data: novel predicates and abstractive test examples. To improve the performance on these subsets, we investigate two techniques: providing predicate descriptions in the context and re-ranking generated candidates by information reflected in the source. Finally, we conduct a human evaluation of model errors and show that D2T generation tasks would benefit from datasets with more careful manual curation.[32]

5.1 Introduction

Structured data repositories, or knowledge bases, contain a wealth of information organized to facilitate automated access and analysis. Automated data-to-text (D2T) generation systems can transform and summarize this knowledge into natural language text snippets that enable broader access [79]. These systems take as input a set of relations, where each relation is a (subject, predicate, object) triple. Applications of this technology include story or dialogue generation [164], open-domain question-answering [146, 71], and text summarization [232]. Domains span journalism [128], weather [193, 158], finance, sports [186, 39, 225], and summarizing patient medical histories [188].

Historically, D2T systems included pipeline approaches with customized models [78], but have now shifted to pretrained Transformer-based language models (PLMs) [56, 144, 189]. Recent examples include Mager et al. [148] and Kale and Rastogi [105], who use models like GPT-2 [189] and T5 [190] to generate natural language descriptions for relations. To support these types of systems,

[32]This research work has been conducted during an internship at Bloomberg L.P.

Nan et al. [168] introduced DART, a large open-domain data-to-text generation corpus. Models trained on DART, both larger and more diverse than previous corpora, improve the performance of BART [130] and T5 on the standard WebNLG challenge [78]. This approach requires a PLM to be fine-tuned on a task-specific in-domain dataset [97, 210, 112]. The promising results achieved by fine-tuning on DART belie the reality – in spite of DART's aspirations, most domains and relations that one could express fail to appear in DART.

A variety of methods have emerged within PLM research to address domain or task adaptation. For example, auto-regressive models, like GPT, have demonstrated improved performance on a wide range of tasks via few-shot learning from a handful of examples [42]. Other strategies, such as prompt tuning [129], can adapt PLMs to specific down-stream tasks by updating only a small subset of model parameters.

While great progress has been made in utilizing PLMs for D2T generation, the path forward is unclear, as we have an incomplete understanding as to which examples they fall short on and the quantity of training resources they need to achieve acceptable performance. More specifically, it is not clear which classes of D2T examples are challenging for these models. In addition, we do not fully understand what classes of errors PLMs are prone to and how the adaptation mechanism (e.g., k-shot learning, fine-tuning) affects the prevalence of these errors.

In this chapter, we conduct an evaluation of PLMs for D2T generation, focusing on two classes of challenging examples: examples with novel (*unseen*) relations (*predicates*) and examples where the source and target sequences are lexically very different (not amenable to purely extractive D2T systems). We consider how GPT-2, adapted with few-shot learning, prompt tuning, and the addition of predicate descriptions, performs on these example classes as compared to a state-of-the-art fine-tuned T5. We show that while GPT-2 performs poorly on DART in the 0-shot setting, its performance can be drasticahally improved by employing the above techniques. We make the following contributions:

- We evaluate GPT2-XL and fine-tuned T5 for D2T generation. While the 0-shot GPT model performs poorly, we evaluate several strategies to improve performance, including few-shot learning and prompt tuning. Both provide significant improvements on the DART dataset.
- We compare model performance on two classes of difficult examples: examples with unseen predicates, and abstractive examples (examples where source and target sequences are lexically dissimilar). We investigate whether including predicate descriptions in the prompt can improve the ability of PLMs on these classes.
- We conduct a human evaluation of PLMs to quantify the prevalence of hallucination and missing information in generations as a function of the model adaptation technique. We find that a re-ranking strategy for few-shot GPT2-XL, despite having little effect on automatic metrics like BLEU, reduces the incidence of missing information, without requiring additional training data.

Finally, we provide recommendations for future model and dataset research in D2T generation.

5.2 Background and Related Work

In the task of data-to-text generation, we are provided a set of triples that include a predicate, subject, and object. The system then produces a text snippet expressing the predicate in natural language. Figure 5.2 shows examples of predicates about sports. The system can be given a set of triples with related predicates (e.g., CLUB, LEAGUE, FORMER_TEAM) and must generate text that expresses the facts encoded by these relations. The resulting text is typically evaluated by comparison to a set of reference texts, which represent various ways of expressing this triple set.

Variations in the formulation of this task depend on the structure of the relations (e.g., tables, triples), the domain of the task (single or open domain), and the source of the data (manually created, automatically derived).

Harkous et al. [88] follow a generate-and-re-rank paradigm to improve the semantic fidelity of the generated text by fine-tuned GPT-2 model. More recently, Ribeiro et al. [198] propose a new task-adaptive pretraining strategy to adapt BART [130] and T5 [190] models for data-to-text generation. They show that adding an intermediate task-adaptive pretraining step between the task-independent pretraining and fine-tuning further improves the performance of these models on data-to-text generation.

Despite the progress of these models, it is not clear which types of D2T examples are most challenging for PLMs or what errors are prevalent in generations. Futhermore, how does PLM adaptation (tuning/prompting) interact with the occurrence of these errors. On the other hand, D2T datasets are not readily available in many domains. Weakly supervised annotation methods (e.g., based on identifying sentences in a corpus that are likely to express a data record) require significant manual effort and often result in annotations with low fidelity between data records and the corresponding textual expression [163]. Training NLG models on such data can result in generations with missing information or hallucination [64, 66, 67]. These issues render the path forward for D2T generation research unclear.

5.3 Model Adaptation

As a supervised task, D2T generation systems rely on previously observed examples to learn the correct generation or level of required ”re-writing” for a predicate. On the other hand, large auto-regressive PLMs (such as GPT2-XL) are able to perform D2T generation without any explicit fine-tuning at all. However, their efficacy on D2T and potential shortcomings are largely unexplored. How well do PLMs perform on relations with a novel predicate? Do PLMs overly rely on copying verbatim from the input or are they capable of abstraction when required? What classes of errors are prevalent in the generations and how do they interact with the choice of adaptation mechanism? Our focus is on the analysis of PLMs for D2T generation.

We study this problem using two types of PLMs: auto-regressive models like GPT-2 and “supervised” models like T5 [190]. While prior work has demonstrated that T5 achieves state of the art results on D2T, these “supervised” models[33] expect task-specific training data, whereas generative

[33]We note that new findings [206] has demonstrated T5 can handle 0-shot task adaptation with the right prompts; this is an evolving research area.

Zero-shot Prompt

Translate Graph to English:

Graph: $<H>$ Alan Martin (footballer) $<R>$ CLUB $<T>$ Hamilton Academical F.C.
English:

Zero-shot Prompt + Relation Description

Translate Graph to English:

Definition: club is an organization of players and managers associated with a particular football team.
Graph: $<H>$ Alan Martin (footballer) $<R>$ CLUB $<T>$ Hamilton Academical F.C.
English:

Figure 5.1: A customized 0-shot prompt for GPT

PLMs excel at adapting to new tasks. Since auto-regressive models have not been fully benchmarked for D2T, we will evaluate them in multiple settings and compare to T5. For both, we will explore the effect of varying training size and their pathological behaviors.

While PLMs can be fine-tuned, their increasing size and training requirements disfavors this approach. Instead, current work assumes a single PLM capable of performing multiple downstream tasks [129]. We adopt GPT2-XL, a decoder-only Transformer [226] with 1.5B parameters pre-trained for language modeling [189].[34] We utilize GPT2-XL as a D2T generation model by varying the amount of supervised information available. Instead of fine-tuning GPT2-XL, we investigate both few-shot learning [189], which is better suited to settings where little training data is available, and prompt tuning, which enables us to tractably update a subset of model weights in spite of GPT2-XL's large parameter count.

5.3.1 0-shot Setting

We start by evaluating GPT2-XL in the 0-shot setting, an especially challenging setting due to a lack of coverage in the training data of pairings between structured records and unstructured text [84]. Ribeiro et al. [198] handled this by including an additional pretraining step. Our focus is on an off-the-shelf GPT2-XL model. We format the input data using the D2T generation infix and prefix formatting of Ribeiro et al. [198] (example in Figure 5.1). We provide no additional context or task-specific training.

[34] WebText (the training dataset) includes content of more than 8 million documents with outbound links from Reddit, a social media platform. Wikipedia (the main data source for DART) is excluded.

Few-shot Prompt

Translate Graph to English:

Graph: <*H*> Paulo Sousa <*R*> CLUB <*T*> ACF Fiorentina
English: Paulo Sousa plays for ACF Fiorentina.
###
Graph: <*H*> Dave Challinor <*R*> CLUB <*T*> Colwyn Bay F.C.
English: Dave Challinor plays for Colwyn Bay F.C.
###
Graph: <*H*> Alan Martin (footballer) <*R*> CLUB <*T*> Hamilton Academical F.C.
English:

Figure 5.2: A customized 3-shot prompt for GPT

5.3.2 Few-shot Setting

We next consider a few-shot setting by augmenting the format of the 0-shot input with reference generations from the training corpus. We evaluate GPT2-XL under the 3-shot learning setting (example in Figure 5.2). For predicates "seen" in the training set, we select three shots with the same predicate uniformly at random from the training set. For "unseen" predicates – predicates not covered in the training set – we randomly select any three examples. Previous work has found that careful shot selection based on input text similarity can be beneficial [140]. However, it's less clear how this would apply to unseen predicates. We leave this for future work.

5.3.3 Prompt Tuning

The expected task for a PLM is indicated by the choice of prompt; ours (Figure 5.1) follows prior work [198, 168]. The prompt includes a prefix ("Graph") and infix token ("English") that indicate the start of the input and the start of the expected output. Auto-regressive language models are sensitive to the choice of prompt, and significant effort is needed to craft effective prompts [141].

Lester et al. [129] proposed an alternate method: prompt tuning. Instead of using discrete prompt tokens, they use "soft-prompts" which are pseudo-token embeddings that are learned during fine-tuning, with all other model parameters held fixed. We follow previous work [129, 46] and use a generic sequence of tokens to denote the prompt prefix $p_{1:s} = (p_1, p_2, ...p_s)$ and infix $q_{1:t} = (q_1, q_2, ...q_t)$. The PLM is provided the input sequence $p_{1:s}$ <*H*> x_1 <*R*> x_2 <*T*> x_3 $q_{1:t}$, where x_1, x_2 and x_3 are head, predicate (relation), and tail strings from the example.

The objective during prompt tuning is to maximize the probability of output sequence $y_{1:m}$ given input data record, prefix $p_{1:s}$, and infix $q_{1:t}$. During training however, only the embedding of the prompt tokens can be updated. Unlike fine-tuning which updates all model parameters on the target task, prompt tuning tunes a small number of parameters (less than 0.01% of all parameters) while keeping most of the language model fixed. While this requires use of the full training set, as opposed to few-shot learning, it illuminates the abilities of GPT2-XL given access to such data.

5.3.4 Domain Knowledge

We explore another way of improving model performance for novel predicates and for examples where significant re-writing is needed: providing definitions for predicates. In many domains, we may find a knowledge base containing many predicates, and definitions for those relations, but no examples of sentences expressing those relations. In these cases, we want to enhance the context of the PLM with predicate definitions. For examples, for the tuple *<H> Genuine Parts <R> DISTRIBUTOR <T> automotive and industrial replacement parts* we may know that DISTRIBUTOR means *"someone who markets merchandise"*. This definition can be helpful to a model that was never exposed to this predicate at training time.

We source predicate definitions for our data from WordNet, a lexical database in English [161], and WikiData.[35] We use WikiData since Wikipedia was the source of many relations in the DART data.[36] An example of the input prompt enhanced with the predicate definition appears in Figure 5.1. We also consider using predicate descriptions in combination with prompt tuning.

5.3.5 Fine-tuned PLM

Our second model type is $T5_{\text{large}}$ [190], a Transformer encoder-decoder architecture with 770M parameters for text generation. The model is pretrained with a denoising objective on a variety of NLP tasks and web-extracted C4 corpus. Unlike GPT2-XL, the denoising objective means an off-the-shelf model performs poorly on unseen tasks, such as D2T generation [190, 129]. We follow Nan et al. [168] and fine-tune $T5_{\text{large}}$ on the DART training set. While this model requires a large amount of supervised examples, it attains state of the art performance on this task.

5.4 Dataset

For our experiments we use DART [168], the largest publicly available open-domain data-to-text generation corpus. DART relies on data from Wikipedia as well as two other commonly used data sets for this task: WebNLG [78] and E2E [172]. Each instance includes a triple set (a set of one or more predicates and their labels) and a natural language reference that expresses the facts in the triple set. We choose DART due to its size and wide coverage of predicate types. Relevant DART statistics appear in Table 5.1. We use the original train, development, and test splits.[37] [38]

---

[35]wikidata.org

[36]DART includes predicates such as *MARGIN_OF_VICTORY* and *INTERMEDIATE_(SOUTH)_WINNERS*. Since descriptions for such relations cannot be found verbatim in WordNet or WikiData, no description is added to those cases.

[37]Nan et al. [168] use version v1.0.0 of DART, whereas we use the publicly available version, v1.1.1.

[38]In the DART dataset, some data records are paired with more than 30 references. Nan et al. [168] do not report the number of references used for their experiments. However in their adaptation of Ribeiro et al. [198]'s fine-tuning script they only use three references. We follow their methodology and only use up to three references per example.

	Train	Dev	Test
Size	30,526	2,768	5,097
#Unique relation types	4,221	419	494
#Ref per example min/avg/max	1/2.0/48	1/2.5/33	1/2.4/35
#Triples per record min/avg/max	1/3.3/10	1/3.7/8	1/3.6/7

Table 5.1: Descriptive statistics of the DART version 1.1.1

Data Splits: The DART test set includes 5,097 examples, of which 4,826 (94.4%) include at least one relation type that appears in the training set. We refer to this subset as the SEEN partition. The remaining 271 instances (5.3%) are considered UNSEEN.[39]

To support additional system analysis, we create another partition of the test data: EASY and HARD. HARD examples are identified by similarity of the input triple to the reference text. In many cases, the reference has high lexical overlap with and similar meaning to the input, while in other cases the generation is non-trivial (see Figure 5.3 for examples). To create the EASY and HARD partitions, we use BERTScore [247] to compute similarity of the input triple with respect to the reference. Examples are ranked based on BERTScore (F1) and the top 10% (510 examples) comprise the EASY partition, while the bottom 10% comprise the HARD partition. By using BertScore to separate EASY and HARD examples, we are not relying purely on lexical overlap to score the difficulty of an example.

5.5 Experimental Setup

Model Training We use the pretrained models GPT2-XL and T5_{large} released by Hugging Face [233], along with their respective tokenizers, for all experiments.

We use beam search with beam size of three for decoding in all models, lightly post-processing the generated text by truncating generations at the newline character. We set maximum generated tokens to 100 and repetition penalty to 1.01 for all experiments.

We used a single V100 GPU with 32GB of memory for all prompt tuning experiments, tuning for a single epoch on the DART train set with prefix and infix length both set to 8 tokens. We use the Adam optimizer [120] with maximum learning rate of 0.1 and 100 warm up steps for the linear learning rate schedule. Training batch size was fixed to 2, with 32 gradient accumulation steps (effective batch size of 64 examples).

We use the scripts from Ribeiro et al. [198] to fine-tune T5 on DART, using identical hyperparameter settings.[40] We use the Adam optimizer with an initial learning rate of 3e-5 and a linearly decreasing learning rate schedule. We fine-tune the model on four GPUs for a maximum of 100 epochs and stop training early if the performance does not improve on the dev set for 15 epochs. Each training epoch takes approximately two hours for each model.

[39] Note that Nan et al. [168] report performance on the "unseen" portion of WebNLG. "Unseen", in this case, means that the relations do not appear in the WebNLG training data; there is no guarantee that they do not appear in the DART training data. Our splits ensure that the UNSEEN partition only contains predicates not seen during DART training.

[40] https://github.com/UKPLab/plms-graph2text (Apache 2.0 license)

Finally, we include a baseline system to benchmark performance of our machine learning models. In a "copy baseline" we simply copy the input text and remove the prefix tokens (<H>, <R>, <T>) as well as special characters (e.g., underscores) common in DART predicates. This baseline performs well for examples with high lexical overlap between input triple set and reference.

Evaluation Metrics Following previous work, we use automated metrics such as BLEU [178], METEOR [55], translation edit rate (TER) [214], and chrF++ [187] for evaluating our generation results. In addition, we also report BERTScore [247] and BLEURT [211]. These metrics go beyond surface form similarities and use contextual embeddings to measure semantic similarity between the generated and reference text.[41]

5.6 Experiments and Results

We evaluate PLMs with various input types and training regimes to answer the following empirical questions:

- How do the adaptation mechanism and level of supervision at train time affect PLM performance on the D2T task?
- What classes of D2T examples are particularly challenging for each PLM? How well do PLMs perform on out-of-sample predicates and examples that are more abstractive (dissimilar source and target sequences)?
- Can we improve performance on examples with unseen predicates by including predicate descriptions in the prompt, as mentioned in §5.3.4?
- Qualitatively, what kinds of errors do PLMs make on the D2T task? Are some adaptation techniques more susceptible to classes of errors than others?
- Can we mitigate some of these errors by re-ranking the decoding results?

Table 5.2 and 5.3 present model performance on the entire DART dataset (ALL), as well as the SEEN and UNSEEN partitions.

Level of Supervision: We first turn to GPT2-XL, which is evaluated on this task without any training data. Following previous work we find that GPT2-XL makes an effective 0-shot model, outperforming the copy baseline according to BLEU and METEOR (row 2). Examining the output more closely, we find that GPT2-XL mostly copies the input; while it outperforms the copy baseline, its strategy is largely the same. We include example generations in Section 5.7. 3-shot GPT2-XL (row 3) does much better than the 0-shot case. Note that in this setting, no model parameters are updated. In addition, the amount of annotated data used for creating 3-shot prompts is much less than what is used for prompt tuning and fine-tuning. While few-shot prompting leads to a boost in BLEU and METEOR, TER increases by 0.14 point. We conjecture that this is due to an increase

[41] We use the evaluation scripts provided in the official WebNLG challenge: https://github.com/WebNLG/GenerationEval (MIT license)

ID	Model	BLEU ↑			METEOR ↑			TER ↓		
		SEEN	UNSEEN	ALL	SEEN	UNSEEN	ALL	SEEN	UNSEEN	ALL
1	copy baseline	4.48	5.07	4.50	0.28	0.31	0.28	0.92	0.86	0.92
2	GPT2-XL (0-shot)	13.13	13.88	13.26	0.23	0.27	0.23	0.69	0.78	0.70
3	GPT2-XL(3-shot)	26.74	23.72	26.65	0.29	0.28	0.29	0.85	0.78	0.84
4	GPT2-XL-PT	33.55	29.86	33.41	0.24	0.28	0.24	0.65	0.61	0.65
5	GPT2-XL-PT + Reranking	31.03	31.67	31.09	0.28	0.30	0.28	0.63	0.58	0.63
6	$T5_{large}$	48.41	43.48	48.25	0.39	0.40	0.39	0.46	0.44	0.46
+Descriptions										
7	GPT2-XL(0-shot)	11.45	8.05	11.4	0.20	0.19	0.20	0.70	1.00	0.72
8	GPT2-XL(3-shot)	26.32	21.30	26.14	0.28	0.27	0.28	0.83	0.89	0.83
9	GPT2-XL-PT	33.96	31.37	33.85	0.24	0.28	0.24	0.66	0.59	0.66
10	$T5_{large}$	48.56	43.82	48.4	0.39	0.39	0.39	0.46	0.45	0.46

Table 5.2: Model results on test set of the DART dataset. ↑: Higher is better. ↓: Lower is better.

ID	Model	chrF++ ↑			BERTScore(F1) ↑			BLEURT ↑		
		SEEN	UNSEEN	ALL	SEEN	UNSEEN	ALL	SEEN	UNSEEN	ALL
1	copy baseline	0.33	0.34	0.33	0.83	0.85	0.83	-0.59	-0.29	-0.58
2	GPT2-XL (0-shot)	0.34	0.34	0.34	0.88	0.87	0.88	-0.46	-0.30	-0.46
3	GPT2-XL (3-shot)	0.48	0.44	0.48	0.91	0.91	0.91	-0.19	-0.17	-0.19
4	GPT2-XL-PT	0.40	0.44	0.40	0.92	0.92	0.92	-0.11	0.06	-0.10
5	GPT2-XL-PT + Reranking	0.46	0.47	0.46	0.92	0.92	0.92	-0.01	0.12	0.00
6	$T5_{large}$	0.64	0.64	0.64	0.95	0.95	0.95	0.38	0.44	0.39
	+ Description									
7	GPT2-XL (0-shot)	0.31	0.23	0.30	0.88	0.86	0.88	-0.46	-0.54	-0.46
8	GPT2-XL (3-shot)	0.47	0.42	0.46	0.91	0.90	0.91	-0.19	-0.16	-0.19
9	GPT2-XL-PT	0.39	0.45	0.39	0.91	0.92	0.91	-0.14	0.09	-0.13
10	$T5_{large}$	0.64	0.63	0.64	0.95	0.95	0.95	0.38	0.43	0.38

Table 5.3: Performance on the DART test set, partitioned by whether predicates are SEEN, UNSEEN, and overall. ↑: Higher is better.

in hallucinated content in this setting. We take a closer look at these pathological behaviors in our human evaluation.

Both GPT2-XL models prompt tuned on the entire DART dataset (row 4 and 5) outperform the 3-shot model by a wide margin. As reported previously [168], we also notice that fine-tuned T5 (row 6) performs well on this task surpassing either prompt tuned GPT2-XL model.

Consistent with previous findings, we also notice that the more training data that is used to adapt the model (either by few-shot learning or training model weights), the better PLMs perform. However, in a resource-constrained setting, few-shot GPT2-XL achieves reasonable performance. Few-shot adaptation might be a good choice for D2T when the number of unique predicates in the test set is small, and only very few examples can be manually annotated. On the other hand, if more data is available, fine-tuning T5 leads to better results for D2T. In fact, our experiments show that T5 can surpass the 3-shot GPT2-XL after fine-tuning on only 200 examples. See our experiments under ”Training Curves” for more details.

Training Curves: In this experiment, we seek to answer that how much data does T5 require to do well on this task? Specifically, how many examples are required for T5 to exceed the performance of the few-shot GPT2-XL? We fine-tune T5 on increasingly larger amounts of training data. We start off with an off-the-shelf T5 model with no additional training. We then vary the number of training examples in $\{10, 20, 50, 100, 200, 500\}$.[42] We repeat each setting five times by resampling a training set and fine-tuning T5, and report results for each training set size averaged cross all test partitions. Figure 5.4 shows the BLEU performance (y-axis) of T5 as a function of number of training examples (x-axis). Performance of the copy baseline, 0-shot, 3-shot, and prompt tuned GPT2-XL are indicated by horizontal lines. Without any task-specific fine-tuning, T5 does slightly worse than the copy baseline, easily outperformed by 0-shot GPT2-XL. In settings without training data, GPT2-XL is the clear choice. T5 continues to lag behind GPT2-XL 3-shot until trained on at least 200 examples, and meets the performance of GPT2-XL prompt tuned after training on 500.

Predicate Novelty: As expected, the copy baseline (row 1) performs poorly across all conditions, but consistent for both the SEEN and UNSEEN partitions. 0-shot GPT2-XL also performs similarly on both partitions, since it was not trained on any task data. GPT2-XL with a 3-shot prompt (row 3) outperforms 0-shot on both partitions, despite the unseen prompts including unrelated predicates; the model still benefits from multiple shots even if they do not contain the same predicates (+9.84 BLEU points).

Prompt tuning and re-ranking generated samples by overlap with the triple set entities both improve the performance of GPT2-XL on novel predicates. Overall, GPT2-XL performs consistently across SEEN and UNSEEN partitions, while T5 performance is more sensitive to whether the predicate was observed during training (e.g., difference of 4.93 points BLEU in row 6). We do not see a consistent performance drop going from SEEN to the UNSEEN partition when looking at chrF++, BertScore, and BLEURT. This is somewhat surprising, but also hard to interpret given that chrF++ relies on character n-gram and BertScore and BLEU rely in contextualized embeddings.

We next turn to evaluating the impact of augmenting prompts with predicate descriptions for unseen predicates. This process is described in §5.3.4. We evaluate this augmentation in the 0-shot (row 7), 3-shot (row 8) and prompt tuning (row 9) settings, as well as in T5 fine-tuning (row 10). We observe very small improvements on the UNSEEN partition and only in cases where model parameters are updated (rows 9 and 10). We suspect that as descriptions are sourced from WordNet and WikiData, either many predicates could not be resolved to a description in these tables, or the predicates that could be resolved were largely self-explanatory. We conjecture that in the 0-shot setting, conditioning the generation on descriptions might distract the model from the head and tail entity. On the other hand, many of the unseen predicates in DART are not words that can be easily resolved. However, we suspect that if they were to be reliably resolved, specialized domains such as finance or medicine would benefit from adding predicate descriptions.

Generation Difficulty: Table 5.4 shows the performance of all models on the EASY and HARD partitions. All models have noticeably worse performance on HARD examples, where more abstraction is needed. The best performing model, T5 (row 16), has a gap of 0.16 METEOR between the EASY and HARD partition, while the prompt tuned GPT2-XL (row 14) has the smallest difference in performance between the partitions. It is clear that these models perform well overall when copying

[42]We use the same hyper-parameters as before except for the number of training epochs and batch size. To avoid over-fitting on small data, we only fine-tune for 1 epoch. We use batch size of 2.

ID	Model	BLEU ↑		METEOR ↑		chrF++ ↑		TER ↓		BERTScore(F1) ↑		BLEURT ↑	
		EASY	HARD	EASY	HARD	EASY	HARD	EASY	HARD	EASY	HARD	EASY	HARD
11	copy baseline	18.00	2.01	0.41	0.23	0.45	0.32	0.79	0.99	0.88	0,80	0.12	-1.00
12	GPT2-XL (0-shot)	22.20	6.92	0.34	0.18	0.47	0.31	0.83	0.64	0.90	0.88	-0.09	-0.54
13	GPT2-XL (3-shot)	34.97	1.88	0.34	0.06	0.54	0.07	0.82	0.38	0.92	0.93	-0.09	-0.11
14	GPT2-XL-PT	42.81	31.78	0.35	0.23	0.57	0.39	0.48	0.69	0.94	0.92	0.31	-0.17
15	GPT2-XL-PT + Reranking	43.35	25.79	0.37	0.29	0.60	0.48	0.47	0.66	0.94	0.93	0.34	-0.04
16	$T5_{large}$	70.54	38.34	0.51	0.35	0.80	0.57	0.23	0.59	0.97	0.94	0.70	0.20
+Descriptions													
17	GPT2-XL (0-shot)	19.00	6.43	0.30	0.17	0.42	0.31	0.93	0.65	0.89	0.88	-0.20	-0.54
18	GPT2-XL (3-shot)	34.19	20.54	0.38	0.26	0.61	0.44	0.92	0.81	0.93	0.91	0.07	-0.26
19	GPT2-XL-PT	42.52	33.1	0.34	0.23	0.56	0.39	0.5	0.69	0.93	0.91	0.28	-0.21
20	$T5_{large}$	70.06	38.49	0.51	0.34	0.80	0.57	0.23	0.60	0.97	0.94	0.69	0.20

Table 5.4: Model results on EASY and HARD partitions of the DART test set. ↑: Higher is better. ↓: Lower is better.

from the input suffices, but do poorly when significant rewriting is required. In many domains, we may prefer models with more diverse, creative generations, a task at which these models do not do well. On the other hand, DART is a mostly automatically derived dataset, with significant errors in some examples, where the reference text may contain information that is unsupported by the input triple. These examples may pervade the HARD partition.

Next, we investigate the impact of adding predicate descriptions on D2T of the HARD partition. In the few-shot setting, adding predicate descriptions improves the BLEU score to 20.54 on the HARD partition (row 18). Conditioning the model on predicate descriptions significantly enhances it's re-writing ability. For the prompt tuned GPT2-XL, BLEU score improves to 33.1 (row 19). However, we do not see any gains for 0-shot GPT or T5 (row 17 and 20). Overall, GPT2-XL benefits from predicate descriptions on examples where significant re-writing is needed, even when additionally prompt tuned. GPT2-XL with prompt tuning achieves competitive results with benchmark T5 on the HARD partition (33.1 vs 38.49 BLEU).

Human Evaluation: To further examine the pathological behaviors of the models, we randomly sampled 50 examples from the DART test set for human evaluation. For each example, the output of T5 and GPT2-XL in the 3-shot, prompt tuned, and re-ranked settings were presented to two annotators.[43] We also showed the reference text as another candidate, with the generating model identity hidden. Annotators evaluated output quality based on three criteria: (1) whether it contains hallucinated content *(hallucination)* (2) whether the text is missing information from the input records *(missing info)*, and (3) *fluency*. Annotators indicated agreement with each of these Likert items on an ordinal scale from 1 (strongly disagree) to 5 (strongly agree).

Table 5.5 presents average annotator score according to each of these Likert items. GPT2-XL in the 3-shot setting often misses information. Notably, both prompt-tuned variations generate very fluent text. Re-ranking improves the quality of the generations by decreasing the amount of missing information and improving fluency. While the best GPT2-XL model does very similar to $T5_{large}$ in terms of fluency, on average it hallucinates or misses information more often.

Re-ranking: GPT2-XL prompt tuned is both parameter efficient and generalizes very well to novel predicates. It also does very well on examples that require more re-writing. It approaches the performance of fine-tuned $T5_{large}$ according to avoiding hallucinations and fluency. During the

[43]Two of the paper authors.

Source	Hallucination ↓	Missing Info ↓	Fluency ↑
Reference	1.53	1.19	4.51
GPT2-XL(3-shot)	3.26	3.61	3.17
GPT2-XL-PT	1.73	3.35	4.64
GPT2-XL-PT + Ranking	1.73	2.79	4.75
T5 $_{\text{large}}$	1.16	1.23	4.79
Agreement	0.64	0.77	0.50

Table 5.5: Results of the qualitative evaluation. ↓: Lower is better. ↑: Higher is better. Inter-annotator agreement is measured by Kendall's τ rank correlation coefficient.

human evaluation, we observe that this model would often miss subject or object of the predicate in its generations (see our human evaluation for details). We can mitigate this problem without additional model training through a re-ranking strategy to ensure that the selected generation contains all relevant information.

We first create multiple candidate generations by increasing beam size during decoding. Next, we compute the percentage of head and tail entities covered in the text. Finally, we pick the candidate that contains the highest percentage of entity spans from the input triple.[44] Rows 5 and 15 show the results of re-ranking a GPT2-XL prompt tuned model. Re-ranking modestly improves performance on all partitions, and across all metrics except BLEU.

5.7 Sample Model Output

In this section, we share a few samples from the DART test set as well as outputs generated by different models. We qualitatively compare different models and highlight a few of their common errors.

Task Prompting: As seen in Examples 1 and 2, GPT2-XL in the 0-shot setting often copies the input. GPT2-XL with a 3-shot prompt generates a much more fluent text than the 0-shot case. This can be seen in Examples 2, 4, and 5. Although GPT2-XL with few-shot prompting generates more fluent text, it often generates hallucinated content (see Example 3).

We see that prompt tuning further boosts our performance and generates a more coherent text in comparison to few-shot GPT2-XL (see Example 1 and 3). Moreover, it hallucinates much less than the few-shot setting (e.g. see Example 3). We also saw this previously in Table 5.2, as the prompt tuned GPT2-XL achieved lower TER score. In contrast to T5 training, in which all model parameters are updated, prompt tuning adapts only a small fraction of the model parameters. However, in many cases the generated text is as good as the benchmark T5 (see Example 2). Despite generating very fluent text, prompt tuned GPT2-XL often misses information from one or more relations (Examples 1, 3, and 4).

[44]We use a beam size of 20 during decoding. Prior to measuring the entity coverage in the candidates, we normalize the text by lower casing and removing special characters.

Re-ranking: Re-ranking based on entity coverage solves the missing information issue in several cases. For example, in Example 3, the entity *Alvis Speed 25* which is missed by the prompt tuned GPT2-XL, is covered after re-ranking. The benefit of re-ranking also can be seen in Example 4. On the other hand, in Example 2, ranking does not solve the missing information issue. This is because argument ”yes” of ”family-friendly” probably would not naturally appear in generated text (e.g., ”Yes, this is a family-friendly restaurant”). For such cases, the re-ranking heuristic will not provide useful feedback.

Predicate Descriptions: As mentioned before, in several cases, the description extracted from WordNet and WikiData are trivial. In Example 2, the definition of relations *food*, *area*, and *near* add no information beyond the word itself, and therefore not helpful for the model. On the other hand, it seems like defining relation *MANUFACTURER* in Example 3 has improved generations of GPT2-XL in both the few-shot and prompt-tuned settings. In some cases, while the predicate description can be potentially useful, the model ignores the augmented description. For example, in 4, the definition of relation *GENRE* is not covered in the generated text of any of models.

5.8 Conclusion

In this chapter, we systematically analyze the performance of two PLMs – T5 and GPT2-XL – for D2T generation by examining performance based on the choice of adaptation mechanism: fine-tuning, prompt tuning, and few-shot learning. We observe that while fine-tuning on more data leads to better performance, when no training data is available, GPT2-XL (0-shot) outperforms T5. With a small number of training examples, few-shot GPT2-XL is a more appropriate solution for D2T.

We also conduct a thorough investigation of D2T challenges for PLMs by evaluating them on two divisions of the DART test set: novel predicates and abstractive examples. We show that the performance of fine-tuned T5 drops significantly on unseen predicates. On the other hand, the performance of few-shot GPT2-XL on unseen predicates can be enhanced even with shots containing unrelated predicates. We also notice that T5 and GPT2-XL both do well at D2T by copying the input. However, they do noticeably worse on examples where significant re-writing is needed. Adding domain knowledge (predicate descriptions) to the prompts can improve the performance of few-shot GPT2-XL on this subset by a large amount. We also conduct a human evaluation of the generations and find that prompt tuned GPT2-XL generations can be improved by re-ranking generations by overlap with the input entity spans.

EASY **Examples**

Input: <H> Adolfo Suárez Madrid–Barajas Airport <R> LOCATION <T> Madrid, Paracuellos de Jarama, San Sebastián de los Reyes and Alcobendas
Reference: Adolfo Suárez Madrid–Barajas Airport can be found in Madrid, Paracuellos de Jarama, San Sebastián de los Reyes and Alcobendas.'
###
Input: <H> Alderney Airport <R> RUNWAY_NAME <T> "14/32"
Reference: Alderney Airport runway name is 14/32
###
Input: <H> Asunción <R> IS_PART_OF <T> Gran Asunción
Reference: Asunción is a part of Gran Asunción.
###
Input: <H> Airey Neave <R> AWARD <T> Military Cross
Reference: Airey Neave was awarded the Military Cross.

HARD **Examples**

Input: <H> 2004 <R> MOVEMENTS <T> Promotion Playoffs - Promoted <H> 2004 <R> POSITION <T> 1st
Reference: Sports stats for Ljungskile SK
###
Input: <H> thierry morin <R> POSITION <T> defender <H> [TABLECONTEXT] <R> NAME <T> thierry morin <H> [TABLECONTEXT] <R> [TITLE] <T> Players
Reference: Thierry Morin was a defender for Paris Saint-Germain.
###
Input: <H> ALV X-1 <R> COUNTRY_ORIGIN <T> United States <H> United States <R> ETHNIC_GROUP <T> African Americans <H> United States <R> DEMONYM <T> Americans
Reference: Originating in the United States and by Americans, some of African decent is the ALVX-1.', 'ALVX-1 comes from the US where Americans live and African Americans are an ethnic group
###
Input: <H> past tense <R> SEASON_# <T> 4 <H> past tense <R> ORIGINAL_AIR_DATE <T> october29,2008 <H> past tense <R> NO._IN_SERIES <T> 13 <H> past tense <R> U.S._VIEWERS_(MILLIONS) <T> 7.93 <H> past tense <R> DIRECTED_BY <T> michael pressman <H> past tense <R> WRITTEN_BY <T> craig turk
Reference: Past Tense was the 13th episode in the series.

Figure 5.3: Examples from the EASY and HARD partition

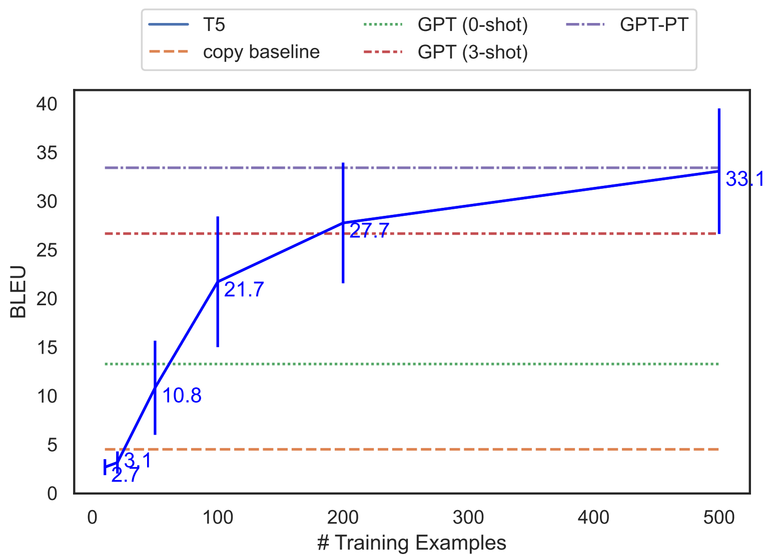


Figure 5.4: Impact of fine-tuning data size on performance of T5. Numbers reflect average performance over 5 different data samples, with standard error of the mean indicated by bars.

Example #1

Input: <H> Aleksey Chirikov (icebreaker) <R> SHIP_BEAM <T> 21.2 <H> Aleksey Chirikov (icebreaker) <R> STATUS <T> ""In service"" <H> Aleksey Chirikov (icebreaker) <R> BUILDER <T> Helsinki <H> Aleksey Chirikov (icebreaker) <R> BUILDER <T> Arctech Helsinki Shipyard
Reference: The icebreaker Aleksey Chirikov was built at the Arctech Helsinki Shipyard in Helsinki. The Aleksey Chirikov (icebreaker) is in service and its ship beam is 21.2
Description: BUILDER means a person who constructs something
GPT2-XL (0-shot): <H> Aleksey Chirikov (icebreaker) <R> SHIP_BEAM <T> 21.2 <H> Aleksey Chirikov (icebreaker) <R> STATUS <T> "In service" <H> Aleksey Chirikov (icebreaker) <R> BUILDER <T> Helsinki <H> Aleksey Chirikov (icebreaker) <R> BUILDER <T>
GPT2-XL (3-shot): Aleksey Chirikov (icebreaker) is of type 21.2.
GPT2-XL (3-shot) + Description: Aleksey Chirikov (icebreaker) was a Russian icebreaker.
GPT2-XL-PT: Aleksey Chirikov (icebreaker) was built by Arctech Helsinki Shipyard.
GPT2-XL-PT + Description: Aleksey Chirikov (icebreaker) was built by Arctech Helsinki Shipyard.
GPT2-XL-PT + Reranking: Aleksey Chirikov (icebreaker) is in service at Arctech Helsinki Shipyard.
$\mathbf{T5_{large}}$: The ship Aleksey Chirikov was built by Arctech, Helsinki Shipyard and has a ship beam of 21.2 metres. It is currently in service.

Example #2

Input: <H> The Mill <R> eatType <T> pub <H> The Mill <R> food <T> Fast food <H> The Mill <R> area <T> city centre <H> The Mill <R> familyFriendly <T> no <H> The Mill <R> near <T> Café Sicilia
Reference: The Mill pub is near Café Sicilia in the city centre that isn't family friendly, is average and has Fast food.
Description: eatType means type of food served by a restaurant or restaurant chain. food means any substance that can be metabolized by an animal to give energy and build tissue. area means area occupied by an object. near means move towards.
GPT2-XL (0-shot): Graph: <H> The Mill <R> eatType <T> pub <H> The Mill <R> food <T> Fast food <H> The Mill <R> area <T> city centre <H> The Mill <R> familyFriendly <T> no <H> The Mill <R> near <T> Café Sicilia
English:
GPT2-XL (3-shot): The Mill is located in the city centre near Café Sicilia. It serves fast food in the price range of £20-25 with a moderate customer rating
GPT2-XL (3-shot) + Description: The fast food restaurant, The Mill, has an average rating of 3 out of 5 perhaps as it is not friendly to kids. The Mill is located in the city centre area near Café Sicilia.
GPT2-XL-PT: The Mill is a fast food pub in the city centre near Café Sicilia.
GPT2-XL-PT + Description: The Mill is a family friendly pub near Cafe Sicilia.
GPT2-XL-PT + Reranking: The Mill is a fast food pub in the city centre near Café Sicilia.
$\mathbf{T5_{large}}$: The Mill is a fast food pub in the city centre near Café Sicilia.

Figure 5.5: Examples of model outputs

Example #3

Input: <H> Alvis Speed 25 <R> MANUFACTURER <T> Alvis Car and Engineering Company <H> Alvis Car and Engineering Company <R> LOCATION_CITY <T> Coventry
Reference: Alvis Speed 25 was made by the Alvis Car and Engineering Company of Coventry.
Description: MANUFACTURER means manufacturer or producer of this product.
GPT2-XL (0-shot): <H> Alvis Speed 25 <R> MANUFACTURER <T> Alvis Car and Engineering Company <H> Alvis Car and Engineering Company <R> LOCATION_CITY <T> Coventry
GPT2-XL (3-shot): Alvis Speed 25 won the race in Coventry on April 25, 1996.
GPT2-XL (3-shot) + Description: Alvis Speed 25 is a car manufactured by Alvis Car and Engineering Company in Coventry, England
GPT2-XL-PT: Alvis Car and Engineering Company is located in Coventry.
GPT2-XL-PT + Description: Alvis Speed 25 is manufactured in Coventry.
GPT2-XL-PT + Reranking: Alvis Speed 25 is manufactured by Alvis Car and Engineering Company in Coventry.
T5$_{\text{large}}$: Alvis Car and Engineering Company is located in Coventry and manufactures the Alvis Speed 25.

Example #4

Input: <H> Anders Osborne <R> ASSOCIATED_BAND_ASSOCIATED_MUSICAL_ARTIST <T> Galactic <H> Anders Osborne <R> ASSOCIATED_BAND_ASSOCIATED_MUSICAL_ARTIST <T> Tab Benoit <H> Anders Osborne <R> GENRE <T> Blues <H> Anders Osborne <R> ASSOCIATED_BAND_ASSOCIATED_MUSICAL_ARTIST <T> Billy Iuso <H> Anders Osborne <R> ACTIVE_YEARS_START_YEAR <T> 1989
Reference: Anders Osborne musical career began in 1989. His career began in the blue's genre. He has been associated with the band, Galactic as well as Tab Benoit. Osborne also has ties to fellow musician Billy Iuso.
Description: GENRE means creative work's genre or an artist's field of work
GPT2-XL (0-shot): <H> Anders Osborne <R> ASSOCIATED_BAND_ASSOCIATED_MUSICAL_ARTIST <T> Galactic <H> Anders Osborne <R> ASSOCIATED_BAND_ASSOCIATED_MUSICAL_ARTIST <T> Tab Benoit <H> Anders Osborne <R> GENRE <T> Blues <H> Anders Osborne <R> ASSOCIATED_BAND_ASSOCIATED_M
GPT2-XL (3-shot): Anders Osborne is an English singer, songwriter and guitarist.
GPT2-XL (3-shot) + Description: Anders Osborne is an active musician and composer.
GPT2-XL-PT: Tab Benoit and Anders Osborne are associated with Galactic.
GPT2-XL-PT + Description: Tab Benoit and Anders Osborne are associated with Galactic.
GPT2-XL-PT + Reranking: Tab Benoit is associated with Galactic and Anders Osborne is associated with Billy Iuso.
T5$_{\text{large}}$: Anders Osborne is associated with the band Galactic, Billy Iuso, Tab Benoit and the genre of Blues. His active years start in 1989.

Figure 5.6: Examples of model outputs

Example #5

Input: <H> Nurburgring <R> FASTEST_LAP <T> PSV Eindhoven <H> Nurburgring <R> WINNING_CLUB <T> A.C. Milan
Reference: PSV Eindhoven had the fastest lab on the Nurburgring, and A.C. Milan was the winning club at that race.
Description: FASTEST_LAP means person who does the fastest lap during a motorsport race.
GPT2-XL (0-shot): <H> Nurburgring <R> FASTEST_LAP <T> PSV Eindhoven <H> Nurburgring <R> WINNING_CLUB <T> A.C. Milan
GPT2-XL (3-shot): A.C. Milan had the fastest lap in the race won by PSV Eindhoven.
GPT2-XL (3-shot) + Description: A.C. Milan set the fastest lap at the Nurburgring when the winning team was PSV Eindhoven.
GPT2-XL-PT: Nurburgring is the fastest lap at A.C. Milan.
GPT2-XL-PT + Description: PSV Eindhoven is the winning club at the Nurburgring.
GPT2-XL-PT + Reranking: Nurburgring is the fastest lap at A.C. Milan
T5$_{large}$: A.C. Milan won the race where PSV Eindhoven had the fastest lap.

Figure 5.7: Examples of model outputs

Chapter 6: Fairness aware summarization for justified decision making

In consequential domains such as recidivism prediction, facility inspection, and benefit assignment, it's important for individuals to know the decision-relevant information for the model's prediction. In addition, predictions should be fair both in terms of the outcome and the justification of the outcome. In other words, decision-relevant features should provide sufficient information for the predicted outcome and should be independent of the membership of individuals in protected groups such as race and gender. In this chapter, we focus on the problem of (un)fairness in the justification of the text-based neural models. We tie the explanatory power of the model to fairness in the outcome and propose a fairness-aware summarization mechanism (`FairSum`) to detect and counteract the bias in such models. Given a potentially biased natural language explanation for a decision, we use a multi-task neural model and an attribution mechanism based on integrated gradients to extract high-utility and low-bias justifications in form of a summary. The extracted summary is then used for training a model to make decisions for individuals. Results on several real-world datasets suggest that our method drastically limits the demographic leakage in the input (fairness in justification) while moderately enhancing the fairness in the outcome. Our model is also effective in detecting and counteracting several types of data poisoning attacks that synthesize race-coded reasoning or irrelevant justifications. In addition, results from a pilot user study indicates that our model is an effective pre-processing approach for neutralizing the textual data (in terms of the protected attribute) while preserving decision-relevant information.

6.1 Introduction

AI systems are increasingly adopted to assist or replace humans in several highly consequential domains including recidivism assessment [14], policing [202, 114], credit card offering [216], lending [123], and prioritizing resources for inspection [45]. To maximize the utility, such models are trained to minimize the error on historical data (decisions made by humans in the past). However, historical decisions can have unfair outcomes or be based on unfair arguments. Training models on historical decisions with unfair outcomes or justifications can reinforce the biases that already exist in our society. In fact, training models without fairness considerations has already resulted in several cases of discrimination [121, 174, 5, 23]. Discrimination in this context is defined as the unjustified distinction between individuals based on their membership in a protected group (e.g. gender identity or ethnicity). The concerns and observations regarding the unfairness of AI algorithms have led to a growing interest in defining, measuring, and mitigating algorithmic unfairness [183, 19, 45, 77, 96].

[45] https://chicago.github.io/food-inspections-evaluation/

A large body of research on the fairness of AI has focused on mitigating the bias in decision-making by minimizing the difference between treatment and outcome among different protected groups (see § 6.2).

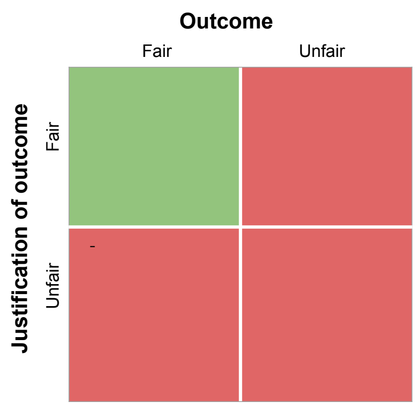


Figure 6.1: A fairly-justified decision should have a fair outcome and be based on fair justifications.

While training models on historical decisions with unfair outcomes is detrimental, using historical training data with unfair justifications is equally harmful. For example, training a text-based neural model on unfair justifications can cause the model to associate a gender or race-coded phrase in the input to a certain outcome. This phenomena is an example of disparate impact [13, 241]. On the other hand, it is possible that individuals from two or more protected groups are treated differently (received different outcomes). But the differences can be justified and explained using multiple fair arguments and therefore is not considered illegal [157]. For example, Kamiran et al [106] state that the difference in income level in females and males in the UCI adult income dataset [46] — a well-studied dataset in algorithmic fairness research — can be attributed to the difference in working hours. Methods that do not take into account the explainability aspect of discrimination will result in reverse discrimination [106]. This highlights the need to distinguish between the fairness of the outcome and fairness in the justification of the outcome (see Figure 6.1). A fairly-justified decision should both have a fair outcome and be fairly justified. In other words, the justification should include enough information to explain the outcome [31] and should not be based on information about membership in protected groups.

While there are several sources for unfairness in the reasoning of AI models, in this chapter, we focus on detecting and counteracting biases in the justification of text-based decision-making models. We propose a fairness-aware summarization mechanism as a pre-processing step to reduce potential biases from textual justifications. We propose methods to first identify and measure bias in textual explanations and then mitigate this bias using a filtering-based approach. We measure bias by using metrics such as demographic parity [27], equalized odds [86], and calibration [121], and by measuring the adversary's ability to identify membership in protected groups given the textual explanations. To counteract the bias, our proposed summarization model obfuscates the arguments that are not useful for decision making or are only useful when they correlate with the protected attribute. Finally, the extracted fairly-justified summaries are used to train a final model. This preprocessing approach ensures learning a model that is both transparent and agnostic about gender-coded or race-coded arguments [47]. Our framework can potentially assist users in understanding the

[46]https://archive.ics.uci.edu/ml/datasets/adult

[47]Note that we do not claim or assume that "fair explanations should avoid mentioning the protected attribute". Rather methodologically, we are removing the signals about the protected attribute to test whether the rest of the arguments still sufficiently justify the outcome.

decisions that are made for them by presenting the most predictive justifications. To summarize, in this study, we make the following contributions:

- We propose the use of a multi-task model and an attribution mechanism to attribute the decision of the model as well as potential biases in the justification to certain parts of the inputs.
- We propose a fairness-aware summarization model (`FairSum`) to condense the input explanations. Our model extracts the decision-relevant justifications while removing the potentially unfair ones. Our proposed preprocessing approach is independent of modeling and can be integrated into the data science pipeline with other in-processing and post-processing fairness enhancement mechanisms.
- We show that applying `FairSum` on the input data does not hurt the utility of the model but significantly limits the leakage of information about protected attributes of individuals.
- We show that using `FairSum` to obfuscate the race-coded or gender-coded input justifications moderately enhances the fairness in the outcome.
- We test the performance of our proposed approach under several types of unfairness attacks.
- We conduct a preliminary pilot user study to examine the impact of `FairSum` on users performance in teaching evaluation task. In addition, we evaluate the impact of our method on neutralizing the teaching reviews in terms of gender.

6.2 Related Work

Machine Learning Fairness: Techniques proposed to enhance fairness in machine learning algorithms can be broadly categorized into pre-processing methods, in-processing methods, and post-processing methods [183]. Pre-processing mechanisms use re-weighting, relabeling, or other transformations of the input data to remove dependencies between the class label and the sensitive attributes before feeding it to the machine learning algorithm [82, 28, 244, 73, 54, 69, 235]. This class of approaches is closely related to the field of privacy [68]. Since both fairness and privacy can be enhanced by obfuscating sensitive information from the input data with the adversary goal of minimal data perturbation [111, 101]. In-processing methods modify the optimization procedure of the classifier to integrate fairness criteria in the objective function [108, 3, 26]. This is often done by using a regularization term [60, 241, 242, 243, 83, 17, 18, 191, 107], meta-learning algorithms [34], reduction-based methods [2, 50], or adversarial training [147, 245, 33, 227]. Post-processing methods adjust the output of the AI algorithm to enhance fairness in decisions [75]. For example, by flipping some of the decisions of the classifier [86] or learning a different classifier [65] or a separate threshold for each group [159]. Our proposed approach of using fairness-aware text summarization to remove bias from the input explanations belongs to the first category. The majority of the introduced methods mitigate bias in decision-making by minimizing the difference between treatment and outcome among different protected groups. Our proposed approach is distinct from previous work in a few ways. In contrast to the approaches that are intended to enhance the fairness of the model's outcome, our proposed approach is intended to enhance the fairness in the justification of the outcome. Moreover, many of the existing preprocessing approaches produce an intermediate data representation that is not interpretable to many stakeholders [82, 244]. The output of our model is an extractive summary of the input justifications. This is preferable for many applications where interpretability is essential.

Text Summarization: Our work is also related to the field of automatic text summarization. The general goal of this task is to shorten a text while preserving the key information. Automatic summarization methods can be broadly categorized as abstractive [204, 209, 135] and extractive [166, 207]. Our work belongs to the latter category. In extractive summarization, a subset of phrases or sentences in the input document are selected based on an importance score to be included in the final summary. Defining importance is highly domain-specific. However, for extracting generic summaries earlier work has explored using heuristics such as frequency of significant words, coverage of salient concepts [61, 74], or the centrality in the document graph [160] to rank and select sentences in a document. More recently, data-driven approaches rely on deep neural models to extract summaries by creating sentence representation and training a supervised model to learn whether to include a sentence in the summary or not [125, 142]. While extractive summarization has proved a great solution for applications such as privacy [153, 113, 116], and legal decision making [72, 109, 250], to the best of our knowledge we are the first to use text summarization for detecting and obfuscating biases in the input data while preserving the decision-relevant information. Next, we will formally define our problem and explain our proposed solution.

6.3 Problem Formulation

Given a dataset consisting of n samples $\{(X_i, Y_i, P_i)\}_{i=1}^{n}$ where X denotes a textual explanation written by the decision-maker to provide evidence or justify an outcome Y and P indicates one or more protected variables[48], we aim to extract a fairly-justified summary $\{X_i{}'\}_{i=0}^{n}$ such that X' provides sufficient information to predict and justify Y and X' is independent of protected variable P. We explain how we measure and attribute these qualities to sentences in the justification X in § 6.4. For instance, Y_i could represent a court decision for individual i, which is a member of the demographic group P_i and has received a textual argument X_i regarding this decision[49]. Potentially, X_i can be biased toward certain demographic groups. Our goal is to transform a given dataset $\{(X_i, Y_i, P_i)\}_{i=1}^{n}$ into a new dataset $\{(X_i{}', Y_i, P_i)\}_{i=1}^{n}$ that is decontaminated from unfair arguments. To achieve this goal, we use a fairness-aware extractive summarization model as a data pre-processing step.

6.4 Proposed Methodology

In this section, we explain our proposed methodology to extract a fairly-justified summary $\{X_i{}'\}_{i=0}^{n}$ such that summary X' provides sufficient information to predict and justify $\hat{Y}$ and the

[48] We only assume the existence of a set of discrete predefined protected attributes that are relevant to the problem in hand. An example of this is protected groups in the US legal system such as race, gender, and nationality. However, the proposed approach is intended to work with any set of predefined groups.

[49] While we assume the availability of information about individuals protected attribute at train time, we do not assume that P_i is known at inference time.

extracted summary X' is independent of protected variable P. A graphical model of the proposed approach is shown in Figure 6.2. Given an input explanation X_i consisting of sentences $\{s_1, s_2, ..., s_m\}$, the goal of our model is to select a subset of these sentences subject to a utility and a fairness constraint. Next, we explain how we measure and attribute the utility and discrimination of the input sentences.

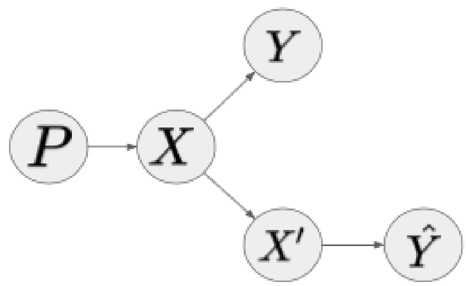


Figure 6.2: A graphical model of the proposed approach. P represents the protected attribute. X indicates the input explanations while X' indicates the farily-justified summary of X which is used to train the final model to predict outcome $\hat{Y}$.

Utility Control: To ensure that the extracted summary X' includes sufficient decision-relevant information in X, we measure the salience of each sentence in X in predicting outcome Y. We train a neural classification model on X using ground truth decision Y as supervision. Next, we use this model to derive the contribution of each sentence in X for predicting outcome $\hat{Y}$. This process is explained in §6.4.2. To ensure learning generalizable patterns, we hypothesize that the dataset is sufficiently large and the model can learn which factors are associated with which outcomes. This assumption especially holds for scenarios in which a decision-maker (e.g. an inspector or judge) is required to go through a standard set of criteria (e.g. a standard form or set of guidelines) and thus, the same arguments may repeatedly be articulated in different ways to justify a certain outcome.

Discrimination Control: To ensure that sentences in input explanation X that are biased toward certain protected groups are excluded from summary X', we attribute a discrimination score to each sentence in X. To do so, we measure the utility of an argument in identifying the membership of an individual i in the protected group P_i. Note that, we do not assume that "if you mention the protected attribute in the justification you are being unfair". Rather, methodologically, we remove the signals about the protected attribute to test whether the rest of the arguments in X' still sufficiently justify the outcome. Moreover, this is a way of demonstrating to the stakeholders that the model decision is not conditioned on the protected attribute. If removing the gender or race-coded language from the justifications does not change the predicted outcome, then we can conclude that the initial gender or race-coded language (that was removed) was not an unfair justification. To measure the discrimination score, we use justification X to predict protected attribute P. Next, we use the trained model to derive the contribution of each sentence in the membership identification task. Sentences with a high discrimination score are removed. We train a multi-task model for decision classification and membership identification tasks. This process is explained in the next section.

6.4.1 Model Architecture

Prior research has adopted word embeddings and Convolutional Neural Networks (CNN) for variety of sentence classification tasks [48, 104, 93, 100, 249, 92]. Kim [119] achieved strong empirical performance using static vectors and little hyper-parameter tuning over a range of benchmarks. Variations of this architecture have achieved good performance for extractive summarization of privacy policies [113] and court cases [250]. CNNs are fast to train and can easily be combined with methods such as Integrated Gradients [219] for attributing predictions to specific parts of the input. These considerations led to our decision to use a slight variant of the sentence-ngram CNN model in [250] for decision outcome prediction and membership identification tasks. Given explanation X_i consisting of m sentences/arguments $\{s_1, ...s_m\}$ to justify decision Y_i for individual i, we use Universal Sentence Encoder [35] to encode each sentence s_j to a 512-dimensional embedding vector v_j. We build the justification matrix $A \in R^{m \times 512}$ by concatenating the sentence vectors v_1 to v_m:

$$A_{1:m} = v_1 \oplus v_2 \oplus ...v_m$$

The Sentence Encoder is pre-trained using a variety of data sources and tasks [35] using the Transformer [226] architecture and is obtained from Tensorflow Hub. Following [48] we apply convolution filters to windows of sentences in explanation X_i to capture compounded and higher-order features. We use multiple filter sizes to capture various features from sentence n-grams. We use filter sizes of $h \times d$ where h is the height or region size of the filter and indicates the number of sentences that are considered jointly when applying the convolution filter. d is the dimensionality of the sentence vectors and is equal to 512. The feature map $c \in R^{m-h+1}$ of the convolution operation is then obtained by repeatedly applying the convolution filter w to a window of sentences $s_{j:j+h-1}$. Each element c_j in feature map $c = [c_1, c_2, ...c_{m-h+1}]$ is then obtained from:

$$c_i = f(w \ . \ A[j : j + h - 1] + b)$$

where $A[j : k]$ is the sub-matrix of A from row j to k corresponding to a window of sentence s_j to s_k and "." represents the dot product between the filter w and the sub-matrices. $b \in R$ represents the bias term and f is an activation function such as a rectified linear unit. We use window sizes 2, 3, and 4 and train 100 filters for each window size. The dimensionality of the feature map c generated by each convolution filter is different for explanations with various lengths and filters with different heights. We apply an average-max pooling operation over the feature maps of each window size to downsample them. Next, we concatenate the output vectors. Eventually, the concatenated vector runs through a dense layer with 64 units followed by an activation function[50]. This is a multi-task model with a decision learner and membership identifier modules. The decision learner is trained using decision outcome Y as supervision and the membership identifier is trained

[50]For classification tasks we used softmax (multi-class) or Sigmoid (binary classes) functions. For scalar outputs, we used Rectified Linear Unit.

using the protected attribute P. The loss at each epoch is computed based on a weighted sum of the decision prediction and membership identification losses. Training details are explained in Section 6.5.2. Next, we explain the method we use for attributing the predictions $\hat{Y}$ and $\hat{P}$ of the model to arguments in X.

6.4.2 Attribution

Sundararajan et al [219] proposed a method called Integrated Gradients to attribute predictions of a deep neural network to its input features. This method is independent of the specific neural architecture and can provide a measure of relevance for each feature by quantifying its impact on the predicted outcome. Zhong et al [250] adopted this method for identifying most decision-relevant aspects of legal cases. We also utilize this method to measure the impact of each input sentence in decision prediction and membership identification tasks. Essentially we take a straight line path from input x to its baseline b [51] and notice how model prediction changes along this path by integrating the gradients along the path. To approximate the integral of the integrated gradients, we simply sum up the gradients at points occurring at small intervals along the straight-line path from the baseline to the input. The resulting single scalar represents the gradients and attributes the prediction to input features. The integrated gradient along the i-th dimension for an input x and baseline b is defined as follows:

$$IG_i(x) ::= (x_i - b_i) \times \sum_{k=1}^{m} \frac{\partial F(b + \frac{k}{m} \times (x - b))}{\partial x_i} \times \frac{1}{m}$$

Here, $F : X \rightarrow Y$ represents the neural model, $\frac{\partial F(x)}{\partial x_i}$ is the gradient of F(X) along the i-th dimension, x represents the input at hand, b represents the baseline input (an all-zero vector), and m is the number of steps in the approximation of the integral[52]. To obtain utility attribution $U = \{u_1, u_2, ...u_m\}$ for sentences $\{s_1, s_2, ..., s_m\}$ in input justification X_i we calculate the attributions for the model using the predicted decision outcome $\hat{Y}$. Note that each input feature is one dimension of sentence embedding. To obtain salience scores for each sentence, we sum up the attribution scores for each dimension. Next, we run U through a softmax function to get a utility distribution over the sentences. Similarly, we obtain discrimination attribution $D = \{d_1, d_2, ...d_m\}$ for sentences $\{s_1, s_2, ..., s_m\}$ by calculating the integrated gradients attributions for the model using the predicted protected attribute $\hat{P}$. We run D through a softmax function to get a discrimination distribution over the sentences. We include high-utility and low-bias sentences in the fairly-justified summary of the explanations. The final inclusion score a_i for each sentence is computed using the following equation[53]:

[51] Conceptually, baselines represent data points that do not contain any useful information for the model. They are used as a benchmark by the integrated gradients method. Sundararajan et al [219] suggest using an all-zero input embedding vector for text-based networks.

[52] Sundararajan et al [219] applied Integrated Gradients to a variety of deep architectures including CNN. The only assumption that they make is that function F should be differentiable almost everywhere. Deep networks built out of Sigmoids, ReLUs, and pooling operators satisfy this condition.

[53] Both U and D satisfy properties of a probability distribution as $\Sigma_{i=1}^{m} u_i = 1$ and $0 \leq u_i \leq 1$. Thus, each u_i and d_i have comparable scales

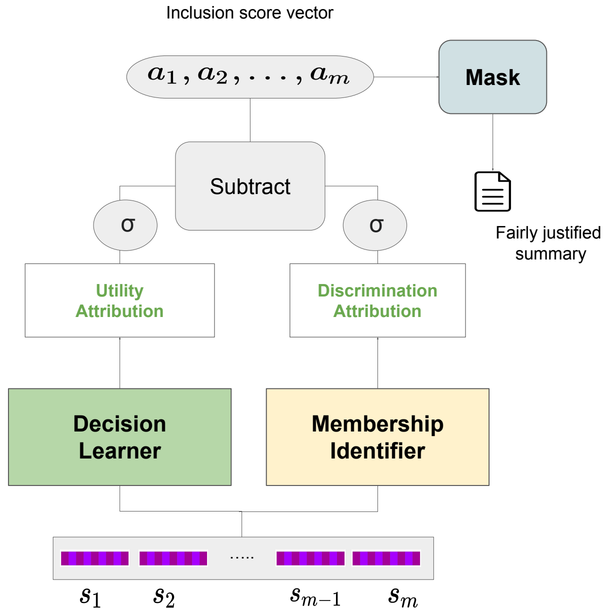


Figure 6.3: An overview of the architecture: Decision learner and membership identifier are trained using decision Y and protected attribute P as supervision respectively. The attributions of each module is normalized and subtracted to obtain the inclusion scores.

$$a_i = \sigma(u)_i - \alpha \times \sigma(d)_i$$

In the equation above, α is a hyper-parameter that controls the utility-discrimination trade-off. Higher values of α correspond to removing more information about the protected attribute from the input justification. Figure 6.3 shows the attribution process. Methodologically, we want to identify and remove arguments that are not useful for decision prediction or are only useful for the prediction of the outcome when they are also helping in the prediction of the protected attribute. The subtraction operation ensures that such arguments get a small inclusion score a_i.

Extracting Fairly-Justified Summarizes: Given sentences $\{s_1, s_2, ..., s_m\}$ and the corresponding inclusion scores $\{a_1, ...a_m\}$, we select sentences with a positive score for inclusion in the output summary. These sentences have high utility for decision prediction but do not reveal the protected attribute of the individuals. We refer to our preprocessing method as `FairSum`. In our experiments, we test whether training a decision classifier on justifications pre-processed by `FairSum` will enhance fairness in the justification on real-world and synthetic datasets.

6.5 Experiments and Results

In this section, we introduce the datasets we use for training and testing our model followed by training details, and metrics in consideration. In §6.5.4 we present our experimental results on two real-world datasets. In §6.5.5 we share our experiments involving several types of synthetic data poisoning attacks. Finally, in §6.5.6 we present a pilot user study that aims to examine the impact of applying `FairSum` on human subjects' decision-making.

6.5.1 Datasets

In this section, we introduce datasets used for training and evaluation of our model.

Inspection Reports of food establishments in Chicago (D1): The City of Chicago has published reports of food inspections conducted since 2010. We extracted the information on food inspections conducted from January 2010 till December 2014 from the City of Chicago's GitHub repository[54]. This dataset contains the outcome of inspection which can be *pass*, *fail*, or *conditional pass* as well as notes that the sanitarian left in the inspection form about the observed violations to justify the outcome and explain what needs to be fixed before the next visit[55]. In food inspections, decisions are being made for both the restaurant owner and the public health. In this work, we focus on fairness concerning customers of the food establishment. Thus, we consider the ethnicity of the majority of the population in the census block group where the food establishment is located as the protected attribute. This is a reasonable proxy given that Chicago is one of the most segregated cities in the US [49][56]. This dataset includes 17,212 inspection reports. The inspector's comments are on average 18.2 sentences long with a standard deviation of 7.2. The breakdown of the inspection outcome for each demographic group is shown in Table 6.1. Note that for the food establishments that have more violation, the inspection reports tend to be longer. In our summarization experiments, we focused on longer inspection reports which often includes establishments with higher number of violations. We train the model explained in §6.4.1 on inspector notes using inspection outcome and the ethnicity of the majority of the customers as supervision for decision classifier and membership identifier respectively. We use 90% of inspections from January 2010 till October 2013 (75% all records in our data-set) as our training set and the remaining 10% as our validation set. The inspections conducted from November 2013 utill December 2014 are used as our test set. We represent this dataset with D1.

Rate My Professor (D2-D4): Students can leave an anonymous review and rating on a scale of 1-5 in several categories for their instructors on the Rate My Professor (RMP) website. Previous work has identified several types of biases in students' evaluations [127, 197, 47, 22, 201, 224]. In our study, we aim to detect and remove potential biases in justifications provided by students to explain their ratings. We rely on the dataset collected by He et al [89]. We combine all the reviews written for each instructor and use the average rating as the supervision for the decision classifier. We use the gender of the instructor as the supervision for the membership identifier model. The

[54]https://github.com/Chicago/food-inspections-evaluation

[55]There could be other outcomes e.g. when the sanitarian could not access the establishment. These cases are excluded from our study.

[56]The demographic information of neighborhoods was extracted from https://www.census.gov/

Race	Pass	Conditional pass	Fail	Total inspection count
White	27.5	25.6	46.8	8339
Black	28.9	15.6	55.4	4444
Hispanic	33.8	19.2	46.8	4010
Asian	29.3	17.4	53.2	419

Table 6.1: The percentage of inspections for each ethnic group that received a pass, conditional pass, or a fail outcome.

rate my professor dataset only includes professor names and reviews. To infer the gender of the professors, we search for pronouns and titles commonly used for each gender[57]. If no pronouns or titles are found in the reviews, the professor's name is used to detect their gender [58].

In our experiments, we exclude the instructors that have less than 5 reviews. We also remove the pronouns and instructors' names from the reviews.[59] The resulting dataset includes reviews written for 1344 instructors which are on average 45.6 sentences long. We indicate this dataset with D2. The breakdown of reviews written for each gender category in D2 is shown in Table 6.2.

	[1,2]	**(2,3]**	**(3,4]**	**(4,5]**	**Total count**
Female	5.6	21.0	35.3	37.9	551
Male	3.7	21.0	35.6	39.5	783

Table 6.2: The percentage of instructors of each gender group in each rating class for dataset D2.

Prior work, has shown that using gender-coded language in teaching evaluations is more common in disciplines with a large gender-gap [217]. Inspired by this observation and to study the impact of the reviewer's gender (students) on teaching evaluations, we create two additional datasets D3 and D4. To do so, we split the RMP dataset based on the gender gap of the students in each discipline. D3 includes student evaluations for professors in fields that are female-dominant such as nursing, psychology, and education while D4 includes student evaluations for male-dominant majors such as engineering, computer science, and philosophy. Fields with less than 20% gender gap are excluded.[60] . For D2-D4, we randomly split our dataset into a 70-15-15 split to build our train, validation, and test sets. The breakdown of reviews written for each gender category for D3

[57]For sake of simplicity we assume binary and static gender classes

[58]We use https://pypi.org/project/gender-detector/ for mapping professors' names to their gender

[59]This pre-processing step ensures that the membership identifier does not rely on blatant signals from the text and instead extracts more latent patterns in the justifications.

[60]The statistics about the bachelor's degrees earned by field and gender is obtained from [182]

and D4 is shown in Tables 6.3 and 6.4. Next we will share our training details and hyperparameter setup.

	[1,2]	(2,3]	(3,4]	(4,5]	Total count
Female	4.3	22.2	31.5	41.9	279
Male	1.7	18.0	32.6	47.5	288

Table 6.3: The percentage of instructors of each gender group in each rating class for dataset D3.

	[1,2]	(2,3]	(3,4]	(4,5]	Total count
Female	5.5	24.4	39.3	30.7	127
Male	6.3	24.6	37.9	31.03	345

Table 6.4: The percentage of instructors of each gender group in each rating class for dataset D4.

6.5.2 Hyper-parameters and Training Details

Training Details: To train the model introduced in §6.4.1 on D1, We employ window sizes of 2, 3 and 4, and train 100 filters for each window size. For smaller datasets D2-D4. we use window sizes 2 and 3 and train 50 filters for each window size. We initialize each convolution layer using the initialization method proposed by He et al. [91]. We use rectified linear unit as the activation function of the convolution layer. After performing the convolution operation, we apply batch normalization [99] followed by a global average-pooling operation over the feature map of each window size.

Next, we concatenate the output vectors. Eventually, we run the concatenated vector through a dense layer with 64 units followed by an activation function. For decision classification and membership identification on D1, we used the softmax operation to obtain class probabilities. For D2-D4 we used rectified linear unit to obtain the output rating, and sigmoid to obtain gender class probabilities. We implement the decision classifier and member identifier networks using the Keras library [61]. We use weighted cross-entropy loss function for classification tasks and mean squared loss for regression tasks and learn the model parameters using Adam optimizer [120] with a learning rate of 0.001.

For D1, we set the maximum length of the arguments to the 70-th percentile of explanation lengths in our train set (18 sentences). Textual explanations that are longer than this are truncated while shorter ones are padded. For D2-D4, we set the maximum length of the arguments to the 70-th percentile of the review length in our train set (64 sentences). Reviews that are longer than this are truncated while shorter ones are padded. We set the loss weight for the decision prediction task and the membership identification task to 1. We train our multi-task network for a maximum of 25 epochs and stop the training if the decision classification loss on the validation set does not improve for 3 consecutive epochs. In the end, we revert the network's weights to those that achieved the

[61] https://keras.io

lowest validation loss. We repeat each experiment 5 times and report the average result. We used a single Nvidia Tesla K80 GPU for our experiments.

Parameters of the attribution Model: For computing the integrated gradients for attribution, we set the number of steps in the path integral approximation from the baseline to the input instance to 50 and use Gauss–Legendre quadrature method [1] for integral approximation. We compute the attributions of the decision classifier and the membership identification networks for the input layer.

6.5.3 Measuring Fairness of Justification and Outcome

In this section, we introduce the metrics we use for evaluating our pipeline. The current automatic evaluation protocol for automatic text summarization is based on the similarity of the model-generated summary to a human-written summary and using metrics such as ROUGE [133]. Shandilya et al [212] was the first to evaluate text summarization systems from the fairness perspective. They verify the fairness of summaries using the notion of *adverse impact* by measuring the fraction of selected tweets to be incorporated in the output summary from each protected group. The goal of our work (enhancing fairness in justification while preserving decision-relevant information) however, is different from traditional text summarization as well as the notion of fairness used by Shandilya et al. [212]. Thus, we use a new perspective for the evaluation of extracted summaries which is based on demographic leakage and fairness of outcome. Essentially, in our experiments we seek to answer the following questions:

- How does applying `FairSum` on the inputs impact the utility of the model?
- Will this pre-processing step effectively remove the proxy information about the protected attribute from the justifications?
- How does `FairSum` impact the fairness of the outcome?
- Is `FairSum` able to mitigate different types of unfairness attacks?

To answer the first question, we report the utility of the decision learner. For categorical outcomes (e.g. in D1) we report the Micro-F1 and Macro-F1 and for scalar outcomes (D2-D4) we report the Mean Absolute Error(MAE). To answer the second question, we report the demographic leakage. Leakage is defined as the ability of the membership identifier network to correctly predict the protected attribute of the individuals given the justification. We report the Micro-F1 and Macro-F1 of our membership identification model. Lower demographic leakage is desirable.

While `FairSum` is not directly designed to address the fairness of the outcome, we seek to study how enhancing fairness in justification impacts the fairness of outcome. To do so, for categorical outcomes we report the demographic parity, equality of odds, and calibration. For each of these metrics, we report the gap between the most favored and the least favored group. For a discussion on fairness measures and their trade-offs see the work by Kleinberg et al. [121] and Hardt et al. [86]. We additionally report False Pass Rate Gap (FPRG) and False Fail Rate Gap (FFRG) across demographic groups. FPRG and FFRG represent the equality in the distribution of the model errors across demographic groups. Similar metrics were used in [227]. We next formally define these metrics in the context of food inspection fairness. These metrics will be used to measure the fairness of outcome for dataset D1. Then, we share our experimental results.

Parity: a decision classifier satisfies demographic parity if the proportion of food establishments predicted to fail the inspection is the same for each demographic group. We report the gap between the most and least favored groups. For sake of consistency with previous work, we present the protected attribute with S.

$$\max(P(\hat{Y} = fail|S = s_i) - P(\hat{Y} = fail|S = s_j)) = \epsilon, \qquad s_i, s_j \in S$$

Equality of odds: for those establishments who actually failed the inspection, the proportion of failed predictions should be the same. We report the gap between the most and least favored groups. Ideally, the gap should be very close to zero.

$$\max(P(\hat{Y} = fail|Y = fail, S = s_i) - P(\hat{Y} = fail|Y = fail, S = s_j)) = \epsilon, \qquad s_i, s_j \in S$$

Calibration: for those establishments who received a fail prediction, the probability of actually failing the inspection should be the same. We report the gap between the most and least favored groups. Ideally, the gap should be very close to zero.

$$\max(P(Y = fail|\hat{Y} = fail, S = s_i) - P(Y = fail|\hat{Y} = fail, S = s_j)) = \epsilon, \qquad s_i, s_j \in S$$

False Pass Rate Gap(FPRG): food establishments that did not pass the inspection should have the same probability of falsely receiving a pass prediction. We report the gap between the most and least favored groups which ideally should be close to 0.

$$\max(P(\hat{Y} = pass|Y \neq pass, S = s_i) - P(\hat{Y} = pass|Y \neq pass, S = s_j)) = \epsilon, \qquad s_i, s_j \in S$$

False Fail Rate Gap(FFRG): establishments of different demographic groups that did not fail the inspection should have the same probability of falsely receiving a fail prediction. We report the gap between the most and least favored groups which ideally should be close to 0.

$$\max(P(\hat{Y} = fail|Y \neq fail, S = s_i) - P(\hat{Y} = fail|Y \neq fail, S = s_j)) = \epsilon, s_i, s_j \in S$$

To measure fairness for scalar outcomes (D2-D4), we report the Mean Absolute Error GAP between the demographic groups (male and female). Our experimental results on dataset D1-D4 are shared in Section 6.5.4

Dataset	Utility ↑ (Micro-F1)			Utility ↑ (Macro-F1)			Demographic Leakage ↓ (Micro-F1)			Demographic Leakage ↓ (Macro-F1)		
	Empty	Full	FairSum	Empty	Full	FairSum	Empty	Full	FairSum	Empty	Full	FairSum
D1	0.48	0.83	0.83	0.22	0.83	0.82	0.56	0.58	0.52	0.18	0.38	0.33

Table 6.5: Results on datasets D1. ”↑”: higher is better. ”↓”: lower is better.

Finally, to answer our last question, we perturb dataset D1 to create several types of injection attacks. We measure the attack success before and after applying `FairSum` on the perturbed data. Our findings are shared in §6.5.5.

6.5.4 Results and Discussion

In our experiments, we compare the utility, demographic leakage, and fairness of models that are identical in terms of architecture but are trained on different versions of the training data. The model architecture is discussed in §6.4.1. Our results on dataset D1 is shared in Table 6.5 and Table 6.6. Our Results on datasets D2-D4 is shared in Table 6.7. In the ”empty” setting, justifications are empty. In the ”full” setting, the model is trained and tested on the original data while in the `FairSum` setting it is trained and tested on justifications summarized by `FairSum`. We use to empty setting to indicate the lower bound of the demographic leakage. We use the full setting, to measure the bias in the justifications in the original dataset. This setting also acts as our baseline. We apply FairSum on both the train and test sets. The parameter α which controls the trade-off between the utility and the demographic leakage is set to 1.

As it can be seen in Table 6.5, `FairSum` reduces the demographic leakage on dataset D1 (by 0.06 in Micro-F1 and 0.05 in Macro-F1) while achieving the same level of accuracy on the decision classification task in comparison to the full setting. `FairSum` also decreases parity by 0.01 while achieving similar results in terms of FFRG and FPRG.

We see in Table 6.7 that on dataset D2, `FairSum` decreases the demographic leakage from 0.71 to 0.61 Micro-F1 and 0.69 to 0.58 Macro-F1 while increasing the MAE by 0.02 in a 5-point scale. `FairSum` outcomes also are more fair on D2. In the full setting, predictions have a 0.06 higher average MAE for females than males. While `FairSum` achieves similar error rates for both gender groups (0 MAE gap). On D3 and D4, fairSum reduces the demographic leakage (from 0.66 to 0.59 and 0.71 to 0.49 Macro-F1 respectively). `FairSum` is noticeably effective in removing the gender-coded language in D4 which is sourced from male-dominated majors with 0.82 gender prediction accuracy in the full setting. This comes with almost no change in the model's utility as the average MSE on D2-D4 is 0.51 for both the full setting and `FairSum`. We conclude that our proposed approach is very effective in reducing the demographic leakage in the input justifications while also not reducing the utility of the model. Removing gender-coded language from D3 justifications comes with the cost of having 0.06 higher MAE for females than males (this was 0.03 for the full setting). On D2 and D4 however, `FairSum` completely closes the MAE gap between the gender groups.

Dataset	Parity ↓		Equality of Odds ↓		Calibration ↓		FPRG ↓		FFRG ↓	
	Full	FairSum	Full	FairSum	Full	FairSum	Full	FairSum	Full	FairSum
D1	0.15	0.14	0.08	0.1	0.05	0.06	0.05	0.05	0.11	0.11

Table 6.6: Fairness metrics for datasets D1. ”↓”: lower is better.

Dataset	MAE ↓			Demographic Leakage ↓ (Micro-F1)			Demographic Leakage ↓ (Macro-F1)			MAE Gap ↓		
	Empty	Full	FairSum	Empty	Full	FairSum	Empty	Full	FairSum	Empty	Full	FairSum
D2	0.72	0.47	0.49	0.59	0.71	0.61	0.37	0.69	0.58	0.07	0.06	0
D3	0.76	0.52	0.53	0.5	0.66	0.61	0.33	0.66	0.59	0.19	0.03	0.06
D4	0.66	0.54	0.53	0.45	0.82	0.74	0.3	0.71	0.49	0.04	0.02	0

Table 6.7: Results on RMP Datasets (D2-D4). ”↓”: lower is better.

An example of applying `FairSum` on a teaching evaluation for a two professors is shown in Figure 6.4 and Figure 6.5. In Figure 6.4, we see that arguments about the looks of the instructor (more frequent for female instructors) are excluded from the text (indicated with orange). The preserved sentences are indicated with purple and have a high inclusion score. In Figure 6.5, arguments about being ”intelligent and funny” (more frequent for male instructors) are removed from x by `FairSum`. While mentioning ”intelligence” is not an unfair argument on its own, more frequent usage for a certain demographic group makes it a gender-coded justification.

Utility-Fairness Trade-Off: Figure 6.6 shows the utility, demographic leakage, and fairness metrics as a function of α on D1 and D2. Too low values of α prioritize utility, selecting even relatively biased sentences and have scores close to the full setting (see Figure 6.6 a and 6.6 c). On D1, increasing α generally decreases the demographic parity while increasing the FPRG (see Figure 6.6 b). It does not have a consistent or noticeable impact on other fairness metrics. On D2 and with α near 1, the gap shrinks to 0 (See Figure 6.6 c). Too high values of α remove too many sentences, leading to a high error rate. This is because many summaries are empty with a high value for α and thus, the resulting decisions are unjustified (justifications are not informative about the outcomes) and unfair (the lack of justification is not uniformly distributed over genders) so the gap emerges once again.

Impact of α on summary length: Figure 6.7 shows the average summary length (sentence count) for datasets D1-D4 as a function of α. The food inspection reports in D1 are on average much shorter than the teaching evaluations in dataset D2 (18.2 vs 45.6 sentences). Too low values of α prioritize utility by preserving even relatively biased sentences. For all datasets, the summaries start shrinking around α equal to 0.85. However, for D2-D4 the compression rate is higher. Around α equal to 1.25, 39.9% input justifications for dataset D1 are empty. This number is 77.8%, 95.7%, 100% for D2, D3, and D4 respectively. We conjecture that the existence of more implicit bias for D2-D4 causes the summaries to shrink faster by increasing α. At this point (1.25 and higher) the resulting decisions are unjustified (justifications are not informative about the outcomes). Therefore in Figure 6.7 we only show impact of changing α from 0.8 to 1.2.

Example 1: Not my favorite instructor. We spent a lot of time on things that seemed not important. Course syllabus included a lot of topics that have no practical use. Some days the presentations were unclear but I would recommend this course to non-majors. Very open and well organized. The guys in the class love . is a pretty good , but not a great teacher. is a great professor also has the physical features that makes you not want to miss a class. Last semester came to class is a short skirt omg! has a lot of experience with undergrad students. Sometimes vague on grading criteria . This is a pretty easy class. – is very nice. is hot and funny. If you get past physical attributes you really learn something. Wow! ... what an interesting topic! I respect for intelligence and ability to teach , not for appearance. Very good looking omg! Great course and additional materials are a great support.

Figure 6.4: Applying `FairSum` on teaching evaluations for a female professor (anonymized and paraphrased for privacy considerations). The pronouns and names have been removed before model training and attribution. Sentences with a positive attribution score (purple) are preserved in the summary x' while the sentences with a negative attribution score (orange) are excluded.

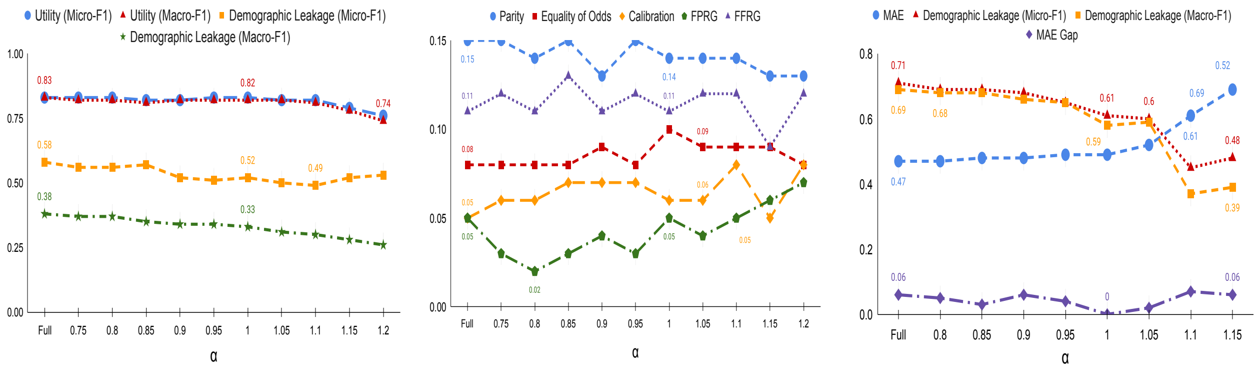
Example 2: Lectures are short. Tests do not really cover what is covered in class. Textbook is not used. Dr. is very knowledgeable and passionate about this subject. You will enjoy the class if you are interested in the topic. Highly recommend if you want a nice grade. is funny , intelligent, and easy to listen to. got an epic beard. post the material online which makes the class very accessible. If you do all the assignments it is impossible to not get an A! is one of very few whom I really think understands the "real world" and its workings. I think it is because of days in navy. is very funny as well. curves quizzes slightly. So, in the end your grade could be better than what you may think. can be very helpful , but you must go to the office hours. Probably the easiest five credit class you can take.

Figure 6.5: Applying `FairSum` on teaching evaluations for a male professor (anonymized and paraphrased for privacy considerations). Sentences with a positive attribution score (purple) are preserved in the summary x' while the sentences with a negative attribution score (orange) are excluded.

6.5.5 Unfairness Attacks

Natural language processing models are vulnerable to test-time adversarial attacks [228]. These attacks often are created to cause the model to make errors by perturbing the input at inference time [228]. In this section, we present our experimental results to test the ability of our model in detecting and counteracting data poisoning attacks. Essentially, we seek to answer the following questions:

- Can `FairSum` detect the injected unfair arguments in the justifications?
- Given that `FairSum` relies on attributing the decision outcome to input arguments, how does it perform in a scenario where decision outcomes are not fair in the first place?



(a) (b) (c)

Figure 6.6: Impact of α on utility and fairness on datasets D1 (a and b) and D2 (c).

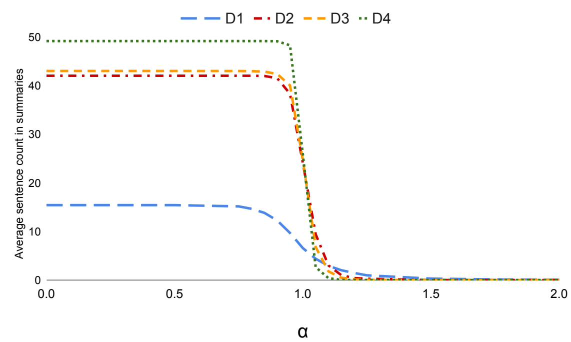


Figure 6.7: Impact of α on summary length on datasets D1-D4.

To answer these questions, we synthesize several unfair decision-making situations. In each of these situations, we create poisoning attacks to synthesize an unfair justification, an unfair outcome, or both. These scenarios are indicated with red in Figure 6.1 in the begining of this chapter. To evaluate the robustness of `FairSum`, we measure the attack success before and after applying `FairSum` on the poisoned train and test data. For our experiments, we rely on the food inspection dataset (D1). We assume that the ground truth outcomes Y in this dataset are often fair [62]. To synthesize an *unfair outcome* for establishment i we simply flip outcome y_i (e.g. by changing pass to fail). *Unfair justifications* can be created in more than one way such as synthesizing decision-making with double standards or implicit bias. Inspired by the work of Wallace et al [229] we create unfair justification attacks by causing a phrase to be a trigger for a desired outcome by poisoning the training data. For example, we could make the phrase ”kitchen manager does not speak English” to trigger the model to predict the food establishment should fail by adding this phrase to enough inspection reports of establishments that failed the inspection. We create an irrelevant justification set including phrases

[62] We cannot make this assumption about the RMP dataset as evaluations are very subjective. Therefore, we do not experiment with this dataset.

about the decoration of the restaurants e.g. *"walls are red"*, *"table cloths are blue"*. None of these phrases are part of the inspection guideline of the city of Chicago. For creating irrelevant justifications, we randomly select an argument from this set and add it to a random position in the inspection report. In all our experiments, we use the entire training data of D1 and a certain subset of the test set depending on the type of attack. Next, we introduce 5 common unfairness scenarios that inspired our experiments as well as the experimental setup in each attack type.

Attack type 1a: Deciding based on items not in the guideline: In this attack, we create a scenario where decision outcomes are fair but the justifications are unfair. To do so, we inject irrelevant arguments into the reports. We use these arguments more frequently for a certain demographic group than others. However, we do not alter the ground truth outcomes Y. A real-world example of this attack is a food inspector who mentions *"the kitchen manager does not speak English"* for several Hispanic restaurants. They write this in reports often when they are frustrated with explaining the hygiene guidelines that were not followed by the establishment to the kitchen manager, resulting in the food establishment failing the inspection. While this is indeed a fair outcome, training a model on such reports can have two undesired side effects. First, the model can wrongly associate the irrelevant argument "not speaking English" with the fail outcome. Thus, results in the model predicting "fail" when "not speaking English" is mentioned in the reports even when all hygiene guidelines are followed. In addition, this model has information about individuals' ethnicity due to the race-coded language of the reports.

To create a poisoned train dataset, we randomly select an item from the irrelevant argument set and add it to a random sample of $a\%$ of individuals in the demographic group p_i that received outcome y_j, trying to trigger the model that e.g. "table cloths are blue" will result in outcome y_j. Especially when the individual belongs to p_i. At test time, we pick K restaurants in p_i, half of which with ground truth outcome y_j and half with other outcomes. We inject an irrelevant argument into the inspection reports of this set. To measure attack success, we measure the false y_j prediction rate as well as the demographic leakage before and after applying `FairSum` to the poisoned test data.

Attack type 1b: Race-coded language: In this attack, decision outcomes are fair but the justifications are unfair. However, in contrast to the attack type $1a$, the irrelevant race or gender-coded language does not impact the outcome. An example is an inspector who reports the address of the restaurant in the inspection reports in Hispanic restaurants irrespective of the outcome. Training a model on such data can be problematic in two ways. First, the model may still associate location with outcome y_j if the majority of the restaurants in that neighborhood have received outcome y_j. Moreover, even if the model does not learn such an association, it might be able to predict the ethnicity of the customers based on the location due to the demographic leakage in the data. In this type of attack, we create the latter problem. To create poisoned training data, we inject an irrelevant argument to $a\%$ of food establishments in demographic group p_i. To de-correlate the injected argument with any outcomes, we make sure that the number of attacked restaurants with each outcome is the same (e.g. 50% pass, 50% fail). At test time, we pick K restaurants from protected group p_i and inject an irrelevant argument to the reports of this set. The measure of attack success, we measure the demographic leakage before and after applying `FairSum` to the data.

Attack type 2a: Implicit bias: In this attack, decision outcomes are unfair but the justifications are fair. In this scenario, a fair justification that is part of the guideline is only used for a specific protected group and impacts the outcome of their inspection. For example, a food inspector only mentions "food prep hygiene violations" when the restaurant is located in a majority-Asian neighborhood. This violation leads to establishments failing the inspection in this neighborhood. While "food prep hygiene violations" can be a fair reason for deciding that a restaurant should fail the inspection, using this argument only for restaurants in Asian neighborhoods is a case of race-coded language. Therefore, training a model on this data may result in the following two issues: (i) the model is race-aware and (ii) this phrase could become a trigger for "fail" prediction, even when the rest of the report justifies another outcome. To create poisoned training data, we select $a\%$ of food establishments in demographic group p_i that did not receive outcome y_j. For example, if $y_j = fail$, we select restaurants from p_i that either passed or conditionally passed the inspection. To create unfair outcomes, we flip the ground truth outcome of these restaurants. We randomly sample a fair argument for receiving outcome y_j from the guideline and inject it into these reports. Since the arguments are part of the guideline, they are fair. At test time, we pick K restaurants from the protected group p_i who did not receive outcome y_j. We inject a fair argument for receiving y_j to their reports. We measure the attack success by measuring the false y_j prediction rate as well as the demographic leakage.

Attack type 2b: Double standard: In this attack, decision outcomes are unfair but the justifications are fair. In this scenario, a fair justification that is part of the guideline is mentioned in the inspection reports of restaurants in several neighborhoods. However, it only impacts the outcome when the restaurant is in p_i. For example, a food inspector mentions "food prep hygiene violations" for multiple restaurants but only decides that this is a serious threat to public health in Latino neighborhoods. Training a model on this data may result in several issues. First, the model is race-aware. In addition, the model uses the same arguments differently for different protected groups; putting a lot of attention to an argument for some individuals and ignoring for others based on their location. To create poisoned training data, we follow the same process as in attack type 2a. The only difference is that we choose half of the attacked restaurants from p_i and half, not in p_i. We only flip the outcome to y_j for those who are in p_i. At test time, we pick K restaurants half of them from protected group p_i and half, not in p_i who did not receive outcome y_j. We inject a fair argument for receiving y_j to their reports. We measure the attack success by measuring the false y_j prediction rate for both groups as well as the demographic leakage for establishments in p_i.

Attack Type 3: Blatant bias: In this attack, both the outcome and the justification of the outcome are unfair. In this scenario, an argument that is not part of the guideline (and therefore unfair) is used to justify an unfair outcome. Using this data for training may result in the model being race-aware. Moreover, the model may wrongly associate an irrational argument with a certain outcome. To create a poisoned train dataset, we randomly sample an argument from the irrelevant argument set and add it to a random sample of $a\%$ of restaurants in p_i that did not receive outcome y_j. Then we flip their outcome to y_j. At test time, we do the same for K restaurants in p_i without changing the outcome. We measure the attack success by measuring false y_j prediction rate as well as demographic leakage in the test set. The summary of the 5 attack types is shown in Table 6.8.

Note that `FairSum` is not intended to address fairness in the outcome (attack types 2a and 2b). Our motivation for exploring these scenarios is to investigate `FairSum`'s behavior in these situations and highlight some of its' limitations. In our experiments, we choose the restaurants in the majority-black neighborhoods that fail the inspection as our target group (p_i = black, y_j = fail). We randomly sample individuals to attack from this pool. We choose this group because it is large enough for us to test various attack scenarios. We change the percentage of the attacked population in this group from 20 to 80 percent with increments of 30%. In all our experiments, we use a

Attack ID	Description	Outcome	Justification
1a	Deciding based on items not in guideline	Fair	Unfair
1b	Race-coded language	Fair	Unfair
2a	Implicit bias	Unfair	Fair
2b	Double standard	Unfair	Fair
3	Blatant bias	Unfair	Unfair

Table 6.8: Summary of the 5 data poisoning attacks

Attack ID	FFR ↓		Demographic ↓ Leakage		FFR ↓		Demographic ↓ Leakage		FFR ↓		Demographic ↓ Leakage	
	Full	FairSum	Full	FairSum	Full	FairSum	Full	FairSum	Full	FairSum	Full	FairSum
1a	0.60	0.47	0.79	0.45	0.86	0.79	0.97	0.88	0.82	0.73	0.97	0.72
1b	[redacted]		0.91	0.51	[redacted]		0.90	0.25	0.90	0.25	0.90	0.36
2a	0.62	0.64	0.69	0.81	0.95	0.92	0.91	0.91	0.96	0.73	0.83	0.64
2b	0.36	0.39	0.44	0.59	0.45	0.41	0.65	0.42	0.59	0.52	0.5	0.53
3	0.96	0.83	0.82	0.82	0.96	0.91	0.93	0.75	0.96	0.85	0.93	0.7
attack rate = 0.2					attack rate = 0.5				attack rate = 0.8			

Table 6.9: Experimental results with 5 types of poisoning attacks. Attack success is measured by measuring false fail rate as well as demographic leakage before (indicated with Full) and after applying FairSum. ↓: lower is better.

poisoned test size (k) of 200. The attacked test set is sampled from different population groups in the original test set depending on the attack type. We repeat each attack experiment 5 times and report the average attack success before and after applying `FairSum` in Table 6.9. Depending on the type of attack, the attack success is measured by the demographic leakage or false fail rate (FFR) over the attacked test set. As it can be seen in the table, `FairSum` is very effective in decreasing the attack success for attack types 1a and 1b (fair outcome and unfair justification). When 20% of the individuals in the target population group are attacked, `FairSum` decreases the FFR by 0.13 points, while decreasing the demographic leakage by 0.34 points. For attack type 1b, using `FairSum` decreases the demographic leakage by 0.4 when 20% of the target subgroup are attacked. This number is 0.65 when half of this subgroup are targeted at train time (0.9 vs 0.25 demographic leakage after using `FairSum`). As expected, when outcomes are unfair the effectiveness of `FairSum` becomes limited at lower attack rates. This is mostly because the model learns wrong associations as the outcomes are flipped. This observation suggests that `FairSum` is an effective mechanism for enhancing fairness in justification, however, it is not very effective when outcomes are unfair. `FairSum` is moderately effective when both outcomes and justifications are unfair (attack type 3). For attack type 3, using `FairSum` decreases the demographic leakage by 0.18 on average when half of the target subgroup are poisoned. It also decreases the FFR by 0.13, 0.05, and 0.11 for attack rates 0.2, 0.5, and 0.8 respectively.

6.5.6 Pilot user study

In this section, we present a pilot user study to assess the impact of `FairSum` on users' performance in a real decision-making scenario. More specifically, we consider a scenario where each user has to guess the gender (female/male) and average rating (low/high) of the instructors given the summary of their teaching reviews extracted by `FairSum`. In addition to `FairSum` summaries, we consider two extreme settings: (a) the *full* setting where the raw teaching evaluation is shown to the user, and (b) the *excluded* setting, where participants only see the parts of reviews that are identified as potentially biased by `FairSum` and therefore are excluded from summaries. We add this baseline to verify whether sentences removed by `FairSum` had a high demographic leakage or low utility for rating estimation.

Study Design: We randomly selected 20 instructors from the RMP dataset; 10 men and 10 women. Half of the male professors have a high rating (above 3) while the rest have a low rating (3 and below). The same situation holds for female instructors. We removed all pronouns and professor names from all the teaching reviews so that users rely on gender-coded language for guessing the gender rather than more blatant signals. We applied `FairSum` to the teaching reviews to decide whether to keep or remove the sentences. We invited 14 graduate students to take part in our study. Each participant evaluated 60 review snippets (20 instructors $\times$ 3 presentation forms). To prevent information retention, reviews for different instructors were shuffled. We recorded user responses for gender and rating prediction. In this pilot study, we seek to answer the following questions: (i) *Does summarizing the reviews by `FairSum` impact the performance of users in the rating prediction task?* (ii) *How likely are users to attribute a positive/negative teaching review preprocessed by `FairSum` to a male/female professor?* (iii) *Does `FairSum` correctly identify and exclude the gender-coded language in teaching reviews?*

Results: The performance of users in rating prediction task as well as inter-annotator agreement [76] is shared in Table 6.10. As it is presented in the Table, using summaries extracted by `FairSum` users can guess the ratings as accurately as the full setting. Users also achieve 68.8% accuracy only by looking at the excluded text. Since removing these snippets from the raw teaching evaluation did not result in a drop in accuracy, we conjecture that most of the insights from the removed text could be inferred only from the fair summaries and therefore the excluded sentences were either redundant or not useful for completing the task [63].

	Accuracy	Annotator Agreement
Excluded	68.8	54.0
Full	75.0	51.3
FairSum	75.5	52.9

Table 6.10: Accuracy of users in rating prediction task and inter-annotator agreement (Fleiss Kappa) given different subsets of the teaching reviews.

[63] In this pilot study we only tested applying `FairSum` with parameter α equal to 1, it is interesting to study the impact of summary length (information load) on the performance of users in other decision-making tasks.

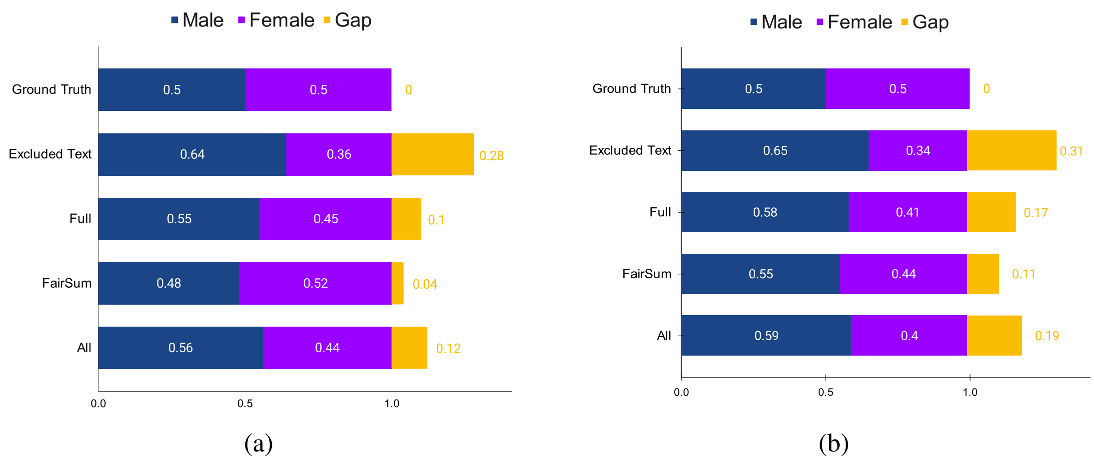


Figure 6.8: Distribution of the predicted gender in (a) positive reviews and (b) negative reviews.

Next, we look at the performance of users in gender prediction task. Figure 6.8 shows the distribution of the predicted gender given teaching evaluation of high-rated and low-rated instructors. As mentioned before, 5 of the 10 high-rated instructors in our sample dataset are female and 5 are male (ground truth). As it can be seen in the Figure 6.8 (a), given the excluded text from high-rated reviews, users guess that the instructor is male in 64% of the cases. The predicted gender distribution in the full setting is closer to the ground truth (0.55% male and 45% female). This indicates that the excluded text indeed contains gender-coded language. In addition, it can be seen that the fairly-justified summaries further close the gender prediction gap in comparison to the full setting (0.04 vs 0.10).

Figure 6.8 (b) shows that given the excluded text from the low-rated reviews, users are much more likely to guess that the instructor is male (percentage gap is 0.31). The gender gap for the full setting is 0.17. Again we can see that `FairSum` reduces the percentage gap to 0.11. Our preliminary pilot study shows that `FairSum` is an effective pre-processing approach for extracting justification summaries that are high-utility and gender-neutral.

6.6 Conclusion

In this chapter, we propose `FairSum` which is a train-attribute-mask pipeline for detecting and mitigating the bias in the justification of the text-based neural models. Our objective for extracting fairly-justified summaries is to maximize the utility of the output summary for the decision prediction task while minimizing the inclusion of proxy information in the summary that can reveal sensitive attributes of individuals. `FairSum` is not intended to enhance the fairness in the outcome but rather to enhance the fairness in the model justification. We achieve this by training a multi-task model for decision classification and membership identification. We attribute predictions of these models back to textual input attributes using an attribution mechanism called integrated gradients. Next, we incorporate the high-utility and low-bias sentences in form of a summary. Eventually, we retrain the decision classifier on the fairly-justified summaries. Our experiments on real and synthetic data sets indicate that our pipeline effectively limits the demographic leakage from the input data. In addition, we present experimental results on effectiveness of `FairSum` under several types of unfairness attacks. We observe that `FairSum` is most effective in detecting and filtering unfairness in justification where outcomes are mostly fair.

Chapter 7: Conclusions, Limitations and Future Work

In this proposal, we explored challenges and opportunities that arise when the summarization process is guided by a control aspect. We specifically, looked at control aspects such as length of the summary (§2), domain-specific information (§3), user query(§4), format(§5), and fairness objectives (§6). We showed generic summarization models are insufficient for generating or extracting controllable summaries in low-resource domains. In this chapter, we first provide a summary of our key contributions and then discuss limitations of our work and future research directions.

7.1 Our Key Contributions

Abstractive Summarization at Controllable Lengths [207]

In Chapter 2, we looked at constraining the summarization process by user preference in length (expressed in terms of number of tokens or compression rate). We proposed `MLS`, a supervised approach to construct abstractive summaries at controllable lengths. Following an extract-then-compress paradigm, we developed the Pointer-Magnifier network – a length-aware, encoder-decoder network that constructs length-constrained summaries by shortening or expanding a prototype summary inferred from the document. The key enabler of this network is an array of semantic kernels with clearly defined human-interpretable syntactic/semantic roles in constructing the summary given a budget-length. We trained our network on limited training samples from two cross-domain datasets. We presented exhaustive experiments on two low-resource datasets in English language and showed that `MLS` outperforms strong baselines by up to 14.70% in the METEOR score. Human evaluation of the summaries also suggest that summaries generated by `MLS` capture the key concepts of the document at various length-budgets.

Domain-guided summarization of privacy policies [113]

In Chapter 3 we first we presented a user study on the impact of presentation format on policy comprehension. We noticed that a format in which the riskier segments are highlighted leads to better comprehension of the policies. Motivated by findings of this study and to further address the information overload problem is this domain, in Chapter 3, we proposed a pipeline for extractive summarization of privacy policies. Our pipeline includes a risk prediction and a redundancy reduction module. We employ a pre-trained encoder and convolutional neural network to classify sentences of the contracts into different risk levels. To address the limitations of previous work, we incorporate the domain information predicted by the classifier in the form of a summary by using two content selection mechanisms– risk-focused and a coverage-focused. The coverage-focused selection mechanism aims to reduce information redundancy by covering the riskiest sentence from each privacy topic. Our approach enables users to select the content to be summarized within a controllable length while relying on substantially less training data than the existing supervised summarization methods. Despite the moderate success in classifying our realistically imbalanced dataset, we observed a noticeable improvement in ROUGE and METEOR metrics compared to domain agnostic baselines.

To address issues posed by lack of training data in this domain, we augmented the TOS;DR dataset and automatically annotated 151 privacy policies using only a few hundreds of user annotations. To design our pipeline, we used an the auxiliary task of risk classification in addition to unsupervised clustering methods for redundancy reduction instead of directly learning to summarize. The classifier is substantially less expensive to train but enables our approach to outperform a selection of domain-agnostic unsupervised summarization methods.

Automated Privacy Policy Question Answering Assistant [116]

To facilitate a more personalized interaction with the policies and make them more accessible to lay users, in Chapter 4, we proposed an automated question answering pipeline that extracts relevant segments from the policy in response to an input user query. We address two main challenges in this domain: (i) the difference between the language of user queries and the legal language of the privacy policies and (ii) low training resources.

To address the first challenge, we explored using familiar methods such as unsupervised lexical substitution and back-translation. We tried to close the domain gap between training corpus of our word vectors and legal language of policies by fine-tuning our word vectors on in-domain privacy policy data. We observed that while using these method can increase the coverage of user queries that we can answer, more in-domain data is needed to build paraphrase techniques to adapt the specificity and style of the user queries to privacy policies.

In our pipeline, following the query expansion, we used a content scoring module that relies on transfer learning and existing in-domain data to find relevant information in the policy. Essentially, we used Legal Bert which is a language model trained on in-domain data. Next, we fine-tuned this model for our task using both task-specific and auxiliary task training data. Our pipeline can find an answer for 87.7% of the user queries in the privacyQA dataset.

Low-resource Data-to-Text Generation Using Pretrained Language Models [117]

In Chapter 5, we looked at data-to-text generation. We investigated the performance of two pretrained language models on this task by evaluating them on two divisions of the DART test set: novel predicates(not seen during training) and abstractive examples. We examined their performance based on choice of the adaptation mechanism: fine-tuning, prompt tuning, and few-shot learning. We showed that the performance of fine-tuned T5 drops significantly on unseen predicates. On the other hand, the performance of few-shot GPT2-XL on unseen predicates can be enhanced even with shots containing unrelated predicates. We also notice that T5 and GPT2-XL both do well at D2T by copying the input. However, they do noticeably worse on examples where significant re-writing is needed. Adding domain knowledge (predicate descriptions) to the prompts can improve the performance of few-shot GPT2-XL on this subset by a large amount. We also conducted a human evaluation of the generations and find that prompt tuned GPT2-XL generations can be improved by re-ranking generations by overlap with the input entity spans.

Fairness aware Summarization for Justified Decision Making [115]

In Chapter 6 we formally defined the notion of fairness in the justification. We proposed `FairSum` which is a train-attribute-mask pipeline for detecting and mitigating the bias in the textual justification. `FairSum` is a novel application for text summarization and intends to enhance the fairness in the justification. Moreover, unlike many other preprocessing approaches in the fairness literature that produce an intermediate data representation that is not interpretable [82, 244], `FairSum`'s output is an extractive natural language summary of the input justifications. This is preferable for many applications where interpretability is essential.

Our objective for extracting fairly-justified summaries is to maximize the utility of the output summary for the decision prediction task while minimizing the inclusion of proxy information in the summary that can reveal sensitive attributes of individuals. We achieve this by training a multi-task model for decision classification and membership identification. We attribute predictions of the model back to textual inputs using integrated gradients. Next, we incorporate the high-utility and low-bias sentences in the form of a summary. Eventually, we retrain the decision classifier on the fairly-justified summaries.

Our experiments on food inspection reports by the city of Chicago and teaching evaluations on RateMyProfessor indicate that our pipeline effectively limits the demographic leakage from the input data. In addition, we present experimental results on the effectiveness of `FairSum` under several types of unfairness attacks. Since `FairSum` relies on attributing outcomes to input justifications, it is most effective in detecting and filtering unfairness in justification where outcomes are mostly fair. Moreover, results from our pilot user study indicates that our model is an effective pre-processing approach for neutralizing the textual data (in terms of the protected attribute) while preserving decision-relevant information.

7.2 Limitations and Future Work

An important objective of this dissertation is to redefine and constrain the automated text summarization task based on users' preferences in length, format, and focus of the extracted summary. Another long-term research goal of this dissertation is to work towards making automated text summarization more accessible in low-resource domains. Throughout this thesis, we attempted to address challenges introduced by the lack of training resources by developing methods and pipelines that effectively utilize limited or no human annotation during training and development. Both of these challenges–lack of task-specific labeled data and the need for control aspects–are crucial in many real-world applications of text summarization and in fields such as law, finance, and medicine.

We extensively studied creating information retrieval and summarization tool for legal contracts. We investigated the challenges posed to these automated systems due to users communication style and comprehension barriers. We also looked at emerging field of ethics and fairness of natural language processing systems. We investigated a novel application of text summarization for detecting and removing biases in the input data. In this last section, we discuss some limitations of our prior contributions and point out some promising directions for future research.

7.2.1 Controllable and Low-Resource Summarization

In this dissertation, we proposed end-to-end architectures and domain-inspired pipelines to constrain the summarization process based on user query or their length preference. An interesting future direction is to construct entity-focused or task-driven summaries. Personalizing a summary based on the user's past interaction is another exciting direction for future work.

Controlling the generated summary's style (e.g., plain English, narrative, descriptive, formal, etc.) is another exciting direction for controllable summarization and has many applications. For example, summarization of medical documents for users with different education levels/ backgrounds or news generation for non-native speakers.

Future work in low-resource text summarization should include more challenging and disparate domains, such as finance or medicine. It would be interesting to investigate how models trained on a high-resource domain (e.g., news) can be efficiently adapted to perform summarization on another unrelated and low-resource domain (e.g., medicine). This would require creating domain-specific datasets for summarization.

Text summarization research should also include low-resource languages that are less studied and less computerized due to data-scarcity.

7.2.2 Legal Text Summarization and Information Retrieval

Our user study presented in Chapter 3 showed that shorter presentation forms might be preferable for users but they do not always cover all the necessary information for a conscious agreement. However, our user study was different from the real-world interaction with policies in a few ways. First, users were informed about a quiz at the end and therefore were mildly motivated to read the long policies. Moreover, previous research by Obar and Oeldorf-Hirsch [173] also indicates that user's do not often read or understand the policies. Shorter presentation formats can be used to motivate more users to read the policy and assist them in finding the section they would like to focus on. An interesting direction for future research is combining the shorter and longer presentation forms for satisfying the information need of users.

Our prior work discussed in Chapter 3 and Chapter 4, has explored extractive and query-guided summarization of privacy contracts. While using automated methods to distill the information in the policies, makes contracts more accessible to lay users, summaries and answers extracted from the policy are still incomprehensible for many users as they are in legal language.

Summaries would be more accessible if written in plain English rather than legalese [173]. An abstractive system could be used to rewrite the contract text in this way. However, the abstractive summaries/answers should not change the legal interpretation of the content and should reliably reflect the relevant information in the source policy.

This issue is related to the challenge of trust in automated text summarization and is even a bigger challenge in legal domain. If users are to trust that the summary is indeed a reliable condensed version of the policy, each summary segment should be link-able to the original content to be considered binding. However, substituting the model generated summary with a full contract is not sensible in many domains. In sensitive applications, summaries can be used to assist users in reading the policies by (i) helping them decide what parts of the policy are more important or relevant to them for further reading or (ii) re-phrasing and simplifying segments that are hard to understand.

Another useful future direction is to use the risk classifier introduced in Chapter 3, independently to enhance the productivity of annotators by identifying the sections that need to be summarized. This can potentially facilitate annotating larger resources for training abstractive models.

Last but not the least, the evaluation of automated approaches such as ours using summarization or information retrieval metrics is insufficient to determine user comprehension of policies in practice. It is also necessary to conduct more extensive evaluation experiments, involving human readers as well as automated metrics. This will help determine the most effective ways to present information from click-through contracts so that users can understand the terms and make a more informed decision.

7.2.3 Data-to-text Generation

Future work in D2T generation should consider more challenging examples, and should consider ways in which to generate more diverse variations for expressing a given predicate. This should include more challenging and disparate domains, such as finance or medicine. In these cases, one may see benefits from including predicate descriptions, which performed well on the most abstractive examples.

An important challenge for D2T is how to train models that can generalize to new domains. While this work looked at a related class of examples (instances with unseen predicates), it would be interesting to investigate how PLMs trained on one domain can be efficiently adapted to perform D2T on another unrelated domain (e.g., sports to finance). This would require creating domain-specific datasets for D2T.

Moreover, we observed that adding domain knowledge (predicate descriptions) to prompts can improve the performance of few-shot GPT2-XL on abstractive examples. We suspect that this idea may work better on specialized domains, with better relation descriptions, or with a larger language model; we could not test this without a specialized D2T dataset with better task relation descriptions.

Finally, many applications prefer generating novel or interesting descriptions for a data record over "safe" and "generic" ones, which are predominant in DART [131, 132, 9, 213]. Evaluating PLMs for diversity of generated text is an orthogonal and promising future direction.

7.2.4 Fairness of Natural Language Processing Models

In Chapter 6, we discussed the notion of (un)fairness in justification and proposed a pre-processing tool based on text summarization called `FairSum`. `FairSum` is a train-attribute-mask pipeline for detecting and mitigating the bias in the textual justification that works at the sentence level. While this method can be used to remove unfair justifications for fair outcomes from training data of text-based neural models, it can also be abused to hide actual discrimination from stakeholders by removing an unfair justification for unfair decisions. While we see the potential for abusing this technology, we want to emphasize that we made certain assumptions about the training data for `FairSum`. Essentially, we assumed that the majority of the decisions in the training data of `FairSum` should have a fair outcome. In addition, our experiments show that `FairSum` does not reliably remove the race-coded or gender-code language when the outcome of the decision is not fair in the first place.

We see several interesting avenues for future research in the intersection of natural language processing and fairness. An immediate extension of our work is an enhancement approach that works at word-level for example, by using text generation and paraphrasing instead of sentence extraction.

Another interesting research direction is using this train-attribute-mask pipeline for removing bias from data of other natural language processing tasks such as sentiment analysis and using other architectures that can work with integrated gradients. It is also interesting to see how this solution can be extended to obfuscate biases for other data types (e.g. images or tabular data). Lastly, conducting a larger user study to evaluate the impact of this tool in explaining predictions for stakeholders in a real-world application is left for future work.

Bibliography

[1] P Abbott. Tricks of the trade: Legendre-gauss quadrature. *Mathematica Journal*, 9(4):689–691, 2005.

[2] Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. A reductions approach to fair classification. In *International Conference on Machine Learning*, pages 60–69. PMLR, 2018.

[3] Sina Aghaei, Mohammad Javad Azizi, and Phebe Vayanos. Learning optimal and fair decision trees for non-discriminative decision-making. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1418–1426, 2019.

[4] Wasi Uddin Ahmad, Jianfeng Chi, Yuan Tian, and Kai-Wei Chang. Policyqa: A reading comprehension dataset for privacy policies. *arXiv preprint arXiv:2010.02557*, 2020.

[5] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias. *ProPublica, May*, 23(2016):139–159, 2016.

[6] Michael Aschbacher. On collineation groups of symmetric block designs. *Journal of Combinatorial Theory, Series A*, 1971.

[7] Hiteshwar Kumar Azad and Akshay Deepak. Query expansion techniques for information retrieval: a survey. *Information Processing & Management*, 56(5):1698–1735, 2019.

[8] Joris Baan, Maartje ter Hoeve, Marlies van der Wees, Anne Schuth, and Maarten de Rijke. Do transformer attention heads provide transparency in abstractive summarization? *arXiv preprint arXiv:1907.00570*, 2019.

[9] Ashutosh Baheti, Alan Ritter, Jiwei Li, and Bill Dolan. Generating more interesting responses in neural conversation models with distributional constraints. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3970–3980, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1431. URL https://aclanthology.org/D18-1431.

[10] Ahsaas Bajaj, Pavitra Dangati, Kalpesh Krishna, Pradhiksha Ashok Kumar, Rheeya Uppaal, Bradford Windsor, Eliot Brenner, Dominic Dotterrer, Rajarshi Das, and Andrew McCallum. Long document summarization in a low resource setting using pretrained language models. *arXiv preprint arXiv:2103.00751*, 2021.

[11] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, 2005.

[12] Colin Bannard and Chris Callison-Burch. Paraphrasing with bilingual parallel corpora. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 597–604, 2005.

[13] Solon Barocas and Andrew D Selbst. Big data's disparate impact. *Calif. L. Rev.*, 104:671, 2016.

[14] Anna Maria Barry-jester, Ben Casselman, and Dana Goldstein. The new science of sentencing, Aug 2015. URL https://www.themarshallproject.org/2015/08/04/

the-new-science-of-sentencing.

[15] Mohaddeseh Bastan and Shahram Khadivi. A preordered rnn layer boosts neural machine translation in low resource settings. *arXiv preprint arXiv:2112.13960*, 2021.

[16] Mohaddeseh Bastan, Shahram Khadivi, and Mohammad Mehdi Homayounpour. Neural machine translation on scarce-resource condition: a case-study on persian-english. In *2017 Iranian Conference on Electrical Engineering (ICEE)*, pages 1485–1490. IEEE, 2017.

[17] Yahav Bechavod and Katrina Ligett. Penalizing unfairness in binary classification. *arXiv preprint arXiv:1707.00044*, 2017.

[18] Richard Berk, Hoda Heidari, Shahin Jabbari, Matthew Joseph, Michael Kearns, Jamie Morgenstern, Seth Neel, and Aaron Roth. A convex framework for fair regression. *arXiv preprint arXiv:1706.02409*, 2017.

[19] Richard Berk, Hoda Heidari, Shahin Jabbari, Michael Kearns, and Aaron Roth. Fairness in criminal justice risk assessments: The state of the art. *Sociological Methods & Research*, page 0049124118782533, 2018.

[20] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009.

[21] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *JMLR*, 2003.

[22] April Bleske-Rechek and Kelsey Michels. Ratemyprofessors com: Testing assumptions about student use and misuse. *Practical Assessment, Research, and Evaluation*, 15(1):5, 2010.

[23] Tolga Bolukbasi, Kai-Wei Chang, James Zou, Venkatesh Saligrama, and Adam Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. *arXiv preprint arXiv:1607.06520*, 2016.

[24] John Brockman. *What to think about machines that think: todays leading thinkers on the age of machine intelligence*. HarperCollins, 2015.

[25] Orkut Buyukkokten, Hector Garcia-Molina, and Andreas Paepcke. Seeing the whole in parts: text summarization for web browsing on handheld devices. In *The Web Conference*, 2001.

[26] Toon Calders and Sicco Verwer. Three naive bayes approaches for discrimination-free classification. *Data Mining and Knowledge Discovery*, 21(2):277–292, 2010.

[27] Toon Calders, Faisal Kamiran, and Mykola Pechenizkiy. Building classifiers with independency constraints. In *2009 IEEE International Conference on Data Mining Workshops*, pages 13–18. IEEE, 2009.

[28] Flavio P Calmon, Dennis Wei, Bhanukiran Vinzamuri, Karthikeyan Natesan Ramamurthy, and Kush R Varshney. Optimized pre-processing for discrimination prevention. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 3995–4004, 2017.

[29] Marco Campana and Anastasios Tombros. Incremental personalised summarisation with novelty detection. In *FQAS*, 2009.

[30] Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. Faithful to the original: Fact aware neural abstractive summarization. In *AAAI*, 2018.

[31] Diogo V Carvalho, Eduardo M Pereira, and Jaime S Cardoso. Machine learning interpretability: A survey on methods and metrics. *Electronics*, 8(8):832, 2019.

[32] Asli Celikyilmaz, Antoine Bosselut, Xiaodong He, and Yejin Choi. Deep communicating agents for abstractive summarization. *arXiv preprint arXiv:1803.10357*, 2018.

[33] L Elisa Celis and Vijay Keswani. Improved adversarial learning for fair classification. *arXiv preprint arXiv:1901.10443*, 2019.

[34] L Elisa Celis, Lingxiao Huang, Vijay Keswani, and Nisheeth K Vishnoi. Classification with fairness constraints: A meta-algorithm with provable guarantees. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 319–328, 2019.

[35] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Céspedes, Steve Yuan, Chris Tar, et al. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*, 2018.

[36] Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. Legal-bert: The muppets straight out of law school. *arXiv preprint arXiv:2010.02559*, 2020.

[37] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *JAIR*, 2002.

[38] Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. One billion word benchmark for measuring progress in statistical language modeling. *arXiv preprint arXiv:1312.3005*, 2013.

[39] David L Chen and Raymond J Mooney. Learning to sportscast: a test of grounded language acquisition. In *Proceedings of the 25th international conference on Machine learning*, pages 128–135, 2008.

[40] Qian Chen, Xiao-Dan Zhu, Zhen-Hua Ling, Si Wei, and Hui Jiang. Distraction-based neural networks for modeling document. In *IJCAI*, 2016.

[41] Yen-Chun Chen and Mohit Bansal. Fast abstractive summarization with reinforce-selected sentence rewriting. *arXiv preprint:1805.11080*, 2018.

[42] Zhiyu Chen, Harini Eavani, Wenhu Chen, Yinyin Liu, and William Yang Wang. Few-shot NLG with pre-trained language model. *arXiv preprint arXiv:1904.09521*, 2019.

[43] Jianpeng Cheng and Mirella Lapata. Neural summarization by extracting sentences and words. *arXiv preprint:1603.07252*, 2016.

[44] Sumit Chopra, Michael Auli, and Alexander M. Rush. Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 93–98, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1012. URL https://www.aclweb.org/anthology/N16-1012.

[45] Alexandra Chouldechova and Aaron Roth. The frontiers of fairness in machine learning. *arXiv preprint arXiv:1810.08810*, 2018.

[46] Jishnu Ray Chowdhury, Yong Zhuang, and Shuyi Wang. Novelty controlled paraphrase generation with retrieval augmented conditional prompt tuning. In *Association for the Advancement of Artificial Intelligence (AAAI)*. AAAI, 2022.

[47] Dennis E Clayson. What does ratemyprofessors. com actually rate? *Assessment & Evaluation in Higher Education*, 39(6):678–698, 2014.

[48] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *JMLR*, 12(Aug):2493–2537, 2011.

[49] Evan Comen. Detroit, chicago, memphis the 25 most segregated cities in america, Jul 2019. URL https://www.usatoday.com/story/money/2019/07/20/detroit-chicago-memphis-most-segregated-cities-america-housing-policy/39703787/.

[50] Andrew Cotter, Maya Gupta, Heinrich Jiang, Nathan Srebro, Karthik Sridharan, Serena Wang, Blake Woodworth, and Seungil You. Training well-generalizing classifiers for fairness metrics and other data-dependent constraints. In *International Conference on Machine Learning*, pages 1397–1405. PMLR, 2019.

[51] Lorrie Faith Cranor, Praveen Guduru, and Manjula Arjula. User interfaces for privacy agents. *TOCHI*, 2006.

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

[53] Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. Language modeling with gated convolutional networks. In *ICML*, 2017.

[54] Eustasio Del Barrio, Fabrice Gamboa, Paula Gordaliza, and Jean-Michel Loubes. Obtaining fairness using optimal transport theory. *arXiv preprint arXiv:1806.03195*, 2018.

[55] Michael Denkowski and Alon Lavie. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the ninth workshop on statistical machine translation*, pages 376–380, 2014.

[56] Jacob Devlin, Ming-Wei Chang, et al. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv:1810.04805*, 2018.

[57] AnHai Doan, Pradap Konda, Adel Ardalan, Jeffrey R Ballard, Sanjib Das, Yash Govind, Han Li, Philip Martinkus, Sidharth Mudgal, Erik Paulson, et al. Toward a system building agenda for data integration (and data science). *IEEE Data Eng. Bull.*, 41(2), 2018.

[58] Li Dong, Jonathan Mallinson, Siva Reddy, and Mirella Lapata. Learning to paraphrase for question answering. *arXiv preprint arXiv:1708.06022*, 2017.

[59] Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unified language model pre-training for natural language understanding and generation. *arXiv preprint arXiv:1905.03197*, 2019.

[60] Michele Donini, Luca Oneto, Shai Ben-David, John Shawe-Taylor, and Massimiliano Pontil. Empirical risk minimization under fairness constraints. *arXiv preprint arXiv:1802.08626*, 2018.

[61] Bonnie Dorr, David Zajic, and Richard Schwartz. Hedge trimmer: A parse-and-trim approach to headline generation. Technical report, MARYLAND UNIV COLLEGE PARK INST FOR ADVANCED COMPUTER STUDIES, 2003.

[62] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *JMLR*, 2011.

[63] Charles W Dunnett. New tables for multiple comparisons with a control. *Biometrics*, 1964.

[64] Ondřej Dušek, David M Howcroft, and Verena Rieser. Semantic noise matters for neural natural language generation. *arXiv preprint arXiv:1911.03905*, 2019.

[65] Cynthia Dwork, Nicole Immorlica, Adam Tauman Kalai, and Max Leiserson. Decoupled classifiers for group-fair and efficient machine learning. In *Conference on Fairness, Accountability and Transparency*, pages 119–133. PMLR, 2018.

[66] Nouha Dziri, Ehsan Kamalloo, Sivan Milton, Osmar Zaiane, Mo Yu, Edoardo M Ponti, and Siva Reddy. FaithDial: A faithful benchmark for information-seeking dialogue. *arXiv preprint arXiv:2204.10757*, 2022.

[67] Nouha Dziri, Sivan Milton, Mo Yu, Osmar Zaiane, and Siva Reddy. On the origin of hallucinations in conversational models: Is it the datasets or the models? *arXiv preprint arXiv:2204.07931*, 2022.

[68] Fahimeh Ebrahimi, Miroslav Tushev, and Anas Mahmoud. Mobile app privacy in software engineering research: A systematic mapping study. *Information and Software Technology*, 2020.

[69] Harrison Edwards and Amos Storkey. Censoring representations with an adversary. *arXiv preprint arXiv:1511.05897*, 2015.

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

[71] Angela Fan, Claire Gardent, Chloé Braud, and Antoine Bordes. Using local knowledge graph construction to scale seq2seq models to multi-document inputs. *arXiv preprint arXiv:1910.08435*, 2019.

[72] Atefeh Farzindar and Guy Lapalme. Legal text summarization by exploration of the thematic structure and argumentative roles. In *Text Summarization Branches Out*, pages 27–34, 2004.

[73] Michael Feldman, Sorelle A Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. In *proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 259–268, 2015.

[74] Katja Filippova and Yasemin Altun. Overcoming the lack of parallel data in sentence compression. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1481–1491, 2013.

[75] Benjamin Fish, Jeremy Kun, and Ádám D Lelkes. A confidence-based approach for balancing fairness and accuracy. In *Proceedings of the 2016 SIAM International Conference on Data Mining*, pages 144–152. SIAM, 2016.

[76] Joseph L Fleiss. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378, 1971.

[77] Sorelle A Friedler, Carlos Scheidegger, Suresh Venkatasubramanian, Sonam Choudhary, Evan P Hamilton, and Derek Roth. A comparative study of fairness-enhancing interventions in machine learning. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 329–338, 2019.

[78] Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. The WebNLG challenge: Generating text from RDF data. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 124–133, 2017.

[79] Albert Gatt and Emiel Krahmer. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *JAIR*, 2018.

[80] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. In *ICML*. JMLR. org, 2017.

[81] Sebastian Gehrmann, Yuntian Deng, and Alexander M Rush. Bottom-up abstractive summarization. *arXiv preprint arXiv:1808.10792*, 2018.

[82] AmirEmad Ghassami, Sajad Khodadadian, and Negar Kiyavash. Fairness in supervised learning: An information theoretic approach. In *2018 IEEE International Symposium on Information Theory (ISIT)*, pages 176–180. IEEE, 2018.

[83] Naman Goel, Mohammad Yaghini, and Boi Faltings. Non-discriminatory machine learning through convex fairness criteria. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[84] Heng Gong, Yawei Sun, Xiaocheng Feng, Bing Qin, Wei Bi, Xiaojiang Liu, and Ting Liu. Tablegpt: Few-shot table-to-text generation with table structure reconstruction and content matching. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1978–1988, 2020.

[85] Aria Haghighi and Lucy Vanderwende. Exploring content models for multi-document summarization. In *NAACL*, 2009.

[86] Moritz Hardt, Eric Price, and Nathan Srebro. Equality of opportunity in supervised learning. *arXiv preprint arXiv:1610.02413*, 2016.

[87] Hamza Harkous, Kassem Fawaz, Rémi Lebret, et al. Polisis: Automated analysis and presentation of privacy policies using deep learning. In *27th USENIX Security Symposium (USENIX Security 18)*, 2018.

[88] Hamza Harkous, Isabel Groves, and Amir Saffari. Have your text and use it too! End-to-end neural data-to-text generation with semantic fidelity. *arXiv preprint arXiv:2004.06577*, 2020.

[89] Jibo He. Big data set from ratemyprofessor.com for professors' teaching evaluation, 2020. URL https://data.mendeley.com/datasets/fvtfjyvw7d/2.

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

[91] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[92] Maryam Heidari and Setareh Rafatirad. Semantic convolutional neural network model for safe business investment by using bert. In *International Conference on Social Networks Analysis, Management and Security (SNAMS)*. IEEE, 2020.

[93] Maryam Heidari and Setareh Rafatirad. Using transfer learning approach to implement convolutional neural network model to recommend airline tickets by using online reviews. In *International Workshop on Semantic and Social Media Adaptation and Personalization (SMA*. IEEE, 2020.

[94] 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.

[95] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

[96] Kenneth Holstein, Jennifer Wortman Vaughan, Hal Daumé III, Miro Dudik, and Hanna Wallach. Improving fairness in machine learning systems: What do industry practitioners need? In *Proceedings of the 2019 CHI conference on human factors in computing systems*, pages 1–16, 2019.

[97] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*, 2018.

[98] Clayton J Hutto and Eric Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *ICWSM*, 2014.

[99] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[100] Fereshteh Jafariakinabad and Kien A Hua. Style-aware neural model with application in authorship attribution. In *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*. IEEE, 2019.

[101] Mimansa Jaiswal and Emily Mower Provost. Privacy enhanced multimodal neural representations for emotion recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7985–7993, 2020.

[102] Xin Jia, Wenjie Zhou, SUN Xu, and Yunfang Wu. How to ask good questions? try to leverage paraphrases. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6130–6140, 2020.

[103] Lifeng Jin, David King, Amad Hussein, Michael White, and Douglas Danforth. Using paraphrasing and memory-augmented models to combat data sparsity in question interpretation with a virtual patient dialogue system. In *Proceedings of the thirteenth workshop on innovative use of NLP for building educational applications*, pages 13–23, 2018.

[104] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*, 2014.

[105] Mihir Kale and Abhinav Rastogi. Text-to-text pre-training for data-to-text tasks. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 97–102, Dublin, Ireland, December 2020. Association for Computational Linguistics.

[106] Faisal Kamiran and Indrė Žliobaitė. Explainable and non-explainable discrimination in classification. In *Discrimination and Privacy in the Information Society*, pages 155–170. Springer, 2013.

[107] Faisal Kamiran, Toon Calders, and Mykola Pechenizkiy. Discrimination aware decision tree learning. In *2010 IEEE International Conference on Data Mining*, pages 869–874. IEEE, 2010.

[108] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. Fairness-aware classifier with prejudice remover regularizer. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 35–50. Springer, 2012.

[109] Ambedkar Kanapala, Sukomal Pal, and Rajendra Pamula. Text summarization from legal documents: a survey. *Artificial Intelligence Review*, 51(3):371–402, 2019.

[110] Tapas Kanungo, David M Mount, Nathan S Netanyahu, Christine D Piatko, Ruth Silverman, and Angela Y Wu. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE TPAMI*, 2002.

[111] Ehsan Kazemi, Morteza Zadimoghaddam, and Amin Karbasi. Scalable deletion-robust submodular maximization: Data summarization with privacy and fairness constraints. In *International conference on machine learning*, pages 2544–2553. PMLR, 2018.

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

[113] Moniba Keymanesh, Micha Elsner, and Srinivasan Parthasarathy. Toward domain-guided controllable summarization of privacy policies. *Natural Legal Language Processing Workshop at KDD*, 2020.

[114] Moniba Keymanesh, Saket Gurukar, Bethany Boettner, Christopher Browning, Catherine Calder, and Srinivasan Parthasarathy. Twitter watch: Leveraging social media to monitor and predict collective-efficacy of neighborhoods. In *Complex Networks XI*, pages 197–211. Springer, 2020.

[115] Moniba Keymanesh, Tanya Berger-Wolf, Micha Elsner, and Srinivasan Parthasarathy. Fairness-aware summarization for justified decision-making. *arXiv preprint arXiv:2107.06243*, 2021.

[116] Moniba Keymanesh, Micha Elsner, and Srinivasan Parthasarathy. Privacy policy question answering assistant: A query-guided extractive summarization approach. *arXiv preprint arXiv:2109.14638*, 2021.

[117] Moniba Keymanesh, Adrian Benton, and Mark Dredze. What makes data-to-text generation hard for pretrained language models? *arXiv preprint arXiv:2205.11505*, 2022.

[118] Yuta Kikuchi, Graham Neubig, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. Controlling output length in neural encoder-decoders. *arXiv preprint:1609.09552*, 2016.

[119] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv:1408.5882*, 2014.

[120] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[121] Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. Inherent trade-offs in the fair determination of risk scores. *arXiv preprint arXiv:1609.05807*, 2016.

[122] Philipp Koehn, Franz J Och, and Daniel Marcu. Statistical phrase-based translation. Technical report, UNIVERSITY OF SOUTHERN CALIFORNIA MARINA DEL REY INFORMATION SCIENCES INST, 2003.

[123] James Koren. What does that web search say about your credit?, Sep 2016. URL https://www.latimes.com/business/la-fi-zestfinance-baidu-20160715-snap-story.html.

[124] Uday Krishna. Nltk module for computing the meteor score. https://www.nltk.org/_modules/nltk/translate/meteor_score.html, 2020.

[125] Wojciech Kryściński, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. Neural text summarization: A critical evaluation. *arXiv preprint arXiv:1908.08960*, 2019.

[126] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196. PMLR, 2014.

[127] Angela M Legg and Janie H Wilson. Ratemyprofessors. com offers biased evaluations. *Assessment & Evaluation in Higher Education*, 37(1):89–97, 2012.

[128] Leo Leppänen, Myriam Munezero, Mark Granroth-Wilding, and Hannu Toivonen. Data-driven news generation for automated journalism. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 188–197, 2017.

[129] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021.

[130] 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.

[131] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*, 2015.

[132] Jiwei Li, Will Monroe, and Dan Jurafsky. A simple, fast diverse decoding algorithm for neural generation. *arXiv preprint arXiv:1611.08562*, 2016.

[133] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*, 2004.

[134] Chin-Yew Lin and Eduard Hovy. Manual and automatic evaluation of summaries. In *ACL*, 2002.

[135] Hui Lin and Vincent Ng. Abstractive summarization: A survey of the state of the art. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9815–9822, 2019.

[136] TurkPrime Litman. com: A versatile crowdsourcing data acquisition platform for the behavioral sciences. *Behavior Research Methods*, 2017.

[137] Chunyi Liu, Peng Wang, Jiang Xu, Zang Li, and Jieping Ye. Automatic dialogue summary generation for customer service. In *SIGKDD*, 2019.

[138] Fei Liu, Jeffrey Flanigan, Sam Thomson, Norman Sadeh, and Noah A. Smith. Toward abstractive summarization using semantic representations. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1077–1086, Denver, Colorado, May–June 2015. Association for Computational Linguistics. doi: 10.3115/v1/N15-1114. URL https://www.aclweb.org/anthology/N15-1114.

[139] Frederick Liu, Shomir Wilson, Peter Story, Sebastian Zimmeck, and Norman Sadeh. Towards automatic classification of privacy policy text. *School of Computer Science Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-ISR-17-118R and CMULTI-17-010*, 2018.

[140] Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. What makes good in-context examples for GPT-3? *arXiv preprint arXiv:2101.06804*, 2021.

[141] Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. GPT understands, too. *arXiv preprint arXiv:2103.10385*, 2021.

[142] Yang Liu. Fine-tune bert for extractive summarization. *arXiv preprint arXiv:1903.10318*, 2019.

[143] Yang Liu and Mirella Lapata. Text summarization with pretrained encoders. *arXiv preprint arXiv:1908.08345*, 2019.

[144] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[145] Yizhu Liu, Zhiyi Luo, and Kenny Zhu. Controlling length in abstractive summarization using a convolutional neural network. In *EMNLP*, 2018.

[146] Kaixin Ma, Hao Cheng, Xiaodong Liu, Eric Nyberg, and Jianfeng Gao. Open domain question answering over virtual documents: A unified approach for data and text. *arXiv preprint arXiv:2110.08417*, 2021.

[147] David Madras, Elliot Creager, Toniann Pitassi, and Richard Zemel. Learning adversarially fair and transferable representations. In *International Conference on Machine Learning*, pages 3384–3393. PMLR, 2018.

[148] Manuel Mager, Ramón Fernandez Astudillo, Tahira Naseem, Md Arafat Sultan, Young-Suk Lee, Radu Florian, and Salim Roukos. GPT-too: A language-model-first approach for amr-to-text generation. *arXiv preprint arXiv:2005.09123*, 2020.

[149] Ahmed Magooda and Diane Litman. Abstractive summarization for low resource data using domain transfer and data synthesis. *arXiv preprint arXiv:2002.03407*, 2020.

[150] Jonathan Mallinson, Rico Sennrich, and Mirella Lapata. Paraphrasing revisited with neural machine translation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 881–893, 2017.

[151] Christopher Manning, Prabhakar Raghavan, and Hinrich Schütze. Introduction to information retrieval. *Natural Language Engineering*, 16(1):100–103, 2010.

[152] Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *ACL*, 2014.

[153] Laura Manor and Junyi Jessy Li. Plain English summarization of contracts. In *Proceedings of the Natural Legal Language Processing Workshop 2019*, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[154] Laura Manor and Junyi Jessy Li. Plain english summarization of contracts. *arXiv:1906.00424*, 2019.

[155] Diana McCarthy and Roberto Navigli. The english lexical substitution task. *Language resources and evaluation*, 43(2):139–159, 2009.

[156] Aleecia M McDonald and Lorrie Faith Cranor. The cost of reading privacy policies. *Isjlp*, 2008.

[157] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *arXiv preprint arXiv:1908.09635*, 2019.

[158] Hongyuan Mei, Mohit Bansal, and Matthew R Walter. What to talk about and how? selective generation using lstms with coarse-to-fine alignment. *arXiv preprint arXiv:1509.00838*, 2015.

[159] Aditya Krishna Menon and Robert C Williamson. The cost of fairness in binary classification. In *Conference on Fairness, Accountability and Transparency*, pages 107–118. PMLR, 2018.

[160] Rada Mihalcea and Paul Tarau. Textrank: Bringing order into text. In *EMNLP*, 2004.

[161] George A Miller. Wordnet: a lexical database for English. *Communications of the ACM*, 38 (11):39–41, 1995.

[162] George A Miller. *WordNet: An electronic lexical database*. MIT press, 1998.

[163] Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011, 2009.

[164] Seungwhan Moon, Pararth Shah, Anuj Kumar, and Rajen Subba. Opendialkg: Explainable conversational reasoning with attention-based walks over knowledge graphs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 845–854, 2019.

[165] Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*, 2016.

[166] Ramesh Nallapati, Bowen Zhou, and Mingbo Ma. Classify or select: Neural architectures for extractive document summarization. *arXiv preprint arXiv:1611.04244*, 2016.

[167] Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[168] Linyong Nan, Dragomir Radev, Rui Zhang, Amrit Rau, Abhinand Sivaprasad, Chiachun Hsieh, Xiangru Tang, Aadit Vyas, Neha Verma, Pranav Krishna, et al. Dart: Open-domain structured data record to text generation. *arXiv preprint arXiv:2007.02871*, 2020.

[169] Shashi Narayan, Shay B Cohen, and Mirella Lapata. Ranking sentences for extractive summarization with reinforcement learning. *arXiv preprint:1802.08636*, 2018.

[170] Najmeh Mousavi Nejad, Damien Graux, and Diego Collarana. Towards measuring risk factors in privacy policies. In *ICAIL*, 2019.

[171] Ani Nenkova and Kathleen McKeown. A survey of text summarization techniques. In *Mining text data*. Springer, 2012.

[172] Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. The E2E dataset: New challenges for end-to-end generation. *arXiv preprint arXiv:1706.09254*, 2017.

[173] Jonathan A Obar and Anne Oeldorf-Hirsch. The biggest lie on the internet: Ignoring the privacy policies and terms of service policies of social networking services. *ICS*, 2020.

[174] Osonde A Osoba and William Welser IV. *An intelligence in our image: The risks of bias and errors in artificial intelligence*. Rand Corporation, 2017.

[175] Jahna Otterbacher, Dragomir Radev, and Omer Kareem. News to go: hierarchical text summarization for mobile devices. In *SIGIR*, 2006.

[176] Jessica Ouyang, Serina Chang, and Kathy McKeown. Crowd-sourced iterative annotation for narrative summarization corpora. In *EACL*, 2017.

[177] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *TKDE*, 2009.

[178] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[179] Shantipriya Parida and Petr Motlicek. Abstract text summarization: A low resource challenge. In *EMNLP*. Association for Computational Linguistics, November 2019.

[180] Romain Paulus, Caiming Xiong, and Richard Socher. A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304*, 2017.

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

[182] Mark J. Perry. Bachelors degrees by field and gender for the class of 2015, Aug 2017. URL https://www.aei.org/carpe-diem/table-of-the-day-bachelors-degrees-by-field-and-gender/.

[183] Dana Pessach and Erez Shmueli. Algorithmic fairness. *arXiv preprint arXiv:2001.09784*, 2020.

[184] Matthew E Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. Semi-supervised sequence tagging with bidirectional language models. *arXiv preprint arXiv:1705.00108*, 2017.

[185] Matthew E Peters, Mark Neumann, Mohit Iyyer, et al. Deep contextualized word representations. *arXiv:1802.05365*, 2018.

[186] Vassilis Plachouras, Charese Smiley, Hiroko Bretz, Ola Taylor, Jochen L Leidner, Dezhao Song, and Frank Schilder. Interacting with financial data using natural language. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 1121–1124, 2016.

[187] Maja Popović. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, 2015.

[188] François Portet, Ehud Reiter, Albert Gatt, Jim Hunter, Somayajulu Sripada, Yvonne Freer, and Cindy Sykes. Automatic generation of textual summaries from neonatal intensive care data. *Artificial Intelligence*, 173(7-8):789–816, 2009.

[189] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[190] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019.

[191] Aida Rahmattalabi, Shahin Jabbari, Himabindu Lakkaraju, Phebe Vayanos, Eric Rice, and Milind Tambe. Fair influence maximization: A welfare optimization approach. *arXiv preprint arXiv:2006.07906*, 2020.

[192] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

[193] Alejandro Ramos-Soto, Alberto Jose Bugarin, Senén Barro, and Juan Taboada. Linguistic descriptions for automatic generation of textual short-term weather forecasts on real prediction data. *IEEE Transactions on Fuzzy Systems*, 23(1):44–57, 2014.

[194] Abhilasha Ravichander, Alan Black, Eduard Hovy, Joel Reidenberg, N Cameron Russell, and Norman Sadeh. Challenges in automated question answering for privacy policies. *Proceedings of the PAL: Privacy-Enhancing Artificial Intelligence and Language Technologies*, 2019.

[195] Abhilasha Ravichander, Alan W Black, Shomir Wilson, Thomas Norton, and Norman Sadeh. Question answering for privacy policies: Combining computational and legal perspectives. *arXiv preprint arXiv:1911.00841*, 2019.

[196] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA. http://is.muni.cz/publication/884893/en.

[197] Landon D Reid. The role of perceived race and gender in the evaluation of college teaching on ratemyprofessors. com. *Journal of Diversity in higher Education*, 3(3):137, 2010.

[198] Leonardo FR Ribeiro, Martin Schmitt, Hinrich Schütze, and Iryna Gurevych. Investigating pretrained language models for graph-to-text generation. *arXiv preprint arXiv:2007.08426*, 2020.

[199] Stefan Riezler, Alexander Vasserman, Ioannis Tsochantaridis, Vibhu O Mittal, and Yi Liu. Statistical machine translation for query expansion in answer retrieval. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 464–471, 2007.

[200] Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. Automatic keyword extraction from individual documents. *Text mining: applications and theory*, 2010.

[201] Andrew S Rosen. Correlations, trends and potential biases among publicly accessible web-based student evaluations of teaching: a large-scale study of ratemyprofessors. com data. *Assessment & Evaluation in Higher Education*, 43(1):31–44, 2018.

[202] Cynthia Rudin. Predictive policing using machine learning to detect patterns of crime. *Wired Magazine, August*, 2013.

[203] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 1986.

[204] Alexander M Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. *arXiv preprint:1509.00685*, 2015.

[205] Itsumi Saito, Kyosuke Nishida, Kosuke Nishida, Atsushi Otsuka, Hisako Asano, Junji Tomita, Hiroyuki Shindo, and Yuji Matsumoto. Length-controllable abstractive summarization by guiding with summary prototype. *arXiv preprint:2001.07331*, 2020.

[206] Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*, 2021.

[207] Ritesh Sarkhel, Moniba Keymanesh, Arnab Nandi, and Srinivasan Parthasarathy. Interpretable multi-headed attention for abstractive summarization at controllable lengths. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6871–6882, 2020.

[208] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45, 1997.

[209] Abigail See, Peter J Liu, and Christopher D Manning. Get to the point: Summarization with pointer-generator networks. *arXiv preprint:1704.04368*, 2017.

[210] Abigail See, Aneesh Pappu, Rohun Saxena, Akhila Yerukola, and Christopher D Manning. Do massively pretrained language models make better storytellers? *arXiv preprint arXiv:1909.10705*, 2019.

[211] Thibault Sellam, Dipanjan Das, and Ankur P Parikh. BLEURT: Learning robust metrics for text generation. *arXiv preprint arXiv:2004.04696*, 2020.

[212] Anurag Shandilya, Kripabandhu Ghosh, and Saptarshi Ghosh. Fairness of extractive text summarization. In *Companion Proceedings of the The Web Conference 2018*, pages 97–98, 2018.

[213] Huajie Shao, Jun Wang, Haohong Lin, Xuezhou Zhang, Aston Zhang, Heng Ji, and Tarek Abdelzaher. Controllable and diverse text generation in e-commerce. In *Proceedings of the Web Conference 2021*, pages 2392–2401, 2021.

[214] Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231, 2006.

[215] Balaji Vasan Srinivasan, Pranav Maneriker, Kundan Krishna, and Natwar Modani. Corpus-based content construction. In *COLING*, 2018.

[216] Emily Steel and Julia Angwin. On the web's cutting edge, anonymity in name only, Aug 2010. URL https://www.wsj.com/articles/SB10001424052748703294904575385532109190198.

[217] Daniel Storage, Zachary Horne, Andrei Cimpian, and Sarah-Jane Leslie. The frequency of "brilliant" and "genius" in teaching evaluations predicts the representation of women and african americans across fields. *PloS one*, 11(3):e0150194, 2016.

[218] Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in nlp. *arXiv preprint arXiv:1906.02243*, 2019.

[219] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328. PMLR, 2017.

[220] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[221] Simon Sweeney, Fabio Crestani, and David E Losada. 'show me more': Incremental length summarisation using novelty detection. *Information Processing & Management*, 2008.

[222] Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. Abstractive document summarization with a graph-based attentional neural model. In *ACL*, 2017.

[223] Welderufael B Tesfay, Peter Hofmann, Toru Nakamura, Shinsaku Kiyomoto, and Jetzabel Serna. Privacyguide: Towards an implementation of the eu gdpr on internet privacy policy evaluation. In *IWSPA*, 2018.

[224] Katherine C Theyson. Hot or not: The role of instructor quality and gender on the formation of positive illusions among students using ratemyprofessors. com. *Practical Assessment, Research, and Evaluation*, 20(1):4, 2015.

[225] Chris van der Lee, Emiel Krahmer, and Sander Wubben. PASS: A dutch data-to-text system for soccer, targeted towards specific audiences. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 95–104, 2017.

[226] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017.

[227] Christina Wadsworth, Francesca Vera, and Chris Piech. Achieving fairness through adversarial learning: an application to recidivism prediction. *arXiv preprint arXiv:1807.00199*, 2018.

[228] Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. Universal adversarial triggers for attacking and analyzing nlp. *arXiv preprint arXiv:1908.07125*, 2019.

[229] Eric Wallace, Tony Z Zhao, Shi Feng, and Sameer Singh. Concealed data poisoning attacks on nlp models. *arXiv preprint arXiv:2010.12563*, 2020.

[230] Frank Wilcoxon, SK Katti, and Roberta A Wilcox. Critical values and probability levels for the wilcoxon rank sum test and the wilcoxon signed rank test. *Selected tables in mathematical statistics*, 1970.

[231] Shomir Wilson, Florian Schaub, Aswarth Abhilash Dara, Frederick Liu, et al. The creation and analysis of a website privacy policy corpus. In *ACL*, 2016.

[232] Sam Wiseman, Stuart M Shieber, and Alexander M Rush. Challenges in data-to-document generation. *arXiv preprint arXiv:1707.08052*, 2017.

[233] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

[234] Richard Saul Wurman, Loring Leifer, David Sume, and Karen Whitehouse. *Information anxiety 2*. Que, 2001.

[235] Depeng Xu, Shuhan Yuan, Lu Zhang, and Xintao Wu. Fairgan: Fairness-aware generative adversarial networks. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 570–575. IEEE, 2018.

[236] Christopher C Yang and Fu Lee Wang. Fractal summarization for mobile devices to access large documents on the web. In *The Web Conference*, 2003.

[237] Yi Yang, Wen-tau Yih, and Christopher Meek. Wikiqa: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 2013–2018, 2015.

[238] Jin-ge Yao, Xiaojun Wan, and Jianguo Xiao. Recent advances in document summarization. *KAIS*, 2017.

[239] Michihiro Yasunaga, Rui Zhang, Kshitijh Meelu, Ayush Pareek, Krishnan Srinivasan, and Dragomir Radev. Graph-based neural multi-document summarization. *arXiv preprint arXiv:1706.06681*, 2017.

[240] Razieh Nokhbeh Zaeem, Rachel L German, and K Suzanne Barber. Privacycheck: Automatic summarization of privacy policies using data mining. *TOIT)*, 2018.

[241] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P Gummadi. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In *Proceedings of the 26th international conference on world wide web*, pages 1171–1180, 2017.

[242] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, Krishna P Gummadi, and Adrian Weller. From parity to preference-based notions of fairness in classification. *arXiv preprint arXiv:1707.00010*, 2017.

[243] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rogriguez, and Krishna P Gummadi. Fairness constraints: Mechanisms for fair classification. In *Artificial Intelligence and Statistics*, pages 962–970. PMLR, 2017.

[244] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning fair representations. In *International conference on machine learning*, pages 325–333. PMLR, 2013.

[245] Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 335–340, 2018.

[246] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization, 2019.

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

[248] Xingxing Zhang, Furu Wei, and Ming Zhou. Hibert: Document level pre-training of hierarchical bidirectional transformers for document summarization. *arXiv preprint arXiv:1905.06566*, 2019.

[249] Ye Zhang and Byron Wallace. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv:1510.03820*, 2015.

[250] Linwu Zhong, Ziyi Zhong, Zinian Zhao, Siyuan Wang, Kevin D Ashley, and Matthias Grabmair. Automatic summarization of legal decisions using iterative masking of predictive sentences. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law*, pages 163–172, 2019.

[251] Qingyu Zhou, Nan Yang, Furu Wei, Shaohan Huang, Ming Zhou, and Tiejun Zhao. Neural document summarization by jointly learning to score and select sentences. *arXiv preprint arXiv:1807.02305*, 2018.

[252] Sebastian Zimmeck and Steven M Bellovin. Privee: An architecture for automatically analyzing web privacy policies. In *23rd USENIX Security Symposium (USENIX Security 14)*, 2014.

[253] Ingrid Zukerman and Bhavani Raskutti. Lexical query paraphrasing for document retrieval. In *COLING 2002: The 19th International Conference on Computational Linguistics*, 2002.