





Thesis Title: Towards Controllable Neural Generation of Arguments

Author: Xinyu Hua

PhD Thesis Approval to complete all degree requirements for the PhD in Computer Science.

 _____ Thesis Advisor	<u>05/24/2021</u> _____ Date
 _____ Thesis Reader	<u>05/24/2021</u> _____ Date
 _____ Thesis Reader	<u>05/24/2021</u> _____ Date
 _____ Thesis Reader	<u>05/24/2021</u> _____ Date
_____ Thesis Reader	_____ Date

KHOURY COLLEGE APPROVAL:

 _____ Associate Dean for Graduate Programs	<u>6/1/21</u> _____ Date
--	--------------------------------

COPY RECEIVED BY GRADUATE STUDENT SERVICES:

 _____ Recipient's Signature	<u>5/26/21</u> _____ Date
---	---------------------------------

Distribution: Once completed, this form should be scanned and attached to the front of the electronic dissertation document (page 1). An electronic version of the document can then be uploaded to the Northeastern University-UMI Website.

TOWARDS CONTROLLABLE NEURAL GENERATION OF ARGUMENTS

A Dissertation

Presented to The Khoury College of Computer Sciences
of Northeastern University

in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy

Thesis Committee:

Lu Wang (Chair), Northeastern University

David Smith, Northeastern University

Byron Wallace, Northeastern University

Vincent Ng, University of Texas at Dallas

by

Xinyu Hua

May 2021

© 2021 Xinyu Hua
ALL RIGHTS RESERVED

TOWARDS CONTROLLABLE NEURAL GENERATION OF ARGUMENTS

Xinyu Hua, Ph.D.

Northeastern University 2021

Argumentation is an essential cognitive skill that we utilize to achieve various communicative goals. It has great impact on all aspects of our lives, ranging from policymaking to conflict resolutions. The automatic construction of persuasive arguments thus presents great opportunities, as it can significantly reduce the workload required to search evidence and existing opinions of the issue at hand. Yet the research on argument generation is relatively under-explored. Existing methods either rely on manually crafted rules tailored for specific domains and language styles, or built on pre-indexed argument inventory for retrieval. It remains unclear whether the retrieved arguments can be organically integrated into the underlying text generation stack in a controlled manner. This dissertation presents a line of research aimed at more controllable neural text generation systems for arguments. We focus on two key aspects: (1) the proper inclusion and organization of diverse external knowledge that are pertinent to the given topic and stance, and (2) the generation modeling design that enables effective and interpretable text planning and conditional realization. We experiment with a newly collected Reddit ChangeMyView corpus and highlight the great impact of retrieval results as well as the quality of text planning. We then showcase how the controlled generation framework can be generalized to other domains with distinct language styles. Finally, we describe two novel formulations designed for pre-trained Transformers to achieve improved fluency and controllability for argument generation and beyond.

ACKNOWLEDGEMENTS

The past five years have been the most challenging, yet rewarding journey of my life. Soon after I joined the PhD program, I learned that a large percentage of PhD students end up not finishing their programs, perhaps many of whom are also smarter and more hard-working than I was. Every time when I look back to those days, I remind myself how lucky I was to have those who supported me throughout my way.

First and foremost, I would like to express my deepest gratitude to Professor Lu Wang, without whom none of my work would be possible. Lu is always extremely dedicated to her research, and make sure every student of hers can walk up to her almost anytime for anything. Before I met Lu, I had very little knowledge of neural networks and text generation. She was willing to teach me from the smallest details. I am constantly amazed by her insights on what are the important problems, and what are attainable in reasonable time. She also taught me that, as researchers, the work we do defines who we are, and it deserves our fullest devotion.

I would also like to thank Professor David Smith, Byron Wallace, and Vincent Ng, for serving as my committee members. Vincent is an expert in argument mining. His thorough and insightful comments has improved many aspects of this dissertation. Byron offered numerous suggestions regarding the ethical considerations and model comparisons. I also benefited enormously from the weekly NLP reading group with his group. David's boundless knowledge in language and machine learning has always encouraged me to explore more. He has also been extremely supportive in research, course-work, and administration during my last year at Northeastern.

During my internship at IBM research, I was fortunate to be supervised by

Avi Sil and Radu (Hans) Florian. Avi taught me what types of research problems are important under industrial settings. Hans and other members at the IBM question answering team offered me opportunities to experiment with the newest training techniques for pre-trained Transformer models. I also had the pleasure to work with Lei Li and Lifeng Hua at ByteDance AI lab. Our collaboration on Chinese entity linking gave me perspectives on unique problems that are rarely seen in English data.

My colleagues and friends at Northeastern have made my journey more enjoyable, and my most difficult times more hopeful. Kechen Qin, Ryan Muther, and Ansel MacLaughlin joined the program at the same time as I did. We spent countless hours in classrooms and labs, and memorable trips exploring Boston. Zhe Hu has been a great co-author, and contributed substantially to our work on argument generation. Mitko Nikolov, Nikhil Badugu, Michael Spector, and Jiaxin Pei are key members in our year-long annotation project on peer reviews. Although I have not collaborated with Shuyang Cao, Luyang Huang, and Ken Chan, our fun times at 177 Huntington and beyond have made deadline nights less stressful. I am also thankful to Sarah Gale, Professor Rajmohan Rajaraman, and Professor Frank Tips, who helped me navigate the academic roadmap to make sure I meet the milestones and actually finish my program.

Lastly, I am eternally in debt to my parents, Ziduo and Wenhao, for their continued support and unconditional love throughout my life. They gave me a peaceful childhood and opportunity to attend college. I hope this dissertation will make them proud.

The work in this dissertation is supported in part by the National Science Foundation Grant IIS-1566382, IIS-1813341, and a GPU gift from Nvidia.

TABLE OF CONTENTS

Acknowledgements	4
Table of Contents	6
List of Tables	10
List of Figures	15
1 Introduction	1
1.1 Motivation	1
1.2 Thesis Outline and Contributions	3
2 Background	6
2.1 Argument Mining	6
2.1.1 Argument Structure Extraction	6
2.1.2 Argument Retrieval	7
2.1.3 Argument Generation	8
2.2 Text Generation	9
2.2.1 Text Generation Pipeline	9
2.2.2 Neural Text Generation Models	10
2.2.3 Knowledge-Enhanced Text Generation	11
2.3 Controllable Generation	12
2.3.1 What Aspects Do We Need to Control?	13
2.3.2 Controlled Generation by Conditional Language Model	14
2.3.3 Controlled Generation by Editing and Refinement	14
3 Understanding Argument Types and Support Relations	16
3.1 Introduction	16
3.2 Data and Annotation	19
3.3 A Study on Argument Type Prediction	22
3.4 Supporting Argument Detection	25
3.5 Conclusion	28
4 Argument Mining on Peer Reviews	29
4.1 Introduction	29
4.2 AMPERE Dataset	32
4.3 Experiments with Existing Models	35
4.3.1 Task I: Proposition Segmentation	36
4.3.2 Task II: Proposition Classification	37
4.3.3 Training Details	38
4.4 Proposition Analysis by Venues	39
4.5 Conclusion	43

5	Argument Generation Augmented with Externally Retrieved Evidence	45
5.1	Introduction	45
5.2	Framework	49
5.3	Data Collection and Processing	49
5.4	Model	51
5.4.1	Model Formulation	52
5.4.2	Hybrid Beam Search Decoding	54
5.5	Relevant Evidence Retrieval	55
5.5.1	Retrieval Methodology	55
5.5.2	Gold-Standard Keyphrase Construction	57
5.6	Experimental Setup	58
5.6.1	Final Dataset Statistics	58
5.6.2	Training Setup	58
5.6.3	Baseline and Comparisons	60
5.7	Results	61
5.7.1	Automatic Evaluation	61
5.7.2	Topic-Relevance Evaluation	63
5.7.3	Human Evaluation	65
5.8	Further Discussion	66
5.9	Conclusion	68
6	Argument Generation with Retrieval, Planning, and Realization	69
6.1	Introduction	69
6.2	Overview of CANDELA	72
6.3	Argument Retrieval	73
6.3.1	Information Sources and Indexing	73
6.3.2	Keyphrase Extraction	75
6.3.3	Passage Ranking and Filtering	76
6.4	Argument Generation	77
6.4.1	Task Formulation	77
6.4.2	Text Planning Decoder	78
6.4.3	Content Realization Decoder	80
6.4.4	Training Objective	81
6.5	Experimental Setups	82
6.5.1	Data Collection and Preprocessing	82
6.5.2	System and Oracle Retrieved Passages	83
6.5.3	Comparisons	83
6.5.4	Training Details	84
6.6	Results and Analysis	85
6.6.1	Automatic Evaluation	85
6.6.2	Human Evaluation	88
6.6.3	Sample Arguments and Discussions	90
6.7	Conclusion	92

7	Sentence-Level Content Planning and Style Specification for Neural Text Generation	93
7.1	Introduction	93
7.2	Model	96
7.2.1	Input Encoding	97
7.2.2	Sentence-Level Content Planning and Style Specification	97
7.2.3	Style-Controlled Surface Realization	99
7.2.4	Training Objective	100
7.3	Tasks and Datasets	101
7.3.1	Task I: Argument Generation	101
7.3.2	Task II: Paragraph Generation for Normal and Simple Wikipedia	104
7.3.3	Task III: Paper Abstract Generation	105
7.4	Experiments	106
7.4.1	Implementation Details	106
7.4.2	Baselines and Comparisons	107
7.5	Results and Analysis	108
7.5.1	Automatic Evaluation	108
7.5.2	Human Evaluation	111
7.5.3	Further Analysis and Discussions	113
7.6	Conclusion	114
8	Planning and Iterative Refinement in Pre-trained Transformers for Long Text Generation	117
8.1	Introduction	117
8.2	Content-controlled Text Generation with PAIR	120
8.2.1	Content Planning with BERT	121
8.2.2	Adding Content Plan with a Template Mask-and-Fill Procedure	123
8.2.3	Iterative Refinement	125
8.3	Experiment Setups	126
8.3.1	Tasks and Datasets	126
8.3.2	Implementation Details	129
8.3.3	Baselines and Comparisons	130
8.4	Results	130
8.4.1	Automatic Evaluation	130
8.4.2	Human Evaluation	133
8.5	Further Discussions on Discourse	135
8.6	Conclusion	138
9	DYPLOC: Dynamic Planning of Content Using Mixed Language Models for Text Generation	139
9.1	Introduction	139
9.2	Model	143

9.2.1	Content Item Augmentation	144
9.2.2	Content Realization via Mixed Conditioning	145
9.3	Experiment Setups	147
9.3.1	Tasks and Datasets	147
9.3.2	Content Item Construction	148
9.3.3	Baselines and Comparisons	151
9.3.4	Reproducibility	152
9.4	Results	153
9.4.1	Automatic Evaluation	153
9.4.2	Human Evaluation	155
9.5	Further Discussions	156
9.6	Conclusion	160
10	Bias and Ethical Considerations	162
10.1	Potential Biases in the Dataset	162
10.2	Ethical Concerns	163
10.3	Unfaithful Generation and Misinformation	164
11	Conclusion and Future Directions	166
11.1	Conclusion	166
11.2	Future Directions	168

LIST OF TABLES

3.1	Annotation scheme and examples for our dataset.	20
3.2	The statistics of four types on our annotated dataset.	22
3.3	One-vs-rest classification results.	24
3.4	Supporting argument detection results. “Sentence” stands for features discussed in section 4 for modeling candidate sentences. “Similarity” refers to similarity features. Comp(type, Sen) stands for composite features of argument type and sentence features, similarly for Comp(type,Sim). Comp(type,Claim) represents composite features of type and claim features. Results that are statistically significantly better than all three baselines are marked with * (paired t -test, $p < 0.05$).	27
3.5	Comparison of feature significance under composition with different types. The number of * stands for the p -value based on t -test between supporting argument sentences and the others after Bonferroni correction. From one * to four, the p -value scales as: 0.05, 1e-3, 1e-5, and 1e-10. When mean value of supporting argument sentences is larger, \uparrow is used; otherwise, \downarrow is displayed. Number of arrows represents the ratio of the larger value over smaller one. “-” indicates no significant difference.	28
4.1	Statistics for review data sets used in our study.	32
4.2	Review length distribution of the full ICLR 2018 dataset and AMPERE, which consists of 400 sampled reviews.	34
4.3	Review rating distribution of AMPERE and the full ICLR 2018 dataset.	34
4.4	Inter-annotator agreement for all categories.	34
4.5	Statistics for AMPERE and some other argument mining corpora, including # of annotated propositions.	35
4.6	Number of propositions per type in AMPERE.	35
4.7	Proposition segmentation results. Result that is significantly better than all comparisons is marked with * ($p < 10^{-6}$, McNemar test).	36
4.8	Proposition classification F1 scores. Results that are significant better than other methods are marked with * ($p < 10^{-6}$, McNemar test).	38
4.9	Salient words ($\alpha = 0.001$, χ^2 test) per proposition type. Top 5 frequent words that are unique for each venue are shown. “<EQN>”, “<URL>”, and “<VAR>” are equations, URL links, and variables.	42
5.1	Lists of politics and non-politics keywords used to obtain Wikipedia abstract labels.	51

5.2	Statistics for evidence sentence retrieval from Wikipedia. Considering query construction from either OP or target user arguments, we show the average numbers of topic signatures collected, queries constructed, and retrieved articles and sentences.	56
5.3	Truncation size (i.e., number of tokens including delimiters) for different stages during training. Note that in the first stage we do not include evidence and keyphrases.	57
5.4	Results on argument generation by BLEU and METEOR, with system retrieved evidence and oracle retrieval. The best performing model is highlighted in bold per metric. Our separate decoder models, with and without keyphrase attention, statistically significantly outperform all seq2seq-based models based on approximation randomization testing (Noreen, 1989), $p < 0.0001$	62
5.5	Evaluation on topic relevance—models that generate arguments highly related with OP should be ranked high by a separately trained relevance estimation model, i.e., higher Mean Reciprocal Rank (MRR) and Precision at 1 (P@1) scores. All models trained with evidence significantly outperform seq2seq trained without evidence (approximation randomization testing, $p < 0.0001$). . .	65
5.6	Human evaluation results on grammaticality, informativeness, and relevance of arguments. Our model with separate decoder and attention over keyphrases receives significantly better ratings in informativeness and relevance than seq2seq (one-way ANOVA, $p < 0.005$).	66
6.1	Statistics on information sources for argument retrieval. News media are sorted by ideological leanings from left to right, according to https://www.adfontesmedia.com/	74
6.2	Statistics on the datasets for experiments.	83
6.3	Main results on argument generation using system retrieval. We report BLEU-2, BLEU-4, ROUGE-2 recall, METEOR, and average number of words per argument and per sentence. Best scores are in bold. *: statistically significantly better than all comparisons (randomization approximation test (Noreen, 1989), $p < 0.0005$). .	86
6.4	Results using oracle retrieved passages. *: statistically significantly better than all comparisons (randomization approximation test (Noreen, 1989), $p < 0.0005$). Input is the same for SEQ2SEQ for both system and oracle setups.	87

6.5	Human evaluation on grammaticality (Gram), appropriateness (Appr), and content richness (Cont.), on a scale of 1 to 5 (best). The best result among automatic systems is highlighted in bold, with statistical significance marked with * (approximation randomization test, $p < 0.0005$). The highest standard deviation among all is 1.0. Top-1/2: % of evaluations a system being ranked in top 1 or 2 for overall quality.	89
7.1	Statistics of the three datasets. Average numbers are reported. For argument dataset, number of unique threads is also shown. On AGENDA, entities are extracted from abstract as keyphrases, hence all candidates are “selected”.	102
7.2	Sentence style distribution for argument and Wikipedia datasets.	103
7.3	Patterns for sentence style label construction on CLAIM and PREMISE for argument generation.	104
7.4	Results on argument generation with BLEU (up to bigrams), ROUGE-L, and METEOR (MTR). Best systems without oracle planning are in bold per metric. Our models that are significantly better than all comparisons are marked with * ($p < 0.001$, approximate randomization test (Noreen, 1989)).	109
7.5	Results on Wikipedia generation. Best results without oracle planning are in bold . *: Our models that are significantly better than all comparisons ($p < 0.001$, approximate randomization test).	110
7.6	Results on paper abstract generation. Notice that GRAPHWRITER models rich information about relations and relation types among entities, which is not utilized by our model.	111
7.7	Human evaluation on argument generation (Upper) and Wikipedia generation (Bottom). Grammaticality (Gram), correctness (Corr), and content richness (Cont) are rated on Likert scale (1 – 5). We mark our model with * to indicate statistically significantly better ratings over the variant without style specification ($p < 0.001$, approximate randomization test).	111
7.8	Top frequent patterns captured in style CLAIM, PREMISE, and FUNCTIONAL from arguments by human and our model.	113
7.9	Evaluation guidelines on argument data and representative examples on rating scales.	115
7.10	Evaluation guidelines on Wikipedia data and representative examples on rating scales.	116

8.1	Statistics on generated templates by our content planner. Tokens are measured in units of WordPiece (Sennrich et al., 2016). KP distance denotes the average number of tokens between two keyphrases that are in the same sentence. Both system output (<i>sys</i>) and human reference (<i>ref</i>) are reported.	124
8.2	Statistics of the three datasets. We report average lengths of the prompt and the target generation, number of unique keyphrases (# KP) used in the input, and the percentage of content words in target covered by the keyphrases (KP Cov).	127
8.3	Key results on argument generation, opinion article writing, and news report generation. BLEU-4 (B-4), ROUGE-L (R-L), METEOR (MTR), and average output lengths are reported (for references, the lengths are 100, 166, and 250, respectively). PAIR _{light} , using keyphrase assignments only, consistently outperforms baselines; adding keyphrase positions, PAIR _{full} further boosts scores. Improvements by our models over baselines are all significant ($p < 0.0001$, approximate randomization test). Iterative refinement helps on both setups.	131
8.4	Human evaluation for argument generation on fluency, coherence, and relevance, with 5 as the best. The Krippendorff’s α are 0.28, 0.30, and 0.37, respectively. Our model outputs are significantly more coherent and relevant than KPSEQ2SEQ (*: $p < 0.0001$), with comparable fluency.	133
8.5	Sample outputs in the news and opinion domain. Keyphrases assigned to different sentences are in boldface and color-coded.	134
8.6	Percentages of samples preferred by human judges before and after refinement [Left]; with and without enforcing keyphrases to appear at the predicted positions [Right]. Ties are omitted.	135
9.1	Statistics of the two datasets. We report average numbers of concepts and entities per content item, and the coverage of words in target generations by different input options.	148
9.2	List of subreddit ids used to construct training data for claim generator.	150
9.3	Automatic evaluation results on both tasks. We report BLEU-2, ROUGE-2, METEOR, and output length. Best scores are in bold. Our DYPLOC model statistically significantly outperforms all baselines and comparisons (randomization approximation test (Noreen, 1989), $p < 0.0005$).	151
9.4	BLEU-2, ROUGE-2, and METEOR results on systems with predicted concepts as input.	154

9.5 Human evaluation results on grammaticality (Gram.), relevance (Rel.), coherence (Coh.), and content richness (Cont.). For each sample, outputs by all three systems are ranked based on the overall preference. We show the percentage each system is ranked as the best. 155

LIST OF FIGURES

3.1	Three different types of arguments used to support the claim “Legally owned guns are frequently stolen and used by criminals”	17
3.2	A typical debate motion consists of a Topic, Claims, and Human Constructed Arguments. Citation article is marked at the end of sentence. Our goal is to find out supporting argument (in <i>italics</i>) from citation article the can back up the given claim.	18
3.3	For each supporting argument type, from left to right shows the percentage of domain names of organizations, scientific, blog, and reference. We do not display statistics for news, because news articles take the same portion in all types (about 50%). . . .	22
3.4	Average features values for different argument types. Numbers in boldface are significantly higher than the others based on paired <i>t</i> -test ($p < 0.05$).	25
4.1	Sample ICLR review excerpts from openreview.net . Propositions are annotated with types, such as FACT (fact), EVAL (evaluation), and REQ (request). Both assigned a rating of 5 (out of 10) for the reviewed paper. Review #2 contains in-depth evaluation and actionable suggestion, thus is perceived to be of a higher quality.	30
4.2	Proposition number in reviews. Differences among venues are all significant except UAI vs. ICLR and ACL vs. NeurIPS ($p < 10^{-6}$, unpaired <i>t</i> -test).	40
4.3	Distribution of proposition type per venue.	40
4.4	Distribution of proposition type per rating (in %) on AMPERE. . . .	41
4.5	Proposition transition probabilities on AMPERE.	41
5.1	Sample user arguments from Reddit Change My View community that argue against original post’s thesis on “government should be allowed to view private emails”. Both arguments leverage supporting information from Wikipedia articles.	46
5.2	Overview of our system pipeline (best viewed in color). Given a statement, relevant articles are retrieved from Wikipedia with topic signatures from statement as queries (marked in red and bold). A reranking module then outputs top sentences as evidence. The statement and the evidence (encoder in gray) are concatenated and encoded as input for our argument generation model. During decoding, the keyphrase decoder first generates talking points as phrases, followed by the argument decoder which constructs the argument by attending both input and keyphrases.	47

5.3	Effect of pre-training on seq2seq-based models. Red dotted line shows the performance of the seq2seq model used for parameter initialization.	59
5.4	Effect of our reranking-based decoder. Beams are reranked at every 5, 10, and 20 steps (p). For each step size, we also show the effect of varying k , where top- k words are selected deterministically for beam expansion, with $10 - k$ randomly sampled over multinomial distribution after removing the k words. Reranking with smaller step size yields better results.	63
5.5	Sample arguments generated by human, our system, and seq2seq trained with evidence. Only the main thesis is shown for the input OP. System generations are manually detokenized and capitalized.	67
6.1	Sample counter-argument for a pro-death penalty statement from Reddit /r/ChangeMyView. The argument consists of a sequence of propositions, by synthesizing opinions and facts from diverse sources. Sentences in italics contain stylistic languages for argumentation purpose.	70
6.2	Architecture of CANDELA. ① Argument Retrieval (§ 6.3): a set of passages are retrieved and ranked based on relevance and stance (§ 6.3.1, § 6.3.3), from which ② a set of keyphrases are extracted (§ 6.3.2), with both as input for argument generation. ③ The biLSTM encoder consumes the input statement and passages returned from step 1. ④ A text planning decoder outputs a representation per sentence, and simultaneously predicts an argumentative function and selects keyphrases to include for the next sentence to be generated (§ 6.4.2). ⑤ A content realization decoder produces the counter-argument (§ 6.4.3).	73
6.3	Average number of distinct n -grams per argument.	87
6.4	Percentage of words in arguments that are not in the top- K ($K = 100, 500, 1000, 2000$) frequent words seen in training. Darker color indicates higher portion of uncommon words found in the arguments.	88
6.5	Sample arguments generated by different systems along with a sample human argument. For our model CANDELA, additionally shown are the keyphrase memory with selected phrases in color, and argumentative filler sentence in italics.	91

7.1	[Upper] Sample counter-argument from Reddit. Argumentative stylistic language for persuasion is in italics. [Bottom] Excerpts from Wikipedia, where sophisticated concepts and language of higher complexity used in the standard version are not present in the corresponding simplified version. Both: key concepts are in bold.	94
7.2	Overview of our framework. The LSTM content planning decoder (§ 7.2.2) first identifies a set of keyphrases from the memory bank conditional on previous selection history, based on which, a style is specified. During surface realization, the hidden states of the planning decoder and the predicted style encoding are fed into the realizer, which generates the final output (§ 7.2.3). Best viewed in color.	95
7.3	Effect of keyphrase selection (F1 score) on generation performance, measured by BLEU and ROUGE. Positive correlations are observed.	110
7.4	Sample outputs for argument generation and Wikipedia generation.	112
8.1	An argument generation example using Reddit ChangeMyView. [Top] Partial output by our planner with keyphrase assignment and positions (in subscripts) for each sentence, segmented by special token [SEN], from which a template is constructed. [Bottom] A draft is first produced and then refined, with updated words highlighted in <i>italics</i>	118
8.2	Content planning with BERT. We use bidirectional self-attentions for input encoding, and apply causal self-attentions for keyphrase assignment and position prediction. The input (\mathbf{x} , \mathbf{m}) and output keyphrase assignments (\mathbf{m}') are distinguished by different segment embeddings.	121
8.3	Our content-controlled text generation framework, PAIR, which is built on BART. Decoding is executed iteratively. At each iteration, the encoder consumes the input prompt \mathbf{x} , the keyphrase assignments \mathbf{m}' , as well as a partially masked template ($\mathbf{t}^{(r-1)}$ for the r -th iteration, [M] for masks). The autoregressive decoder produces a complete sequence $\mathbf{y}^{(r)}$, a subset of which is further masked, to serve as the next iteration’s template $\mathbf{t}^{(r)}$	123
8.4	Results on iterative refinement with five iterations. Both BLEU and ROUGE-L scores steadily increase, with perplexity lowers in later iterations.	132
8.5	End-to-end generation results with automatically predicted content plans. Our models outperform KPSEQ2SEQ in both metrics, except for BLEU-4 on opinion articles where results are comparable.	132

8.6	Distributions of RST tree depth. PAIR _{full} better resembles the patterns in human-written texts.	136
8.7	Discourse markers that are correctly and incorrectly (shaded) generated by PAIR _{full} , compared to aligned sentences in human references. Discourse markers are grouped (from left to right) into senses of CONTINGENCY (higher marker generation accuracy observed), COMPARISON, and EXPANSION. y-axis: # of generated sentences with the corresponding marker.	137
9.1	Sample argument on Reddit ChangeMyView. Our generator considers an input containing (1) a title and (2) an unordered set of content items. Each content item consists of <i>elements</i> of an <i>entity set</i> [ENT], a <i>concept set</i> [CON], and an optional sentence-level <i>claim</i> [CLAIM]. Each output token is generated by conditioning on all content items, and the best aligned ones (learned by our model) are highlighted in corresponding colors. We also underline words that reflect the input concepts and entities. . . .	140
9.2	Our proposed text generation framework, DYPLOC. [Left] For each input content item (a title t , an entity set E_i , and a core concept set C_i), we first expand it with more relevant concepts, i.e., C_i^+ . For sentences to be realized as claims, we employ a separate generator to produce one draft claim, m_i . [Right] The augmented content items, denoted as $\{x_i\}$, are encoded in parallel. At each decoding step, a plan scoring network estimates a distribution $d(x_i y_{<t})$ for all content items and decides on relevant content. A word is predicted based on probabilities marginalized over all content item-conditioned language models, i.e., $p(y_i y_{<t}, x_i)$ for the i -th model.	143
9.3	Sample generations on CMV [Upper] and NYT [Lower]. System generated concepts and claims are in <i>italics</i> . For DYPLOC, we highlight sentence to content item alignment using colors.	158
9.4	The percentage of content items that are aligned to at least one output sentence.	159

CHAPTER 1

INTRODUCTION

1.1 Motivation

Argumentation is perhaps one of the most common modes of communication (Woodson, 1979; Smith, 2003) that we participate on a daily basis. From parliamentary debate that shapes the state policy, to the analytical essays written by students to nourish critical thinking, arguments help to present relevant facts to the target audience, as well as articulate the reasoning process necessary for making better decisions. The amount of textual data on arguments has grown dramatically with the expansion of the Internet and digitized archives. Naturally, the study of arguments has also gain more research interests from the natural language processing (NLP) community (Peldszus and Stede, 2013; Lippi and Torroni, 2016; Cabrio and Villata, 2018; Lawrence and Reed, 2019). The majority of work has focused on understanding the composition of arguments, i.e., the basic component detection (Moens et al., 2007; Levy et al., 2014; Rinott et al., 2015; Eckle-Kohler et al., 2015), classification (Duthie et al., 2016; Eger et al., 2017a; Nguyen and Litman, 2018; Stab et al., 2018), and structural prediction (Persing and Ng, 2016; Niculae et al., 2017). Promising results are achieved in various domains and language styles (Moens et al., 2007; Persing and Ng, 2016; Stab and Gurevych, 2017; Habernal and Gurevych, 2017; Niculae et al., 2017).

Meanwhile, the automatic *construction* of arguments is relatively less explored, although it has great potential to benefit a wide range of applications, such as informing the decision makers of alternative perspectives, and edu-

cating students on debate strategies and essay writing. Existing research can be divided into two main categories: (1) rule-based systems that rely on expert knowledge and argument theories (Reed et al., 1996; Carenini and Moore, 2000; Zukerman et al., 2000). Such methods are usually domain-specific and are costly to extend. (2) Retrieval-based approaches select relevant existing argument snippets from a database (Sato et al., 2015; Le et al., 2018; Hidey and McKeown, 2019). This paradigm requires less manual efforts and scales better to different topics and language styles. Unfortunately, combining multiple retrieved snippets into a fluent and coherent natural language discourse is still an open problem. We identify the following additional challenges which we aim to address in this thesis:

- Generating arguments of a specified stance not only requires sufficient knowledge resource that has to be collected externally, but also a selection method to ensure the correct polarity of them.
- Persuasive arguments usually combine a diverse set of supporting arguments, such as factual evidence, expert opinion, and logical reasoning. They need to be reflected in the collection of knowledge resource, and also combined in an appropriate order for successful delivery.
- Language style plays an important role in argumentation, as they signal the argumentative functions within the discourse. For example, to promote persuasiveness, a claim sentence is usually more affirmative, and use the collective “we” as the subject, so as to demonstrate attention of the listener as well.
- The underlying neural text generation model is difficult to control, as it only incurs loss over the sequential token prediction. This is particularly

the issue for recent pre-trained Transformers with large and complicated architectures. Yet controllability is crucial to enforce the inclusion and ordering of the external knowledge.

In order to tackle the above challenges, we present a variety of model architecture designs as well as training techniques. As a crucial component of modern NLP pipelines, high-quality datasets and annotation on arguments are also collected to support our research. We also adopt our systems over a wide range of distinct domains and different task formulations, to showcase their usefulness for neural text generation in general. The overview of each chapter and the main contributions are summarized below.

1.2 Thesis Outline and Contributions

- In Chapter 3, we investigate argument component classification and support relation detection. We first exemplify the two tasks with an online debate forum. Then, an argument scheme tailored for this domain is described in detail. *We annotate and release a new dataset with labels on argument types and supporting arguments from the cited documents.* Based on this dataset, we propose feature-based models for argument type prediction and support relation prediction.
- In Chapter 4, we make efforts to understand arguments in academic peer reviews, a distinct domain that has not been studied from an argument-mining perspective before. *We collect review data from a prominent machine learning conference and annotate argument boundaries, as well as label their types.* We benchmark this new dataset using state-of-the-art models. Fi-

nally, we run the model over a large set of reviews from different venues and characterize the distinct properties.

- In Chapter 5, we present an argument generation system that utilizes externally retrieved evidence. We first briefly introduce the system pipeline, then describe a newly collected counter-argument generation dataset from Reddit ChangeMyView forum. *This dataset is the first of its kind to facilitate research on argument generation. We also propose a novel neural network based model, which encodes relevant retrieval snippets and generates the argument as well as keyphrase talking points.* Both automatic metrics and human evaluation are conducted to assess our model efficacy.
- In Chapter 6, we introduce argument generation with text planning. We motivate this Chapter by highlighting the importance of text planning using sample arguments and numerous retrieval results. It is followed by the formulation of task stages as well as data source used for retrieval. Then, *we formally introduce our proposed CANDELA model, which first selects sentence-level keyphrases using a LSTM decoder. Conditioned on the selection results, the realization decoder produces the natural language output.* CANDELA compares favorably against existing method and non-trivial baselines, confirming the benefits of the text planning step.
- In Chapter 7, we extend the idea of text planning to build a novel generic text generation framework, with the overall goal of promoting controllability for neural generation models. *We propose improved content selection mechanism that is aware of past selection results. Our decoder is also equipped with a dedicated style module, that further informs the realization decoder of the desired output style.* We conduct extensive experiments on three distinct tasks to demonstrate the model effectiveness. Our analysis reveals a

strong positive correlation between content selection and output quality, further validating our design choice.

- In Chapter 8, we study the content-control for large pre-trained Transformer models. We design **PAIR**, a generic text generation framework that utilizes the masked language model as a way to represent template, which is predicted from a separate content planner. To ensure the generation quality, we introduce an iterative refinement strategy to decode the output sequence in multiple rounds. We showcase this framework over three distinct domains
- In Chapter 9, we propose a novel end-to-end text generation framework called DYPLOC, which incorporates diverse input content and automatically determines the ordering and realization of these inputs using a mixed language model. Each of the input content item consists of entities, concepts, and claims, encompassing different aspects of the desired semantic constraints. Our model can be directly built on pre-trained Transformer to generate fluent output. Experiments on two opinion text generation datasets indicate our model outperforms competitive comparisons with the same input.
- In Chapter 11, we conclude this thesis and discuss future directions.

CHAPTER 2

BACKGROUND

2.1 Argument Mining

Analysis of arguments using computational methods have recently attracted significant research interests. In this section, we first introduce prominent approaches to argument structure understanding (Section 2.1.1). We next discuss argument search engines that enable customized argument retrieval (Section 2.1.2). We then highlight existing work on argument generation (Section 2.1.3). For a more detailed and thorough review, we refer the readers to the surveys by [Peldszus and Stede \(2013\)](#); [Cabrio and Villata \(2018\)](#); [Lawrence and Reed \(2019\)](#).

2.1.1 Argument Structure Extraction

Argument mining research aims to detect and recognize the argumentative discourse units and their interactions in textual data. Existing work typically divides the pipeline into three stages. (1) *Argument unit segmentation* first identifies the token-level boundaries for argumentative propositions and the non-argumentative segments. This task is usually treated as either a sequence tagging problem ([Ajjour et al., 2017](#); [Eger et al., 2017b](#); [Stab and Gurevych, 2017](#)) or sentence classification ([Moens et al., 2007](#); [Biran and Rambow, 2011](#); [Stab and Gurevych, 2014](#); [Levy et al., 2014](#)). Our work in Chapter 4 shows the LSTM-based sequence tagging model enhanced with ELMo embeddings ([Peters et al., 2018](#)) achieves satisfactory results, with over 0.81 F1 scores on arguments in

the peer-review domain. (2) *Argument component classification* then labels each argument segment into different types, according to a predefined scheme. Feature engineering has been exploited to tackle this task, with promising results in persuasive essays (Persing and Ng, 2016; Stab and Gurevych, 2017; Nguyen and Litman, 2018; Persing and Ng, 2020), online forums (Niculae et al., 2017; Hua and Wang, 2017), and legal (Palau and Moens, 2009; Teruel et al., 2018) and scientific (Teufel et al., 2009; Lauscher et al., 2018) texts. (3) *Argument structure recognition* further predicts the support and attack relations among the argumentative units. Early work relies on hand-crafted features that are customized for specific domains (Peldszus and Stede, 2013; Nguyen and Litman, 2016; Stab and Gurevych, 2017). Recently, neural network based models have shown promising results (Eger et al., 2017b; Chakrabarty et al., 2019; Mayer et al., 2020). However, most of them focus on relation detection within the same document. In Chapter 3, we examine the usage of support in external documents, and examine the benefits of argument component types for structure prediction.

2.1.2 Argument Retrieval

Argument retrieval is investigated to deliver relevant arguments for user-specified queries. Prior work (Levy et al., 2014; Rinott et al., 2015) has studied evidence retrieval from Wikipedia for given claims. Wachsmuth et al. (2017b) build a more generic search engine indexed with 291,440 arguments collected from various online debate portals. By contrast, Stab et al. (2018) focus on argument search over heterogeneous web sources from CommonCrawl.¹ For both systems, the key challenges lie in the accurate assessment of argument topics

¹<http://commoncrawl.org/>

and stances, as well as ranking large volume of arguments given the input query, highlighting the necessities for developing efficient argument analysis tools as we described in the previous section. In our work (Chapter 5 and Chapter 6), argument retrieval are utilized as a pre-processing step to supply relevant external resource for the generation model. We show that the retrieval results can substantially improve the generation quality.

2.1.3 Argument Generation

Traditional approaches to automatically generating arguments mainly rely on rule-based systems, e.g., designing operators that reflect strategies from argumentation theory (Reed et al., 1996; Carenini and Moore, 2000; Zukerman et al., 2000). For instance, Reed (1999) describes the first full natural language argument generation system, called Rhetorica. These systems can output highly accurate and controlled arguments, but are limited to specific topics and domains. Information retrieval systems are recently deployed to extract existing arguments relevant to a debate motion (Sato et al., 2015). Although content ordering has been proposed for argument synthesis (Reisert et al., 2015; Yanase et al., 2015), the output arguments are usually a collection of sentences from heterogeneous information sources, thus lacking coherence and conciseness. Our work aims to close the gap by generating eloquent and coherent arguments, assisted by an argument retrieval system. It is also worth mentioning that, concurrent to our research, IBM has showcased an automated agent that can carry out complex debate against human experts (Slonim et al., 2021). Their architecture involves several manually defined modules such as argument detection, argument knowledge base, and a rule-based argument constructor, which are

carefully engineered to tackle the oxford-style format. Whereas our focus is to design an end-to-end solution for paragraph-level argument generation without manual efforts.

2.2 Text Generation

Generating coherent text that satisfies certain input constraints is a central task in natural language processing. The goal of this dissertation is to automatically produce arguments with user specified topic and stance. The underlying generation stack is thus crucial to ensure the output quality. In this section, we first introduce the high-level overview of the typical text generation pipeline (Section 2.2.1). We then discuss the recent progress achieved by neural network based generation models (Section 2.2.2), which is followed by existing work on knowledge-enhanced generation (Section 2.2.3).

2.2.1 Text Generation Pipeline

Traditionally, text generation is divided into multiple subtasks (McKeown, 1985; Reiter and Dale, 2000) including **content selection**, which determines which information to include (“what to say”), **text planning**, that organizes the selected information in order and allocates them in sentences (“when to say what”), and **surface realization**, which chooses the appropriate lexical items and expressions to reflect the text plan (“how to say it”).

Early work for content selection and text planning mainly relies on rules based on discourse theory (Scott and de Souza, 1990; Hovy, 1993), expert knowl-

edge (Reiter et al., 2000), or trained by statistical models (Duboue and McKeown, 2001, 2003). Although these methods provide interpretable selection results, they are difficult to scale to new domains. Recently, data-driven approaches have received growing interests. Barzilay and Lee (2004) characterize content change as topic shift, and adopt a distributional approach based on Hidden Markov Models (HMM) to capture the information presentation order, which achieves substantial improvements over a competitive feature-based comparison model. Barzilay and Lapata (2005) treats content selection as binary classification of database items for inclusion in the given context. Crucially, they propose a collective classification method that considers multiple related items simultaneously, which yields more coherent output compared to the context-agnostic model. Following these two ideas, our proposed text generation systems (Chapter 6 and Chapter 7) are designed to learn the shift of content as well as capture the dependencies among content items.

2.2.2 Neural Text Generation Models

In the past decade, the sequence-to-sequence (seq2seq) encoder-decoder paradigm has become the dominant approach for text generation models. The input and output, which are of varied lengths, are treated as sequences of symbols and passed into either the recurrent neural network (RNN) or Transformer models (Vaswani et al., 2017). The former is usually instantiated as the Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997), or the Gated Recurrent Unit (GRU) (Cho et al., 2014), with the attention mechanism (Bahdanau et al., 2015; Luong et al., 2015b) to allow direct access to the encoder states during generation. The Transformer framework (Vaswani et al., 2017) in-

stead relies on multi-headed self-attention and stacked layers to learn the input representation as well as the output language model. An important advantage of seq2seq framework is that it can be end-to-end trained by maximizing log-likelihood, thus requires no manual efforts on rule crafting or feature engineering.

The seq2seq framework is first proposed for machine translation (Sutskever et al., 2014), and then achieves state-of-the-art performance for a wide range of text generation tasks, such as abstract summarization (Rush et al., 2015; Nallapati et al., 2016), chatbot (Vinyals and Le, 2015; Sordoni et al., 2015), and data-to-text generation (Gehrmann et al., 2018). In this dissertation, we employ this framework for argument generation, which substantially differing from other tasks in that additional output content needs to be inferred and realized besides what is provided in the input, and there also exists multiple plausible arguments for the same input.

2.2.3 Knowledge-Enhanced Text Generation

As mentioned earlier, unlike standard text generation tasks, for argument generation, the input information alone is usually insufficient to determine the output arguments. Moreover, world-knowledge that are pertinent to the arguments is often implicit. For example, given the input *“death penalty is more rational than life in prison”*, a human written counter-argument contains *“[W]e will never have a perfect justice system. Unreliable evidence is used when there is no witnesses, which could result in wrongful convictions”*. In order to generate this argument, the system needs to infer that death penalty is a result of legal conviction, which might

be wrong if there is no witnesses. To tackle this challenge, we enable the generation system with external knowledge that are orders of magnitude larger than the training data, to better inform the model of necessary knowledge (Chapter 5 and Chapter 6).

More broadly, knowledge-enhanced text generation aims to incorporate different types of knowledge to the generation pipeline. The most common methods utilize existing knowledge base with entities and relations, such as Wikipedia and ConceptNet (Speer et al., 2017), in order to provide common-sense and factual events that are relevant to the generation. Experiments have been conducted on question answering (Long et al., 2017; Mihaylov and Frank, 2018; Bi et al., 2019), summarization (Fan et al., 2019a; Huang et al., 2020; Gunel et al., 2020), and dialogue generation (Zhou et al., 2018; Lian et al., 2019). These systems generally consist of a knowledge retriever that selects the relevant facts, which is then linearized and encoded with the seq2seq framework to inform the decoder. In this dissertation, we utilize passage-level retrieval over Wikipedia, and extract keyphrases from the passages to carry the salient information. We show that this design choice is both flexible and expressive, as the system is able to handle arbitrary combinations of keyphrases and generate fluent output.

2.3 Controllable Generation

Neural generation models are capable to generate fluent output. But since they are fully end-to-end trained and conflate the text planning and realization stages, it is difficult to control certain content to be mentioned in specified location. Consequently, they are prone to factual errors (Wiseman et al., 2017;

Rohrbach et al., 2018) and even toxic output (Holtzman et al., 2019; Sheng et al., 2019). In this section, we first discuss what are the dimensions researchers aim to control (Section 2.3.1), and how controllability are viewed as conditional language model (Section 2.3.2). Finally, we describe controlled generation using refinement (Section 2.3.3).

2.3.1 What Aspects Do We Need to Control?

As we described in Section 2.2.1, text generation systems can be modeled in different stages. Correspondingly, the level of control can also range from enforcing the high-level content and style (Hu et al., 2017; Fidler and Goldberg, 2017; Fu et al., 2018), semantics (Wen et al., 2015; Elder et al., 2018), to manipulate the low-level syntactic structure (Dušek and Jurčiček, 2016; Iyyer et al., 2018; Goyal and Durrett, 2020). Specific applications encourage the model to cover a given topic (Wang et al., 2017a; See et al., 2019), mention specified entities (Fan et al., 2018a), or display a certain attribute (Hu et al., 2017; Luo et al., 2019; Balakrishnan et al., 2019). However, most existing work relies on model engineering, limiting the generalizability to new domains and adaptability to large pre-trained Transformers. One exception is the Plug and Play model (Dathathri et al., 2020), which directly modifies the key and value states of GPT-2 (Radford et al., 2019). However, since the signal is derived from the whole generated text, it is too coarse to provide precise sentence-level content control. In Chapter 7 and Chapter 8, we instead gain fine-grained controllability through keyphrase assignment and positioning per sentence, which can be adapted to any off-the-shelf pre-trained Transformer generators.

2.3.2 Controlled Generation by Conditional Language Model

A straightforward way to achieve controllability is to train the neural generator as a conditional language model. In addition to the input, the system is supplied with a control variable, the generation probability is then conditioned on both these two elements. Early work has adopted this approach for image captioning (Kiros et al., 2014; Vinyals et al., 2015b), dialogue generation (Wen et al., 2016; Zhang et al., 2019), and review generation (Tang et al., 2016; Fidler and Goldberg, 2017). The control variable typically takes the form of a dense vector that participates in the calculation of decoder states, and it is trained together with the seq2seq model in an end-to-end fashion.

A more recent line of research has explored conditioning over symbolic information. Most notably, it has been shown that first generating sequences of entities and their semantic relations, and then conditioning on them yields more coherent and diverse output in story generation (Fan et al., 2019b; Yao et al., 2019a; Goldfarb-Tarrant et al., 2020). Our proposed text generation models (Chapter 6, Chapter 7, Chapter 9) also follow this setting. The predicted keyphrase selection results for each sentence are treated as the conditional variables. They are either directly fed into the decoder states, or accessed through attention mechanism, to reflect the desired control.

2.3.3 Controlled Generation by Editing and Refinement

So far the neural generation models we discussed are auto-regressive in nature, i.e., the output is generated in one-shot as a sequence of tokens. An alternative approach is to run the generator in multiple passes, allowing modifications over

already generated tokens. This framework has also been leveraged to improve controllability over the generation. [Li et al. \(2018\)](#) present a text attribute transfer system that first identifies attribute markers (words need to be edited), and then retrieves sentences with target attributes (e.g., desired sentiment word). Finally, an RNN generates the output given the edits and original sentence. [Guu et al. \(2018\)](#) study a more generic unconditional sentence generation task, where they first draw a random sentence (prototype) from the training corpus, together with a random editing vector. Analyses reveal that the prototype-then-edit framework yields more fluent output and achieves fine-grained control over the semantic space. More broadly, the non-autoregressive generation has been used in machine translation ([Lee et al., 2018](#); [Freitag et al., 2019](#); [Mansimov et al., 2019](#); [Kasai et al., 2020](#)) to gradually improve the generation quality. It is further popularized by the emergence of large pre-trained Transformers ([Devlin et al., 2019](#); [Liu et al., 2019](#)), which are based on the masked language model. [Lawrence et al. \(2019\)](#) employ the special placeholder token to leverage the pre-trained BERT ([Devlin et al., 2019](#)) model for text generation. The output is initialized with all placeholders, which are then iteratively replaced with tokens from the output vocabulary. Their system outperforms competitive baselines over two conversational tasks. In [Chapter 8](#), we introduce a refinement based generation framework based on the seq2seq BART ([Lewis et al., 2020a](#)), offering better generalizability and stronger capacity for long text generation. Our proposed strategy substantially differs from prior solutions that rely on in-place word substitutions ([Novak et al., 2016](#); [Xia et al., 2017](#); [Weston et al., 2018](#)), as we leverage the seq2seq architecture to offer more flexible edits.

CHAPTER 3

UNDERSTANDING ARGUMENT TYPES AND SUPPORT RELATIONS

In this chapter, we propose an argument type scheme and discuss the automatic classification of argument sentences from an online debate portal. We further introduce the support argument detection task, which aims to pinpoint a sentence that best supports a given argument, from an external document. This work is published at ACL 2017 (Hua and Wang, 2017). The relevant resources can be found at: <http://xinyuhua.github.io/Resources>.

3.1 Introduction

Argumentation plays a crucial role in persuasion and decision-making processes. An argument usually consists of a central claim (or conclusion) and several supporting premises. Constructing arguments of high quality would require the inclusion of diverse information, such as factual evidence and solid reasoning (Rieke et al., 1997; Park and Cardie, 2014). For instance, as shown in Figure 3.1, the editor on idebate.org – a Wikipedia-style website for gathering pro and con arguments on controversial issues, utilizes arguments based on study, factual evidence, and expert opinion to support the anti-gun claim “legally owned guns are frequently stolen and used by criminals”. However, it would require substantial human effort to collect information from diverse resources to support argument construction. In order to facilitate this process, there is a pressing need for tools that can automatically detect supporting arguments.

Study	A June 2013 IOM report states that “almost all guns used in criminal acts enter circulation via initial legal transaction”.
Factual	Between 2005 and 2010, 1.4 million guns were stolen from US homes during property crimes (including burglary and car theft), a yearly average of 232,400.
Expert opinion	Ian Ayres, JD, PhD, . . . states, “with guns being a product that can be easily carried away and quickly sold at a relatively high fraction of the initial cost, the presence of more guns can actually serve as a stimulus to burglary and theft.”

Figure 3.1: Three different types of arguments used to support the claim “Legally owned guns are frequently stolen and used by criminals”.

To date, most of the argument mining research focuses on recognizing argumentative components and their structures from constructed arguments based on curated corpus (Mochales and Moens, 2011; Stab and Gurevych, 2014; Feng and Hirst, 2011; Habernal and Gurevych, 2015; Nguyen and Litman, 2016). Limited work has been done for retrieving supporting arguments from external resources. Initial effort by Rinott et al. (2015) investigates the detection of relevant factual evidence from Wikipedia articles. However, it is unclear whether their method can perform as good on documents of different genres (e.g. news articles vs. blogs) for detecting distinct types of supporting information.

In this work, *we present a novel study on the task of sentence-level supporting argument detection from relevant documents for a user-specified claim.* Take Figure 3.2 as an example: assume we are given a claim on topic of “banning cosmetic surgery” and a relevant article (cited for argument construction), we aim to automatically pinpoint the sentence(s) (in *italics*) among all sentences in cited article, that can be used to back up the claim. We define such tasks as *supporting argument detection*. Furthermore, another goal of this work is to understand and characterize different types of supporting arguments. Indeed, human edi-

Topic: This house would ban cosmetic surgery
Claim: An outright ban would be easier than the partial bans that have been enacted in some places.
Human Constructed Argument: ...This potentially leaves difficulty drawing the line for what is allowed.[1] ...

Citation Article
[1]: "Australian State Ban Cosmetic Surgery for Teens"
- ...It is unfortunate that a parent would consider letting a 16-year-old daughter have a breast augmentation."
- *But others worry that similar legislation, if it ever comes to pass in the United States, would draw a largely arbitrary line – and could needlessly restrict some teens from procedures that would help their self-esteem.*
- Dr. Malcolm Z. Roth, director of plastic surgery at Maimonides Medical Center in Brooklyn, N.Y. , said he believes that some teens are intelligent and mature enough to comprehend the risks and benefits of cosmetic surgery...

Figure 3.2: A typical debate motion consists of a Topic, Claims, and Human Constructed Arguments. Citation article is marked at the end of sentence. Our goal is to find out supporting argument (in *italics*) from citation article the can back up the given claim.

tors do use different types of information to promote persuasiveness as we will show in Section 3.2. Prediction performance also varies among different types of supporting arguments.

Given that none of the existing datasets is suitable for our study, we collect and annotate a corpus from Idebate, which contains hundreds of debate topics and corresponding claims. As is shown in Figure 3.2, each claim is supported with some human constructed argument, with cited articles marked on sentence level. After careful inspection on the supporting arguments, we propose to label them as STUDY, FACTUAL, OPINION, or REASONING. Substantial inter-annotator agreement rate is achieved for both supporting argument labeling (with Cohen's κ of 0.8) and argument type annotation, on 200 topics with 621 reference articles.

Based on the new corpus, we first carry out study on characterizing arguments of different types via type prediction. We find that arguments of STUDY and FACTUAL tend to use more concrete words, while arguments of OPINION contain more named entities of person names. We then investigate whether argument type can be leveraged to assist supporting argument detection. Experimental results based on LambdaMART (Burgess, 2010) show that utilizing features composite with argument types achieves a Mean Reciprocal Rank (MRR) score of 57.65, which outperforms an unsupervised baseline and the same ranker trained without type information. Feature analysis also demonstrates that salient features have significantly different distribution over different argument types.

3.2 Data and Annotation

We rely on data from idebate.org, where human editors construct paragraphs of arguments, either supporting or opposing claims under controversial topics. We also extract textual citation articles as source of information used by editors during argument construction. In total we collected 383 unique debates, out of which 200 debates are randomly selected for study. After removing invalid ones, our final dataset includes 453 claims and 629 citation articles with 17,910 sentences.

Annotation Process. As shown in Figure 3.2, we first annotate which sentence(s) from a citation articles is used by the editor as supporting arguments. Then we annotate the type for each of them as STUDY, FACTUAL, OPINION, or REASONING, based on the scheme in Table 3.1. For instance, the highlighted

Type	Explanation	Example
STUDY	Results and discoveries, usually quantitative, as a result of some research investment. For example, results of some experiments or poll.	<i>A World Bank survey in 2011 showed that about 40% of those who join rebel movements say they are motivated by a lack of jobs.</i>
FACTUAL	Description of some occurred events or facts, or chapters in law or declaration. Usually can be obtained without research investment. For example, description of objective environment, issuance of law, historical events.	<i>The US trade deficit widened to £50.2 bn in May, the highest level for 31 months.</i>
OPINION	Quotes from some person or group, either direct or indirect. It usually contains subjective, judgemental and evaluative languages, and might reflect the position or stance of someone. For example, comments on laws and policies from officials, speculation or prediction on stock markets.	<i>“The Tea Party movement is interesting in that there is a combination of localism, nativism and populism that we’ve seen at various points in America, ” said Nathaniel Persily, a law Professor at Columbia.</i>
REASONING	Logical structures. It usually can be further broken down into causal or conditional substructures. For example, to explain how oil extraction could break ecosystem by giving causal chains and their effects.	<i>Obviously, the light from these objects needed billions of years to reach us, and therefore the universe has to be billions of years old.</i>

Table 3.1: Annotation scheme and examples for our dataset.

supporting argument in Figure 3.2 is labeled as REASONING.

Two experienced annotator were hired to identify supporting arguments by reading through the whole cited article and locating the sentences that best match the reference human constructed argument. This task is rather complicated since human do not just repeat or directly quote the original sentences from citation articles, they also paraphrase, summarize, and generalize. For instance, the original sentence is “The global counterfeit drug trade, a billion-

dollar industry, is thriving in Africa”, which is paraphrased to “This is exploited by the billion dollar global counterfeit drug trade” in human constructed argument.

Multiple rounds of annotation were carried out. First round was a pilot annotation where each annotator was assigned 30 topics. The annotators were asked to annotate independently, then discuss and resolve disagreements and give feedback about current scheme. This process continued until 200 topics were finished. Each time we add details to scheme guidelines and revise scheme if necessary. We compute inter-annotator agreement based on Cohen’s κ for both supporting arguments labeling and argument type annotation. For supporting arguments we have high degree of consensus, with Cohen’s κ score ranges from 0.76 to 0.83 in all rounds and 0.80 overall. For most type annotation we achieve reasonably good agreements, except for type REASONING which is 0.29 overall. This is because many times annotators have different interpretations on REASONING, and frequently label it as OPINION.

Statistics. In total 1107 sentences are identified as supporting arguments, which account for 6.18% of all sentences in the dataset. Among those, 108 (9.76%) are labeled as STUDY, 575 (51.94%) as FACTUAL, 382 (34.51%) as OPINION, and 42 (3.79%) as REASONING. The statistics are listed in Table 3.2.

We further analyze the source of the supporting arguments. Domain names of the citation articles are collected based on their URL, and then categorized into “news”, “organization”, “scientific”, “blog”, “reference”, and others, according to a taxonomy provided by Alexa¹ with a few edits to fit our dataset. News articles are the major source for all types, which account for roughly 50%

¹<http://www.alexa.com/topsites/category>

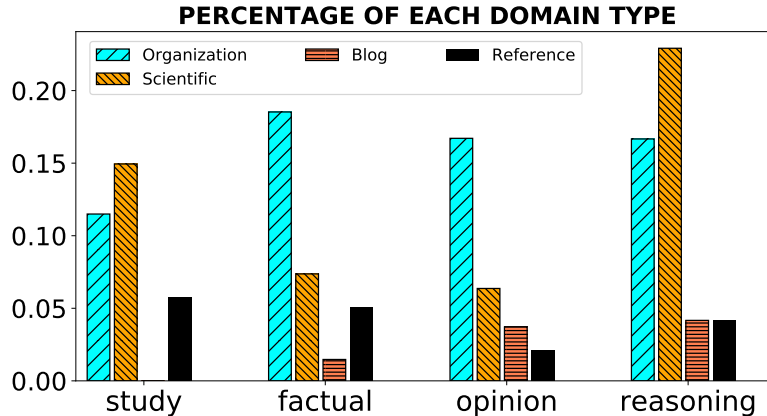


Figure 3.3: For each supporting argument type, from left to right shows the percentage of domain names of organizations, scientific, blog, and reference. We do not display statistics for news, because news articles take the same portion in all types (about 50%).

	STUDY	FACTUAL	OPINION	REASONING
# Sentence	95	497	363	40
Cohen’s κ	0.61	0.75	0.71	0.29

Table 3.2: The statistics of four types on our annotated dataset.

for each. We show the distribution of other four types in Figure 3.3. Arguments of STUDY and REASONING are mostly from “scientific” websites (14.9% and 22.9%), whereas “organization” websites contribute a large portion of arguments of FACTUAL (18.5%) and OPINION (16.7%).

3.3 A Study on Argument Type Prediction

Here we characterize arguments of different types based on diverse features under the task of predicting argument types. Supporting arguments identified from previous section are utilized for experiments. We also leverage the learned classifier in this section to label the sentences that are not supporting arguments,

which will be used for supporting argument detection in the next section. Four major types of features are considered.

Basic Features. We calculate frequencies of unigram and bigram words, number of four major types of part-of-speech tags (verb, noun, adjective, and adverb), number of dependency relations, and number of seven types of named entities (Chinchor and Robinson, 1997).

Sentiment Features. We also compute number of positive, negative and neutral words in MPQA lexicon (Wilson et al., 2005), and number of words from a subset of semantic categories from General Inquirer (Stone et al., 1966). Specifically, we used the following categories: Strong, Weak, Virtue, Vice, Ovrst (Overstated), Undrst (Understated), Academ (Academic), Doctrin (Doctrine), Econ (Economic), Relig (Religious), Causal, Ought, Perceiv (Perception).

Discourse Features. We use number of discourse connectives from the top two levels of Penn Discourse Tree Bank (Prasad et al., 2008).

Style Features. We measure word attributes for their concreteness (perceptible vs. conceptual), valence (or pleasantness), arousal (or intensity of emotion), and dominance (or degree of control) based on the lexicons collected by Brysbaert et al. (2014) and Warriner et al. (2013).

We utilize Log-linear model for argument type prediction with one-vs-rest setup. Three baselines are considered: (1) random guess, (2) majority class, and (3) unigrams and bigrams as features for Log-linear model. Identified supporting arguments are used for experiments, and divided into training set (50%), validation set (25%) and test set (25%). From Table 3.3, we can see that Log-linear model trained with all features outperforms the ones trained with ngram

	Accuracy	Micro-F1
Majority class	0.520	0.171
Random	0.240	0.199
Log-linear (ngrams features)	0.535	0.277
Log-linear (all features)	0.622	0.436

Table 3.3: One-vs-rest classification results.

features. To further characterize arguments of different types, we display sample features with significant different values in Figure 3.4. As can be seen, arguments of STUDY and FACTUAL tend to contain more concrete words and named entities. Arguments of OPINION mention more person names, which implies that expert opinions are commonly quoted.

Classification results show that Log-linear model trained with all features as one-vs-rest classifier has accuracy of 0.622, outperforming a Log-linear model trained with ngram features (unigrams and bigrams) with accuracy of 0.535. To further characterize arguments of different types, we display sample features with significant different values in Figure 3.4. As can be seen, arguments of STUDY and FACTUAL tend to contain more concrete words and named entities. Arguments of OPINION mention more person names, which implies that expert opinions are commonly quoted. And REASONING has more words used in causal and conditional structures, which accords with our expectation.

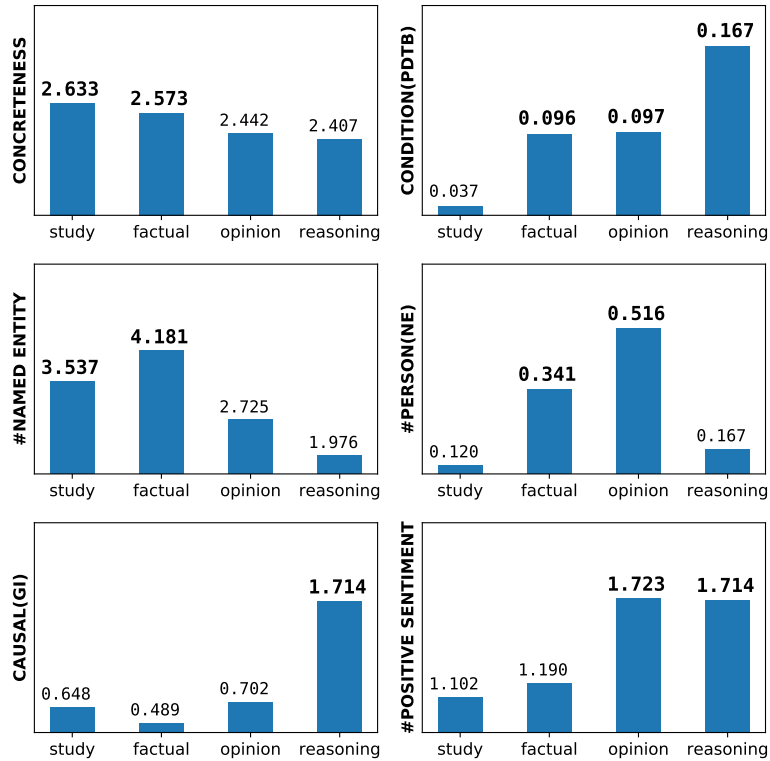


Figure 3.4: Average features values for different argument types. Numbers in **boldface** are significantly higher than the others based on paired t -test ($p < 0.05$).

3.4 Supporting Argument Detection

We cast the sentence-level supporting argument detection problem as a ranking task.² Features in Section 3.3 are also utilized here as “Sentence features” with additional features considering the sentence position in the article. We further employ features that measure similarity between claims and sentences, and the composite features that leverage argument type information.

Similarity Features. We compute similarity between claim and candidate

²Many sentences in the citation article is relevant to the topic to various degrees. We focus on detecting the most relevant ones, and thus treat it as a ranking problem instead of a binary classification task.

sentence based on TF-IDF and average word embeddings. We also consider ROUGE (Lin, 2004), a recall oriented metric for summarization evaluation. In particular, ROUGE-L, a variation based on longest common subsequence, is computed by treating claim as reference and each candidate sentence as sample summary. In similar manner we use BLEU (Papineni et al., 2002), a precision oriented metric.

Composite Features. We adopt composite features to study the interaction of other features with type of the sentence. Given claim c and sentence s with any feature mentioned above, a composite feature function $\phi_{M(\text{type}, \text{feature})}(s, c)$ is set to the actual feature value if and only if the argument type matches. For instance, if the ROUGE-L score is 0.2, and s is of type STUDY, then $\phi_{M(\text{study}, \text{ROUGE})}(s, c) = 0.2$. $\phi_{M(\text{factual}, \text{ROUGE})}(s, c)$, $\phi_{M(\text{opinion}, \text{ROUGE})}(s, c)$, $\phi_{M(\text{reasoning}, \text{ROUGE})}(s, c)$ are all set to 0.

We choose LambdaMART (Burgess, 2010) for experiments, which is shown to be successful for many text ranking problems (Chapelle and Chang, 2011). Our model is evaluated by Mean Reciprocal Rank (MRR) and Normalized Discounted Cumulative Gain (NDCG) using 5-fold cross validation. We compare to TFIDF and Word embedding similarity baselines, and LambdaMART trained with ngrams (unigrams and bigrams).

Results in Table 3.4 show that using composite features with argument type information (Comp(type, Sen) + Comp(type, Sim)) can improve the ranking performance. Specifically, the best performance is achieved by adding composite features to sentence features, similarity features, and ngram features. As can be seen, supervised methods outperform unsupervised baseline methods. And similarity features have similar performance as those baselines. The best per-

	MRR	NDCG
Baselines		
TF-IDF similarity	45.48	56.48
word2vec similarity	47.65	59.00
Ngrams	27.26	43.83
Separate feature sets		
Sentence (Sen)	55.38	65.09
Similarity (Sim)	43.13	55.16
Comp(type, Sen) + Comp(type, Sim)	55.75	64.91
Additive feature sets		
Sen + Ngrams + Sim	56.43	65.79
+ Comp(type, Sen) + Comp(type, Sim)	57.65	66.51
+ Comp(type, Claim)	56.58	65.68

Table 3.4: Supporting argument detection results. “Sentence” stands for features discussed in section 4 for modeling candidate sentences. “Similarity” refers to similarity features. Comp(type, Sen) stands for composite features of argument type and sentence features, similarly for Comp(type, Sim). Comp(type, Claim) represents composite features of type and claim features. Results that are statistically significantly better than all three baselines are marked with * (paired t -test, $p < 0.05$).

formance is achieved by combination of sentence features, N-grams, similarity, and two composite types, which is boldfaced. Feature sets that significantly outperform all three baselines are marked with *.

For feature analysis, we conduct t -test for individual feature values between supporting arguments and the others. We breakdown features according to their argument types and show top salient composite features in Table 3.5. For all sentences of type STUDY, relevant ones tend to contain more “percentage” and more concrete words. We also notice those sentences with more hedging words are more likely to be considered. For sentences of FACTUAL, position of sentence in article plays an important role, as well as their similarity to the claim based on ROUGE scores. For type OPINION, unlike all other types, position of sentence seems to be insignificant. As we could imagine, opinionated informa-

Feature	STUDY	FACTUAL	OPINION	REASONING
# PERC, NE	** ↑↑↑↑	–	–	–
# LOC, NE	–	** ↑↑	–	** ↑
position of sentence	** ↓↓	* * * ↓↓	–	* * * ↓↓↓↓
concreteness of sentence	* * * ↑↑	–	** ↑↑	* * * ↓
arousal of sentence	* * * ↑↑	–	** ↑↑	** ↓
# hedging word	** ↑↑↑	–	–	–
ROUGE-L	* * * ↑↑	* * * ↑	** ↑↑	–
concreteness of claim	* * * ↑↑	–	** ↑↑	* * * ↓
arousal of claim	* * * ↑↑	–	** ↑↑	* * * ↓

Table 3.5: Comparison of feature significance under composition with different types. The number of * stands for the p -value based on t -test between supporting argument sentences and the others after Bonferroni correction. From one * to four, the p -value scales as: 0.05, 1e-3, 1e-5, and 1e-10. When mean value of supporting argument sentences is larger, ↑ is used; otherwise, ↓ is displayed. Number of arrows represents the ratio of the larger value over smaller one. “–” indicates no significant difference.

tion might scatter around the whole documents. For sentences of REASONING, the ones that can be used as supporting arguments tend to be less concrete and less emotional, as opposed to opinion.

3.5 Conclusion

We presented a novel study on the task of sentence-level supporting argument detection from relevant documents for a user-specified claim. Based on our newly-collected dataset, we characterized arguments of different types with a rich feature set. We also showed that leveraging argument type information can further improve the performance of supporting argument detection.

CHAPTER 4

ARGUMENT MINING ON PEER REVIEWS

In the previous chapter, we discussed a general domain argument mining framework over an online debate portal. In practice, specific argument scheme is often necessary to better capture the unique patterns for specific domains. In this chapter, we focus on arguments present in academic peer reviews. We outline steps for the schematic design, the annotation procedure, and various modeling choices for automatic argument extraction and classification.

This work is published at NAACL 2019 (Hua et al., 2019b). This is a joint work with Mitko Nikolov, Nikhil Badugu, and Lu Wang. Relevant resources can be found at: <http://xinyuhua.github.io/Resources/naacl19/>.

4.1 Introduction

Peer review is a process where domain experts scrutinize the quality of research work in their field, and it is a cornerstone of scientific discovery (Hettich and Pazzani, 2006; Kelly et al., 2014; Price and Flach, 2017). In 2015 alone, approximately 63.4 million hours were spent on peer reviews (Kovanis et al., 2016). To maximize their benefit to the scientific community, it is crucial to understand and evaluate the construction and limitation of reviews themselves. However, minimal work has been done to analyze reviews' content and structure, let alone to evaluate their quality.

As seen in Figure 4.1, peer reviews resemble arguments: they contain **argumentative propositions** (henceforth propositions) that convey reviewers' in-

Review #1 (rating: 5, number of sentences: 11)

[Quality: This paper demonstrates that convolutional and relational neural networks fail to solve visual relation problems ...]**FACT** [This points at important limitations of current neural network architectures where architectures depend mainly on rote memorization.]**EVAL** ... [Significance: This work demonstrates failures of relational networks on relational tasks...]**FACT**

[Pros: Important message about network limitations.]**EVAL** [Cons: Straightforward testing of network performance on specific visual relation tasks.]**EVAL** ...

Review #2 (rating: 5, number of sentences: 10)

[The authors present two autoregressive models ...]**FACT** ... [In that context, this work can be viewed as applying deep autoregressive density estimators to policy gradient methods.]**EVAL** ... [At least one of those papers ought to be cited.]**REQ** [It also seems like a simple, obvious baseline is missing from their experiments ...]**EVAL** ...

[The method could even be made to capture dependencies between different actions by adding a latent probabilistic layer ...]**EVAL** ... [A direct comparison against one of the related methods in the discussion section would help]**REQ** ...

Figure 4.1: Sample ICLR review excerpts from openreview.net. Propositions are annotated with types, such as **FACT** (fact), **EVAL** (evaluation), and **REQ** (request). Both assigned a rating of 5 (out of 10) for the reviewed paper. Review #2 contains in-depth evaluation and actionable suggestion, thus is perceived to be of a higher quality.

terpretation and evaluation of the research. Constructive reviews, e.g., review #2, often contain in-depth analysis as well as concrete suggestions. As a result, automatically identifying propositions and their types would be useful to understand the composition of peer reviews.

Therefore, we propose *an argument mining-based approach to understand the content and structure of peer reviews*. Argument mining studies the automatic detection of argumentative components and structure within discourse (Peldszus and Stede, 2013). Specifically, argument types (e.g. evidence and reasoning) and

their arrangement are indicative of argument quality (Habernal and Gurevych, 2016; Wachsmuth et al., 2017a).

In this work, we focus on two specific tasks: (1) **proposition segmentation**—detecting elementary argumentative discourse units that are propositions, and (2) **proposition classification**—labeling the propositions according to their types (e.g., evaluation vs. request).

Since there was no annotated dataset for peer reviews, as part of this study, we first collect 14.2K reviews from major machine learning (ML) and natural language processing (NLP) venues. We create a dataset, **AMPERE** (Argument Mining for PEer REviews), by annotating 400 reviews with 10,386 propositions and labeling each proposition with the type of EVALUATION, REQUEST, FACT, REFERENCE, QUOTE, or NON-ARG. Significant inter-annotator agreement is achieved for proposition segmentation (Cohen’s $\kappa = 0.93$), with good consensus level for type annotation (Krippendorff’s $\alpha_U = 0.61$). We release the dataset and annotation guideline at <http://xinyuhua.github.io/Resources/naacl19/>.

We benchmark our new dataset with state-of-the-art and popular argument mining models to better understand the challenges posed in this new domain. We observe a significant drop of performance for proposition segmentation on AMPERE, mainly due to its different argument structure. For instance, 25% of the sentences contain more than one proposition, compared to that of 8% for essays (Stab and Gurevych, 2017), motivating new solutions for segmentation and classification.

We further investigate review structure difference across venues based on

proposition usage, and uncover several patterns. For instance, ACL reviews tend to contain more propositions than those in ML venues, especially with more requests but fewer facts. We further find that reviews with extreme ratings, i.e., strong reject or accept, tend to be shorter and make much fewer requests. Moreover, we probe the salient words for different proposition types. For example, ACL reviewers ask for more “examples” when making requests, while ICLR reviews contain more evaluation of “network” and how models are “trained”.

4.2 AMPERE Dataset

We collect review data from three sources: (1) openreview.net—an online peer reviewing platform for ICLR 2017, ICLR 2018, and UAI 2018¹; (2) reviews released for accepted papers at NeurIPS from 2013 to 2017; and (3) opted-in reviews for ACL 2017 from Kang et al. (2018). Table 4.1 shows review count for each venue. All venues except NeurIPS have paper rating scores attached to the reviews.

ICLR	UAI	ACL	NeurIPS	Total
4,057	768	275	9,152	14,202

Table 4.1: Statistics for review data sets used in our study.

Annotation Process. For proposition segmentation, we adopt the concepts from Park et al. (2015) and instruct the annotators to identify elementary argumentative discourse units on sentence or sub-sentence level, based on their

¹ICLR reviews are downloaded using the public API: <https://github.com/iesl/openreview-py>. UAI reviews are collected directly by the OpenReview team.

discourse functions and topics. They then classify the propositions into five types with an additional non-argument category, as explained below:

- **EVALUATION:** Subjective statements, often containing qualitative judgment. Ex: *“This paper shows nice results on a number of small tasks.”*
- **REQUEST:** Statements suggesting a course of action. Ex: *“The authors should compare with the following methods.”*
- **FACT:** Objective information of the paper or commonsense knowledge. Ex: *“Existing works on multi-task neural networks typically use hand-tuned weights...”*
- **REFERENCE:** Citations and URLs. Ex: *“see MuseGAN (Dong et al), MidiNet (Yang et al), etc ”*
- **QUOTE:** Quotations from the paper. Ex: *“The author wrote ‘where r is lower bound of feature norm’.”*
- **NON-ARG:** Non-argumentative statements. Ex: *“Aha, now I understand.”*

We sample 400 ICLR 2018 reviews for annotation. To ensure the subset is representative of the full dataset, samples are drawn based on two aspects: review length and rating score. Table 4.2 shows the distribution of reviews with regard to their length in the full ICLR 2018 dataset and the subset we sampled for annotation (AMPERE). As can be seen, the distribution over five bins are consistent between AMPERE and the full dataset. A similar trend is observed on rating distribution in Table 4.3.

Two annotators who are fluent English speakers first label the 400 reviews with proposition segments and types, and a third annotator then resolves disagreements.

Length	(0,200]	(200,400]	(400,600]	(600,800]	(800, ∞)
AMPERE	14.8%	35.5%	25.3%	10.0%	14.6%
ICLR2018	17.6%	39.3%	23.8%	11.4%	7.9%

Table 4.2: Review length distribution of the full ICLR 2018 dataset and AMPERE, which consists of 400 sampled reviews.

Rating	1	2	3	4	5
AMPERE	3.0%	32.5%	43.8%	19.3%	1.5%
ICLR2018	2.6%	32.5%	42.4%	20.6%	1.8%

Table 4.3: Review rating distribution of AMPERE and the full ICLR 2018 dataset.

Inter-Annotator Agreement (IAA). We calculate the inter-annotator agreement between the two annotators. A Cohen’s κ of 0.93 is achieved for proposition segmentation, with each review treated as a BIO sequence. For classification, unitized Krippendorff’s α_U (Krippendorff, 2004), which considers disagreements among segmentation, is calculated per review and then averaged over all samples, and the value is 0.61. Among the exactly matched proposition segments, we report a Cohen’s κ of 0.64. The agreement scores for each type are listed in Table 4.4.

	EVAL	REQ	FACT	REF	QUOT	NON-A	overall
α_U	0.51	0.64	0.60	0.63	0.41	0.18	0.61
κ	0.60	0.68	0.64	0.88	0.59	0.27	0.64

Table 4.4: Inter-annotator agreement for all categories.

Statistics. Table 4.5 shows comparison between AMPERE and some other argument mining datasets of different genres. We also show the number of propositions in each category in Table 4.6. The most frequent types are evaluation (38.3%) and fact (36.5%).

Dataset	# Doc	# Sent	# Prop
Comments (Park and Cardie, 2018)	731	3,994	4,931
Essays (Stab and Gurevych, 2017)	402	7,116	6,089
News (Al-Khatib et al., 2016)	300	11,754	14,313
Web (Habernal and Gurevych, 2017)	340	3,899	1,882
AMPERE	400	8,030	10,386

Table 4.5: Statistics for AMPERE and some other argument mining corpora, including # of annotated propositions.

EVAL	REQ	FACT	REF	QUOT	NON-A	Total
3,982	1,911	3,786	207	161	339	10,386

Table 4.6: Number of propositions per type in AMPERE.

4.3 Experiments with Existing Models

We benchmark AMPERE with popular and state-of-the-art models for proposition segmentation and classification. Both tasks can be treated as sequence tagging problems with the setup similar to Schulz et al. (2018). For experiments, 320 reviews (7,999 propositions) are used for training and 80 reviews (2,387 propositions) are used for testing. Following Niculae et al. (2017), 5-fold cross validation on the training set is used for hyperparameter tuning. For pre-processing, we utilize the Stanford CoreNLP toolkit (Manning et al., 2014). To improve the accuracy of tokenization, we manually replace mathematical formulas, variables, URL links, and formatted citation with special tokens such as <EQN>, <VAR>, <URL>, and <CIT>.

	Precision	Recall	F1
FullSent	73.68	56.00	63.64
PDTB-conn	51.11	49.71	50.40
RST-parser	30.28	43.00	35.54
CRF	66.53	52.92	58.95
BiLSTM-CRF	82.25	79.96	81.09*
CRF-joint	74.99	63.33	68.67
BiLSTM-CRF-joint	81.12	78.42	79.75

Table 4.7: Proposition segmentation results. Result that is significantly better than all comparisons is marked with * ($p < 10^{-6}$, McNemar test).

4.3.1 Task I: Proposition Segmentation

We consider three baselines. **FullSent**: treating each sentence as a proposition. **PDTB-conn**: further segmenting sentences when any discourse connective (collected from Penn Discourse Treebank (Prasad et al., 2008)) is observed. **RST-parser**: segmenting discourse units by the RST parser in Feng and Hirst (2011).

For learning-based methods, we start with Conditional Random Field (**CRF**) (Lafferty et al., 2001; Okazaki, 2007) with features proposed by Stab and Gurevych (2017) (2017, Table 7), and **BiLSTM-CRF**, a bidirectional Long Short-Term Memory network (BiLSTM) connected to a CRF output layer and further enhanced with ELMo representation (Peters et al., 2018). We adopt the BIO scheme for sequential tagging (Ramshaw and Marcus, 1999), with O corresponding to NON-ARG. Finally, we consider **jointly modeling** segmentation and classification by appending the proposition types to BI tags, e.g., B-fact, with CRF (**CRF-joint**) and BiLSTM-CRF (**BiLSTM-CRF-joint**).

Table 4.7 shows that BiLSTM-CRF outperforms other methods in F1. More importantly, the performance on reviews is lower than those reached on existing datasets, e.g., an F1 of 86.7 is obtained by CRF for essays (Stab and Gurevych,

2017). This is mostly due to essays' better structure, with frequent use of discourse connectives.

4.3.2 Task II: Proposition Classification

With given proposition segments, predicted or gold-standard, we experiment with proposition-level models to label proposition types. We utilize two baselines. **Majority** simply assigns the majority type in the training set. **PropLexicon** matches the following lexicons for different proposition types in order, and returns the first corresponding type with a match; if no lexicon is matched, the proposition is labeled as NON-ARG:

- REFERENCE: <URL>, <CIT>
- QUOTE: “, ”, ’
- REQUEST: *should, would be nice, why, please, would like to, need*
- EVALUATION: *highly, very, unclear, clear, interesting, novel, well, important, similar, clearly, quite, good*
- FACT: *author, authors, propose, present, method, parameters, example, dataset, same, incorrect, correct*

For supervised models, we employ linear **SVM** with a squared hinge loss and group Lasso regularizer (Yuan and Lin, 2006). It is trained with the top 500 features selected from Table 9 in (Stab and Gurevych, 2017) by χ^2 test. We also train a convolutional neural network (CNN) proposed by Kim (2014), with the same setup and pre-trained word embeddings from word2vec (Mikolov et al.,

	Overall	EVAL	REQ	FACT	REF	QUOT
<i>With Gold-Standard Segments</i>						
Majority	40.75	57.90	–	–	–	–
PropLexicon	36.83	40.42	36.07	32.23	59.57	31.28
SVM	60.98	63.88	69.02	54.74	69.47	7.69
CNN	66.56*	69.02	63.26	66.17	67.44	52.94
<i>With Predicted Segments</i>						
Majority	33.30	47.60	–	–	–	–
PropLexicon	23.21	22.45	23.97	23.73	35.96	16.67
SVM	51.46	54.05	48.16	52.77	52.27	4.71
CNN	55.48	57.75	53.71	55.19	48.78	33.33
CRF-joint	50.69	46.78	55.74	52.27	55.77	26.47
BiLSTM-CRF-joint	62.64*	62.36*	67.31*	61.86	54.74	37.36

Table 4.8: Proposition classification F1 scores. Results that are significant better than other methods are marked with * ($p < 10^{-6}$, McNemar test).

2013). Finally, results by joint models of CRF and BiLSTM-CRF are also reported.

F1 scores for all propositions and each type are reported in Table 4.8. A prediction is correct when both segment and type are matched with the true labels. CNN performs better for types with significantly more training samples, i.e., evaluation and fact, indicating the effect of data size on neural model’s performance. Joint models (CRF-joint and BiLSTM-CRF-joint) yield the best F1 scores for all categories when gold-standard segmentation is unavailable.

4.3.3 Training Details

For all models except CNN, we conduct 5-fold cross validation on training set to select hyperparameters.

CRF. We utilize the CRFSuite (Okazaki, 2007) implementation and tune coeffi-

coefficients C_1 and C_2 for ℓ_1 and ℓ_2 regularizer. For segmentation task the optimal setup is $C_1 = 0.0$ and $C_2 = 1.0$; for joint prediction, $C_1 = 1.0$ and $C_2 = 0.01$ is used.

BiLSTM-CRF. We experiment with implementation by [Reimers and Gurevych \(2017\)](#) with an extra ELMo embedding. Based on the cross validation for both segmentation and joint learning, the optimal network architecture selected has two layers with 100 dimensional hidden states each, with dropout probabilities of 0.5 for both layers. The word embedding pre-trained by [Komninos and Mandhar \(2016\)](#) is chosen, as it outperforms GloVe embeddings ([Pennington et al., 2014](#)) trained either on Google News or Wikipedia.

SVM. We utilize SAGA ([Defazio et al., 2014](#)) implemented in the Lightning library ([Blondel and Pedregosa, 2016](#)) to learn a linear SVM optimized with Coordinate Descent ([Wright, 2015](#)). The coefficient for a group Lasso regularizer ([Yuan and Lin, 2006](#)) is set to 0.001 by cross validation.

CNN. We implement the CNN-non-static variant as described in [Kim \(2014\)](#), with the following configuration: filter window sizes of {3,4,5}, with 128 feature maps each. Dropout probability is 0.5. 300 dimensional word embeddings are initiated with the pre-trained `word2vec` on 100 billion Google News ([Mikolov et al., 2013](#)).

4.4 Proposition Analysis by Venues

Here we leverage the BiLSTM-CRF-joint model trained on the annotated AMPERE data to identify propositions and their types in unlabeled reviews from

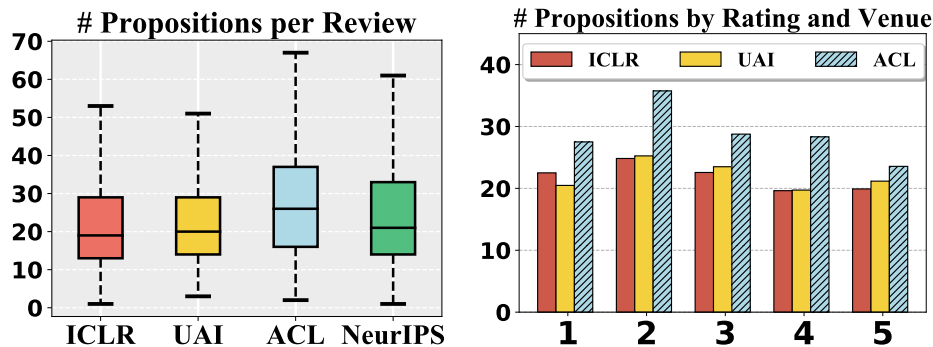


Figure 4.2: Proposition number in reviews. Differences among venues are all significant except UAI vs. ICLR and ACL vs. NeurIPS ($p < 10^{-6}$, unpaired t -test).

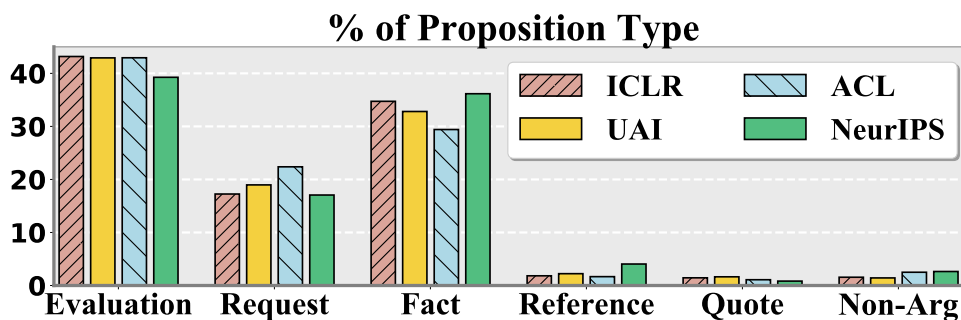


Figure 4.3: Distribution of proposition type per venue.

the four venues (ICLR, UAI, ACL, and NeurIPS), to understand the content and structure of peer reviews at a larger scale.

Proposition Usage by Venue and Rating. Figure 4.2 shows the average number of propositions per review, grouped by venue and rating. Scores in 1 – 10 are scaled to 1 – 5 by $\lceil x/2 \rceil$, with 1 as strong reject and 5 as strong accept. ACL and NeurIPS have significantly more propositions than ICLR and UAI. Ratings, which reflect a reviewer’s judgment of paper quality, also affect proposition usage. We find that reviews with extreme ratings, i.e., 1 and 5, tend to have fewer propositions.

We further study the distribution of proposition type in each venue. As ob-

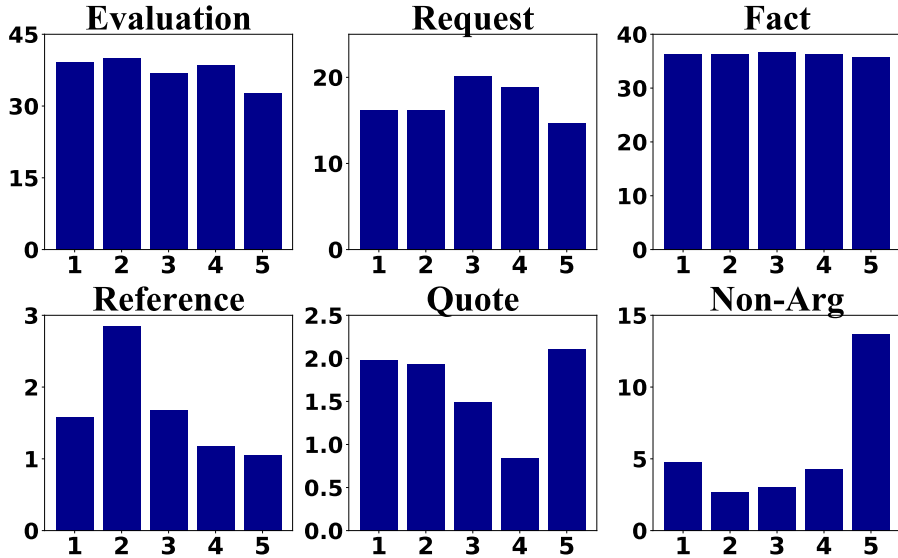


Figure 4.4: Distribution of proposition type per rating (in %) on AMPERE.

served in Figure 4.3, ACL reviews contain more requests but fewer facts than other venues. Specifically, we find that 94.6% of ACL reviews have at least one REQUEST proposition, compared to 81.5% for ICLR and 84.7% for UAI. We also show proposition type distribution based on ratings in Figure 4.4. Reviews with the highest rating tend to use fewer evaluation and reference, while reviews with ratings of 3 – 4 (borderline or weak accept) contain more requests. We further observe a sharp decrease of QUOTE usage in rating group 4, and a surge of NON-ARG for rating group 5, while FACT remains consistent across rating ranges.

	EVAL	REQ	FACT	REF	QUOT	NON-A
EVAL	50.3	17.2	27.3	1.0	1.4	2.9
REQ	32.2	41.6	19.4	1.8	2.3	2.8
FACT	33.5	11.0	51.2	1.3	0.9	2.0
REF	15.0	10.8	18.0	50.9	3.6	1.8
QUOT	31.2	23.6	25.5	1.3	12.1	6.4
NON-A	31.9	15.5	22.7	1.3	2.8	25.9

Figure 4.5: Proposition transition probabilities on AMPERE.

Proposition Structure. Argumentative structure, which is usually studied as

EVALUATION	REQUEST	FACT	REFERENCE	QUOTE
All Venues				
overall, unclear, not, please, contribution, interesting	could, if, think, seem, should, more, suggest	each, some, <URL>, et, al., conference, paper, proceedings, arxiv	paper, we, ; our	
ICLR				
network, general, ac-network, trained	appendix, training, work, then, deep, ;, speech	nips, pp., not, section, 4, 5,	agent	
UAI				
quality, found, rel- <VAR>, evant, major	model, stochastic, nice, col- considers, umn	called, artificial, discovery, -, second, column, sense, etc., via, systems	processes, connections	
ACL				
weaknesses, so, main	word, consider, further, proposed	method, language, extraction, proposed, emnlp, computational, linguistics		
NeurIPS				
theoretical, interest, nips	<EQN>, following, practical, address, quality	clarity, <EQN>, maximum, for, see, class, de- of, in, which, <EQN>, comments, tailed, guidelines	reviewer	

Table 4.9: Salient words ($\alpha = 0.001, \chi^2$ test) per proposition type. Top 5 frequent words that are unique for each venue are shown. “<EQN>”, “<URL>”, and “<VAR>” are equations, URL links, and variables.

support and attack relations, reveals how propositions are organized into coherent text. According to [Park and Cardie \(2018\)](#), 75% of support relations happen between adjacent propositions in user comments. We thus plot the proposition transition probability matrix in [Figure 4.5](#), to show the argument structure in AMPERE. The high probabilities along the diagonal line imply that propositions of the same type are often constructed consecutively, with the exception of quote, which is more likely to be followed by evaluation.

Proposition Type and Content. We also probe the salient words used for each proposition type, and the difference of their usage across venues. For each venue, we utilize log-likelihood ratio test ([Lin and Hovy, 2000a](#)) to identify the representative words in each proposition type compared to other types. [Table 4.9](#) shows both the commonly used salient words across venues and the unique words with top frequencies for each venue ($\alpha = 0.001$, χ^2 test). For evaluation, all venues tend to focus on clarity and contribution, with ICLR discussing more about “network” and NeurIPS often mentioning equations. ACL reviews then frequently request for “examples”.

4.5 Conclusion

We study the content and structure of peer reviews under the argument mining framework. AMPERE, a new dataset of peer reviews, is collected and annotated with propositions and their types. We benchmark AMPERE with state-of-the-art argument mining models for proposition segmentation and classification. We leverage the classifiers to analyze the proposition usage in reviews across ML and NLP venues, showing interesting patterns in proposition types and

content.

CHAPTER 5
ARGUMENT GENERATION AUGMENTED
WITH EXTERNALLY RETRIEVED EVIDENCE

Our goal in the previous two chapters is to understand the argument structure and their interplay with external supports. This chapter describes an argument generation system, equipped with a retrieval component to access the useful external information from Wikipedia. We also introduce the ChangeMyView dataset, collected as part of this work to facilitate argument generation research.

This work is published at ACL 2018 (Hua and Wang, 2018). Relevant resources can be found at: <https://xinyuhua.github.io/neural-argument-generation/>.

5.1 Introduction

Generating high quality arguments plays a crucial role in decision-making and reasoning processes (Bonet and Geffner, 1996; Byrnes, 2013). A multitude of arguments and counter-arguments are constructed on a daily basis, both online and offline, to persuade and inform us on a wide range of issues. For instance, debates are often conducted in legislative bodies to secure enough votes for bills to pass. In another example, online deliberation has become a popular way of soliciting public opinions on new policies' pros and cons (Albrecht, 2006; Park et al., 2012). Nonetheless, constructing persuasive arguments is a daunting task, for both human and computers. We believe that developing effective argument generation models will enable a broad range of compelling applications, includ-

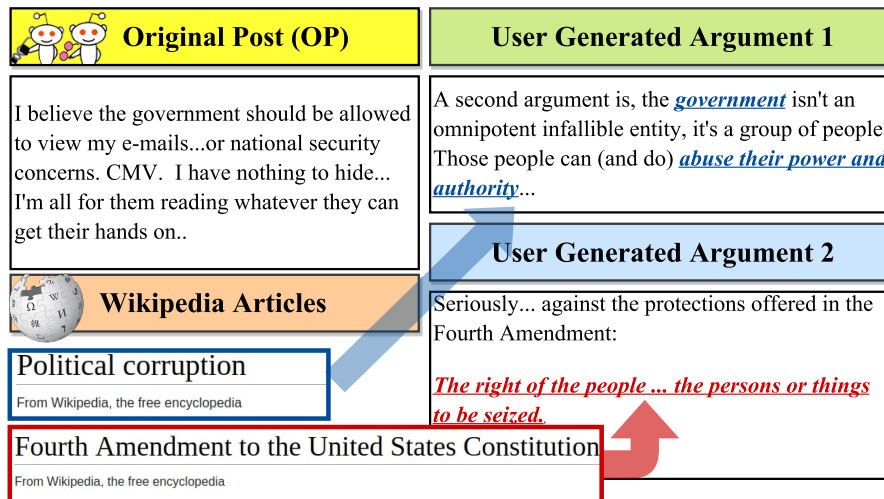


Figure 5.1: Sample user arguments from Reddit Change My View community that argue against original post’s thesis on “government should be allowed to view private emails”. Both arguments leverage supporting information from Wikipedia articles.

ing debate coaching, improving students’ essay writing skills, and providing context of controversial issues from different perspectives. As a consequence, there exists a pressing need for automating the argument construction process.

To date, progress made in argument generation has been limited to retrieval-based methods—arguments are ranked based on relevance to a given topic, then the top ones are selected for inclusion in the output (Rinott et al., 2015; Wachsmuth et al., 2017b; Hua and Wang, 2017). Although sentence ordering algorithms are developed for information structuring (Sato et al., 2015; Reisert et al., 2015), existing methods lack the ability of synthesizing information from different resources, leading to redundancy and incoherence in the output.

In general, the task of argument generation presents numerous challenges, ranging from aggregating supporting evidence to generating text with coherent logical structure. One particular hurdle comes from the underlying natural language generation (NLG) stack, whose success has been limited to a small set of

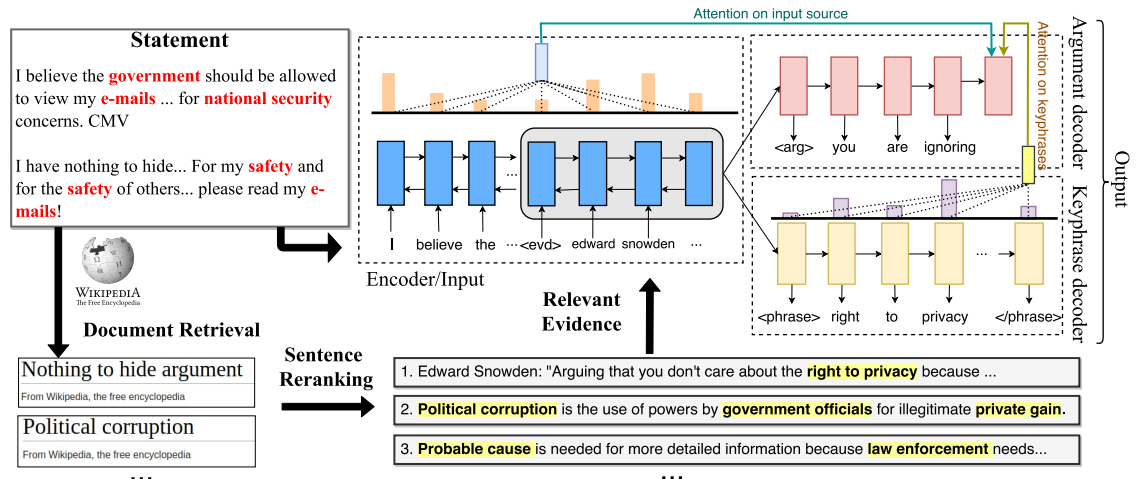


Figure 5.2: Overview of our system pipeline (best viewed in color). Given a statement, relevant articles are retrieved from Wikipedia with topic signatures from statement as queries (marked in red and bold). A reranking module then outputs top sentences as evidence. The statement and the evidence (encoder in gray) are concatenated and encoded as input for our argument generation model. During decoding, the keyphrase decoder first generates talking points as phrases, followed by the argument decoder which constructs the argument by attending both input and keyphrases.

domains. Especially, most previous NLG systems rely on templates that are either constructed by rules (Hovy, 1993; Belz, 2008; Bouayad-Agha et al., 2011), or acquired from a domain-specific corpus (Angeli et al., 2010) to enhance grammaticality and coherence. This makes them unwieldy to be adapted for new domains.

In this work, we study the following novel problem: *given a statement on a controversial issue, generate an argument of an alternative stance*. To address the above challenges, we present a *neural network-based argument generation framework augmented with externally retrieved evidence*. Our model is inspired by the observation that when humans construct arguments, they often collect references from external sources, e.g., Wikipedia or research papers, and then write their own arguments by synthesizing talking points from the references.

Figure 5.1 displays sample arguments by users from Reddit subcommunity /r/ChangeMyView¹ who argue against the motion that “government should be allowed to view private emails”. Both replies leverage information drawn from Wikipedia, such as “political corruption” and “Fourth Amendment on protections of personal privacy”.

Concretely, our neural argument generation model adopts the popular encoder-decoder-based sequence-to-sequence (seq2seq) framework (Sutskever et al., 2014), which has achieved significant success in various text generation tasks (Bahdanau et al., 2015; Wen et al., 2015; Wang and Ling, 2016; Mei et al., 2016; Wiseman et al., 2017). Our encoder takes as input a statement on a disputed issue, and a set of relevant evidence automatically retrieved from English Wikipedia². Our decoder consists of two separate parts, one of which first generates keyphrases as intermediate representation of “talking points”, and the other then generates an argument based on both input and keyphrases.

Automatic evaluation based on BLEU (Papineni et al., 2002) shows that our framework generates better arguments than directly using retrieved sentences or popular seq2seq-based generation models (Bahdanau et al., 2015) that are also trained with retrieved evidence. We further design a novel evaluation procedure to measure whether the arguments are on-topic by predicting their relevance to the given statement based on a separately trained relevance estimation model. Results suggest that our model generated arguments are more likely to be predicted as on-topic, compared to other seq2seq-based generations models.

¹<https://www.reddit.com/r/changemyview>

²<https://en.wikipedia.org/>

5.2 Framework

Our argument generation pipeline, consisting of *evidence retrieval* and *argument construction*, is depicted in Figure 5.2. Given a statement, a set of queries are constructed based on its topic signature words (e.g., “government” and “national security”) to retrieve a list of relevant articles from Wikipedia. A reranking component further extracts sentences that may contain supporting evidence, which are used as additional input information for the neural argument generation model.

The generation model then encodes the statement and the evidence with a shared encoder in sequence. Two decoders are designed: the *keyphrase decoder* first generates an intermediate representation of talking points in the form of keyphrases (e.g., “right to privacy”, “political corruption”), followed by a separate *argument decoder* which produces the final argument.

5.3 Data Collection and Processing

We draw data from Reddit subcommunity `/r/ChangeMyView` (henceforth CMV), which focuses on facilitating open discussions on a wide range of disputed issues. Specifically, CMV is structured as discussion threads, where the original post (OP) starts with a viewpoint on a controversial topic, followed with detailed reasons, then other users reply with counter-arguments. Importantly, when a user believes his view has been changed by an argument, a *delta* is often awarded to the reply.

In total, 26,761 threads from CMV are downloaded, dating from January

2013 to June 2017. Only root replies (i.e., replies directly addressing OP) that meet all of the following requirements are included: (1) longer than 5 words, (2) without offensive language³, (3) awarded with *delta* or with more upvotes than downvotes, and (4) not generated by system moderators. We publicly release the processed dataset at: <http://xinyuhua.github.io/Resources/>.

After filtering, the resultant dataset contains 26,525 OPs along with 305,475 relatively high quality root replies. We treat each OP as the input statement, and the corresponding root replies as target arguments, on which our model is trained and evaluated.

A Focused Domain Dataset. The current dataset contains diverse domains with unbalanced numbers of arguments. We therefore choose samples from the politics domain due to its large volume of discussions and good coverage of popular arguments in the domain.

However, topic labels are not available for the discussions. We thus construct a domain classifier for politics vs. non-politics posts based on a logistic regression model with unigram features, trained from our heuristically labeled Wikipedia abstracts⁴. Concretely, we manually collect two lists of keywords that are indicative of politics and non-politics (Table 5.1). Each abstract is labeled as politics or non-politics if its title only matches keywords from one category. In total, 264,670 politics abstracts and 827,437 of non-politics are labeled. Starting from this dataset, our domain classifier is trained in a bootstrapping manner by gradually adding OPs predicted as politics or non-politics. Finally, 12,549 OPs are labeled as politics, each of which is paired with 9.4 high-quality target argu-

³We use offensive words collected by Google’s What Do You Love project: <https://gist.github.com/jamiew/1112488>, last accessed on February 22nd, 2018.

⁴About 1.3 million English Wikipedia abstracts are downloaded from <http://dbpedia.org/page/>.

Politics		non-Politics	
politics	political	science	media
policy	congress	automobiles	sports
rights	election	football	fashion
president	trump	entertainment	movie
clinton	immigration	movies	music
democracy	democrats	musics	art
democratic	republican	arts	television
constitution	liberal	religion	philosophy
government	legalization	morality	dating
surveillance	amnesty	eugenics	marriage
antisemitism	terrorism	parenthood	history
war	taxation	organic	handicaps
liberalism	libertarianism	disease	
marxism	conservatism		
anarchism	autocracy		
fascism	voting		

Table 5.1: Lists of politics and non-politics keywords used to obtain Wikipedia abstract labels.

ments on average. The average length for OPs is 16.1 sentences of 356.4 words, and 7.7 sentences of 161.1 words for arguments.

5.4 Model

In this section, we present our argument generation model, which jointly learns to generate talking points in the form of keyphrases and produce arguments based on the input and keyphrases. Extended from the successful seq2seq attentional model (Bahdanau et al., 2015), our proposed model is novel in the following ways. First, two separate decoders are designed, one for generating keyphrases, the other for argument construction. By sharing the encoder with keyphrase generation, our argument decoder is better aware of salient

talking points in the input. Second, a novel attention mechanism is designed for argument decoding by attending both input and the previously generated keyphrases. Finally, a reranking-based beam search decoder is introduced to promote topic-relevant generations.

5.4.1 Model Formulation

Our model takes as input a sequence of tokens $\mathbf{x} = \{\mathbf{x}^O; \mathbf{x}^E\}$, where \mathbf{x}^O is the statement sequence and \mathbf{x}^E contains relevant evidence that is extracted from Wikipedia based on a separate retrieval module. A special token `<evd>` is inserted between \mathbf{x}^O and \mathbf{x}^E . Our model then first generates a set of keyphrases as a sequence $\mathbf{y}^p = \{y_i^p\}$, followed by an argument $\mathbf{y}^a = \{y_i^a\}$, by maximizing $\log P(\mathbf{y}|\mathbf{x})$, where $\mathbf{y} = \{\mathbf{y}^p; \mathbf{y}^a\}$.

The objective is further decomposed into $\sum_t \log P(y_t|y_{1:t-1}, \mathbf{x})$, with each term estimated by a softmax function over a non-linear transformation of decoder hidden states s_t^a and s_t^p , for argument decoder and keyphrase decoder, respectively. The hidden states are computed as done in Bahdanau et al. (2015) with attention:

$$\mathbf{s}_t = g(\mathbf{s}_{t-1}, \mathbf{c}_t, y_t) \quad (5.1)$$

$$\mathbf{c}_t = \sum_{j=1}^T \alpha_{tj} \mathbf{h}_j \quad (5.2)$$

$$\alpha_{tj} = \frac{\exp(e_{tj})}{\sum_{k=1}^T \exp(e_{tk})} \quad (5.3)$$

$$e_{tj} = \mathbf{v}^T \tanh(\mathbf{W}_h \mathbf{h}_j + \mathbf{W}_s \mathbf{s}_t + \mathbf{b}_{attn}) \quad (5.4)$$

Notice that two sets of parameters and different state update functions $g(\cdot)$

are learned for separate decoders: $\{\mathbf{W}_h^a, \mathbf{W}_s^a, \mathbf{b}_{attn}^a, g^a(\cdot)\}$ for the argument decoder; $\{\mathbf{W}_h^p, \mathbf{W}_s^p, \mathbf{b}_{attn}^p, g^p(\cdot)\}$ for the keyphrase decoder.

Encoder. A two-layer bidirectional LSTM (bi-LSTM) is used to obtain the encoder hidden states \mathbf{h}_i for each time step i . For biLSTM, the hidden state is the concatenation of forward and backward hidden states: $\mathbf{h}_i = [\vec{\mathbf{h}}_i; \overleftarrow{\mathbf{h}}_i]$. Word representations are initialized with 200-dimensional pre-trained GloVe embeddings (Pennington et al., 2014), and updated during training. The last hidden state of encoder is used to initialize both decoders. In our model the encoder is shared by argument and keyphrase decoders.

Decoders. Our model is equipped with two decoders: *keyphrase decoder* and *argument decoder*, each is implemented with a separate two-layer unidirectional LSTM, in a similar spirit with one-to-many multi-task sequence-to-sequence learning (Luong et al., 2015a). The distinction is that our training objective is the sum of two loss functions:

$$\mathcal{L}(\theta) = -\frac{\alpha}{T_p} \sum_{(x, y^p) \in D} \log P(y^p | x; \theta) - \frac{(1 - \alpha)}{T_a} \sum_{(x, y^a) \in D} \log P(y^a | x; \theta) \quad (5.5)$$

where T_p and T_a denote the lengths of reference keyphrase sequence and argument sequence. α is a weighting coefficient, and it is set as 0.5 in our experiments.

Attention over Both Input and Keyphrases. Intuitively, the argument decoder should consider the generated keyphrases as talking points during the generation process. We therefore propose an attention mechanism that can attend both encoder hidden states and the keyphrase decoder hidden states. Additional context vector \mathbf{c}'_i is then computed over keyphrase decoder hidden states

s_j^p , which is used for computing the new argument decoder state:

$$s_t^a = g'(s_{t-1}^a, [c_t; c'_t], y_t^a) \quad (5.6)$$

$$c'_t = \sum_{j=1}^{T_p} \alpha'_{tj} s_j^p \quad (5.7)$$

$$\alpha'_{tj} = \frac{\exp(e'_{tj})}{\sum_{k=1}^{T_p} \exp(e'_{tk})} \quad (5.8)$$

$$e'_{tj} = \mathbf{v}'^T \tanh(\mathbf{W}'_p s_j^p + \mathbf{W}'_a s_t^a + \mathbf{b}'_{attn}) \quad (5.9)$$

where s_j^p is the hidden state of keyphrase decoder at position j , s_t^a is the hidden state of argument decoder at timestep t , and c_t is computed in Eq. 5.2.

Decoder Sharing. We also experiment with a shared decoder between keyphrase generation and argument generation: the last hidden state of the keyphrase decoder is used as the initial hidden state for the argument decoder. A special token `<arg>` is inserted between the two sequences, indicating the start of argument generation.

5.4.2 Hybrid Beam Search Decoding

Here we describe our decoding strategy on the argument decoder. We design a hybrid beam expansion method combined with segment-based reranking to promote diversity of beams and informativeness of the generated arguments.

Hybrid Beam Expansion. In the standard beam search, the top k words of highest probability are selected deterministically based on the softmax output to expand each hypothesis. However, this may lead to suboptimal output for text generation (Wiseman and Rush, 2016), e.g., one beam often dominates and thus inhibits hypothesis diversity. Here we only pick the top n words ($n < k$), and

randomly draw another $k - n$ words based on the multinomial distribution after removing the n expanded words from the candidates. This leads to a more diverse set of hypotheses.

Segment-based Reranking. We also propose to rerank the beams every p steps based on beam’s coverage of content words from input. Based on our observation that likelihood-based reranking often leads to overly generic arguments (e.g., “I don’t agree with you”), this operation has the potential of encouraging more informative generation. $k = 10$, $n = 3$, and $p = 10$ are used for experiments. The effect of parameter selection is studied in Section 5.7.

5.5 Relevant Evidence Retrieval

5.5.1 Retrieval Methodology

We take a two-step approach for retrieving *evidence sentences*: given a statement, (1) constructing one query per sentence and retrieving relevant articles from Wikipedia, and (2) reranking paragraphs and then sentences to create the final set of evidence sentences. Wikipedia is used as our evidence source mainly due to its objective perspective and broad coverage of topics. A dump of December 21, 2016 was downloaded. For training, evidence sentences are retrieved with queries constructed from target user arguments. For test, queries are constructed from OP. Key statistics are displayed in Table 5.2.

Article Retrieval. We first create an inverted index lookup table for Wikipedia as done in Chen et al. (2017). For a given statement, we construct one query

	<i>Queries Constructed from</i>	
	OP	Argument
Average # Topic Signature	17.2	9.8
Average # Query	6.7	1.9
Average # Article Retrieved	26.1	8.0
Average # Sentence Retrieved	67.3	8.5

Table 5.2: Statistics for evidence sentence retrieval from Wikipedia. Considering query construction from either OP or target user arguments, we show the average numbers of topic signatures collected, queries constructed, and retrieved articles and sentences.

per sentence to broaden the diversity of retrieved articles. Therefore, multiple passes of retrieval will be conducted if more than one query is created. Specifically, we first collect topic signature words of the post. Topic signatures (Lin and Hovy, 2000b) are terms strongly correlated with a given post, measured by log-likelihood ratio against a background corpus. We treat posts from other discussions in our dataset as background. For each sentence, one query is constructed based on the noun phrases and verbs containing at least one topic signature word. For instance, a query “the government, my e-mails, national security” is constructed for the first sentence of OP in the motivating example (Figure 5.2). Top five retrieved articles with highest TF-IDF similarity scores are kept per query.

Sentence Reranking. The retrieved articles are first segmented into paragraphs, which are reranked by TF-IDF similarity to the given statement. Up to 100 top ranked paragraphs with positive scores are retained. These paragraphs are further segmented into sentences, and reranked according to TF-IDF similarity again. We only keep up to 10 top sentences with positive scores for inclusion in the evidence set.

Component	Stage 1	Stage 2	Stage 3
<i>Encoder</i>			
OP	50	150	400
Evidence	0	80	120
<i>Decoder</i>			
Keyphrases	0	80	120
Target Argument	30	80	120

Table 5.3: Truncation size (i.e., number of tokens including delimiters) for different stages during training. Note that in the first stage we do not include evidence and keyphrases.

5.5.2 Gold-Standard Keyphrase Construction

To create training data for the keyphrase decoder, we use the following rules to identify keyphrases from evidence sentences that are reused by human writers for argument construction:

- Extract noun phrases and verb phrases from evidence sentences using Stanford CoreNLP (Manning et al., 2014).
- Keep phrases of length between 2 and 10 that overlap with content words in the argument.
- If there is span overlap between phrases, the longer one is kept if it has more content word coverage of the argument; otherwise the shorter one is retained.

The resultant phrases are then concatenated with a special delimiter `<phrase>` and used as gold-standard generation for training.

5.6 Experimental Setup

5.6.1 Final Dataset Statistics

Encoding the full set of evidence by our current decoder takes a huge amount of time. We there propose a sampling strategy to allow the encoder to finish encoding within reasonable time by considering only a subset of the evidence: For each sentence in the statement, up to three evidence sentences are randomly sampled from the retrieved set; then the sampled sentences are concatenated. This procedure is repeated three times per statement, where a statement is an user argument for training data and an OP for test set. In our experiments, we remove duplicates samples and the ones without any retrieved evidence sentence. Finally, we break down the augmented data into a training set of 224,553 examples (9,737 unique OPs), 13,911 for validation (640 OPs), and 30,417 retained for test (1,892 OPs).

5.6.2 Training Setup

For all models, we use a two-layer biLSTM as encoder and a two-layer unidirectional LSTM as decoder, with 200-dimensional hidden states in each layer. We apply dropout (Gal and Ghahramani, 2016) on RNN cells with a keep probability of 0.8. We use Adam (Kingma and Ba, 2015) with an initial learning rate of 0.001 to optimize the cross-entropy loss. Gradient clipping is also applied with the maximum norm of 2.0. The input and output vocabulary sizes are both 50k.

Curriculum Training. We train the models in three stages where the truncated

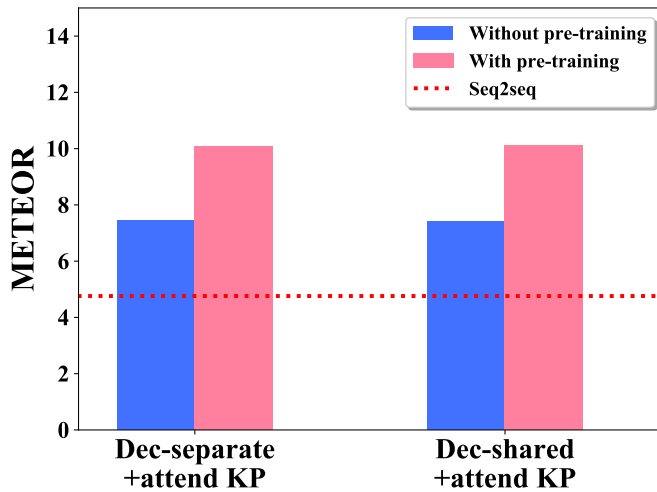


Figure 5.3: Effect of pre-training on seq2seq-based models. Red dotted line shows the performance of the seq2seq model used for parameter initialization.

input and output lengths are gradually increased. Details are listed in Table 5.3. Importantly, this strategy allows model training to make rapid progress during early stages. Training each of our full models takes about four days on a Quadro P5000 GPU card with a batch size of 32. The model converges after about 10 epochs in total with pre-training initialization, which is described below.

Adding Pre-training. We pre-train a two-layer seq2seq model with OP as input and target argument as output from our training set. After 20 epochs (before converging), parameters for the first layer are used to initialize the first layer of all comparison models and our models (except for the keyphrase decoder). Experimental results show that pre-training boosts all methods by roughly 2 METEOR (Denkowski and Lavie, 2014) points, as illustrated in Figure 5.3.

5.6.3 Baseline and Comparisons

We first consider a RETRIEVAL-based baseline, which concatenates retrieved evidence sentences to form the argument. We further compare with three seq2seq-based generation models with different training data: (1) SEQ2SEQ: training with OP as input and the argument as output; (2) SEQ2SEQ + *encode evd*: augmenting input with evidence sentences as in our model; (3) SEQ2SEQ + *encode KP*: augmenting input with gold-standard keyphrases, which assumes some of the talking points are known. All seq2seq models use a regular beam search decoder with the same beam size as ours.

Variants of Our Models. We experiment with variants of our models based on the proposed separate decoder model (DEC-SEPARATE) or using a shared decoder (DEC-SHARED). For each, we further test whether adding keyphrase attention for argument decoding is helpful (+ *attend KP*).

System vs. Oracle Retrieval. For test time, evidence sentences are retrieved with queries constructed from OP (*System Retrieval*). We also experiment with an *Oracle Retrieval* setup, where the evidence is retrieved based on user arguments, to indicate how much gain can be expected with better retrieval results.

5.7 Results

5.7.1 Automatic Evaluation

For automatic evaluation, we use BLEU (Papineni et al., 2002), an n -gram precision-based metric (up to bigrams are considered), and METEOR (Denkowski and Lavie, 2014), measuring unigram recall and precision by considering paraphrases, synonyms, and stemming. Human arguments are used as the gold-standard. Because each OP may be paired with more than one high-quality arguments, we compute BLEU and METEOR scores for the system argument compared against all arguments, and report the best. We do not use multiple reference evaluation because the arguments are often constructed from different angles and cover distinct aspects of the issue. For models that generate more than one arguments based on different sets of sampled evidence, the one with the highest score is considered.

As can be seen from Table 5.4, our models produce better BLEU scores than almost all the comparisons. Especially, our models with separate decoder yield significantly higher BLEU and METEOR scores than all seq2seq-based models (approximation randomization testing, $p < 0.0001$) do. Better METEOR scores are achieved by the RETRIEVAL baseline, mainly due to its significantly longer arguments.

Moreover, utilizing attention over both input and the generated keyphrases further boosts our models' performance. Interestingly, utilizing system retrieved evidence yields better BLEU scores than using oracle retrieval for testing. The reason could be that arguments generated based on system retrieval

	<i>w/ System Retrieval</i>			<i>w/ Oracle Retrieval</i>		
	BLEU	METEOR	Length	BLEU	METEOR	Length
Baseline						
RETRIEVAL	15.32	12.19	151.2	10.24	16.22	132.7
Comparisons						
SEQ2SEQ	10.21	5.74	34.9	7.44	5.25	31.1
+ <i>encode evd</i>	18.03	7.32	67.0	13.79	10.06	68.1
+ <i>encode KP</i>	21.94	8.63	74.4	12.96	10.50	78.2
Our Models						
DEC-SHARED	21.22	8.91	69.1	15.78	11.52	68.2
+ <i>attend KP</i>	24.71	10.05	74.8	11.48	10.08	40.5
DEC-SEPARATE	24.24	10.63	88.6	17.48	13.15	86.9
+ <i>attend KP</i>	24.52	11.27	88.3	17.80	13.67	86.8

Table 5.4: Results on argument generation by BLEU and METEOR, with system retrieved evidence and oracle retrieval. The best performing model is highlighted in **bold** per metric. Our separate decoder models, with and without keyphrase attention, statistically significantly outperform all seq2seq-based models based on approximation randomization testing (Noreen, 1989), $p < 0.0001$.

contain less topic-specific words and more generic argumentative phrases. Since the later is often observed in human written arguments, it may lead to higher precision and thus better BLEU scores.

Decoder Strategy Comparison. We also study the effect of our reranking-based decoder by varying the reranking step size (p) and the number of top words expanded to beam hypotheses deterministically (k). From the results in Figure 5.4, we find that reranking with a smaller step size, e.g., $p = 5$, can generally lead to better METEOR scores. Although varying the number of top words for beam expansion does not yield significant difference, we do observe more diverse beams from the system output if more candidate words are selected stochastically (i.e. with a smaller k).

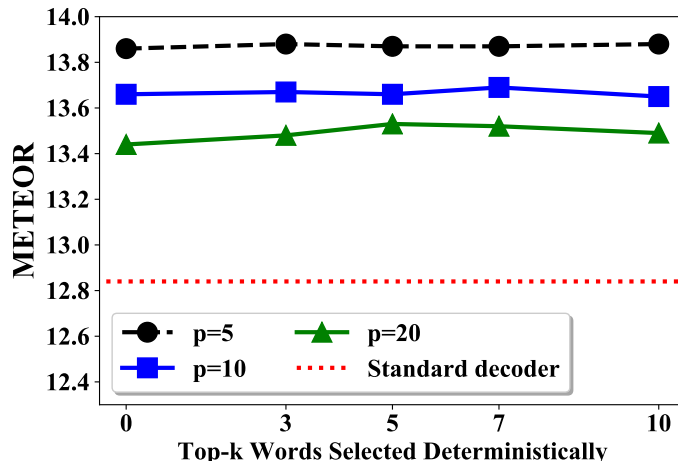


Figure 5.4: Effect of our reranking-based decoder. Beams are reranked at every 5, 10, and 20 steps (p). For each step size, we also show the effect of varying k , where top- k words are selected deterministically for beam expansion, with $10-k$ randomly sampled over multinomial distribution after removing the k words. Reranking with smaller step size yields better results.

5.7.2 Topic-Relevance Evaluation

During our pilot study, we observe that generic arguments, such as “I don’t agree with you” or “this is not true”, are prevalent among generations by seq2seq models. We believe that good arguments should include content that addresses the given topic. Therefore, we design a novel evaluation method to measure whether the generated arguments contain topic-relevant information.

To achieve the goal, we first train a topic-relevance estimation model inspired by the latent semantic model in [Huang et al. \(2013\)](#). A pair of OP and argument, each represented as the average of the 300-dimensional GloVe ([Pennington et al., 2014](#)) word embeddings, are separately fed into a two-layer transformation model. A dot-product is computed over the two projected low-dimensional vectors, and then a sigmoid function outputs the relevance score. For model learning, we further divide our current training data into training,

developing, and test sets. For each OP and argument pair, we first randomly sample 100 arguments from other threads, and then pick the top 5 dissimilar ones, measured by Jaccard distance, as negative training samples. This model achieves a Mean Reciprocal Rank (MRR) score of 0.95 on the test set.

We then take this trained model to evaluate the relevance between OP and the corresponding system arguments. Each system argument is treated as positive sample; we then select five negative samples from arguments generated for other OPs whose evidence sentences most similar to that of the positive sample.

Intuitively, if an argument contains more topic relevant information, then the relevance estimation model will output a higher score for it; otherwise, the argument will receive a lower similarity score, and thus cannot be easily distinguished from negative samples. Ranking metrics of MRR and Precision at 1 (P@1) are utilized, with results reported in Table 5.5. The ranker yields significantly better scores over arguments generated from models trained with evidence, compared to arguments generated by SEQ2SEQ model.

Moreover, we manually pick 29 commonly used generic responses (e.g., “I don’t think so”) and count their frequency in system outputs. For the seq2seq model, more than 75% of its outputs contain at least one generic argument, compared to 16.2% by our separate decoder model with attention over keyphrases. This further implies that our model generates more topic-relevant content.

	Standard Decoder		Our Decoder	
	MRR	P@1	MRR	P@1
Baseline				
RETRIEVAL	81.08	65.45	-	-
Comparisons				
SEQ2SEQ	75.29	58.85	74.46	57.06
+ <i>encode evd</i>	83.73	71.59	88.24	78.76
Our Models				
DEC-SHARED	79.80	65.57	95.18	90.91
+ <i>attend KP</i>	94.33	89.76	93.48	87.91
DEC-SEPARATE	86.85	76.74	91.70	84.72
+ <i>attend KP</i>	88.53	79.05	92.77	86.46

Table 5.5: Evaluation on topic relevance—models that generate arguments highly related with OP should be ranked high by a separately trained relevance estimation model, i.e., higher Mean Reciprocal Rank (MRR) and Precision at 1 (P@1) scores. All models trained with evidence significantly outperform seq2seq trained without evidence (approximation randomization testing, $p < 0.0001$).

5.7.3 Human Evaluation

We also hire three trained human judges who are fluent English speakers to rate system arguments for the following three aspects on a scale of 1 to 5 (with 5 as best): *Grammaticality*—whether an argument is fluent, *informativeness*—whether the argument contains useful information and is not generic, and *relevance*—whether the argument contains information of a different stance or off-topic. 30 CMV threads are randomly selected, each of which is presented with randomly-shuffled OP statement and four system arguments.

Table 5.6 shows that our model with separate decoder and attention over keyphrases produce significantly more informative and relevant arguments than seq2seq trained without evidence.⁵ However, we also observe that hu-

⁵Inter-rater agreement scores for these three aspects are 0.50, 0.60, and 0.48 by Krippendorff’s α .

System	Grammaticality	Informativeness	Relevance
RETRIEVAL	4.5 ± 0.6	3.7 ± 0.9	3.3 ± 1.1
SEQ2SEQ	3.3 ± 1.1	1.2 ± 0.5	1.4 ± 0.7
OUR MODEL	2.5 ± 0.8	1.6 ± 0.8	1.8 ± 0.8

Table 5.6: Human evaluation results on grammaticality, informativeness, and relevance of arguments. Our model with separate decoder and attention over keyphrases receives significantly better ratings in informativeness and relevance than seq2seq (one-way ANOVA, $p < 0.005$).

man judges prefer the retrieved arguments over generation-based models, illustrating the gap between system arguments and human edited text. Sample arguments are displayed in Figure 5.5.

5.8 Further Discussion

Keyphrase Generation Analysis. Here we provide further analysis over the generated keyphrases by our separate decoder model. First, about 10% of the keyphrases output by our model also appear in the gold-standard (i.e., used by human arguments). Furthermore, 36% of generated keyphrases are reused by our system arguments. With human inspection, we find that although some keyphrases are not directly reused by the argument decoder, they represent high level talking points in the argument. For instance, in the first sample argument by our model in Figure 5.5, keyphrases “the motive” and “russian” are generated. Although not used, they suggest the topics that the argument should stay on.

Sample Arguments and Future Directions. As can be seen from the sample outputs in Figure 5.5, our model generally captures more relevant concepts,

Input: Putin is trying to re-form a “Soviet Union” with his past actions in Georgia and current actions in Ukraine

Human: So far Putin has only really targeted areas where the local population is majority russian (or historically majority russian) and sympathetic to russia. That’s not to say that he wouldn’t go further, but I don’t think there is yet any compelling reason to believe he would.

Retrieval: In 1968, the Soviet Union and Warsaw pact allies invaded Czechoslovakia to halt the Prague spring reforms. It was this significant lack of flexibility that was exploited by the US to undermine the strength of the Soviet Union and thus foster its reform. In the summer of 1991, the new union treaty, which would have turned the Soviet Union into a much looser union, was agreed upon by eight republics.

Our Model: There are two reasons you are so far beyond. There is no reason to see the military army. You can infer what they assume it, so they tend to protect up. There. If we assume it were bad, they can not be controlled. So we’re talking going to ignore the wars of the world. The fact of the matter is not bad. I would also assume it. However, the government can not be reminded of the world.

Seq2seq + evd: 1. The reason is that the government isn’t the power of the war of Russia, at the time of the war of Russia. In Russia, there is no way to continue to control the military.

Seq2seq + KP: I don’t think it’s a bit. I don’t think you’re forgetting. I don’t want to see the Syrian American dream in the US. I don’t see the Syrian American dream of American citizens. I don’t see the Syrian American dream of American citizens. I don’t see the Syrian American military in the world. I don’t think it wouldn’t be true.

Input: Voting for either Hillary or Trump is morally reprehensible.

Human: I don’t think a vote for either of them means “I’m totally fine with...” as you say. Why is it not acceptable to pick the option whose platform most closely aligns to your values/beliefs/desires, while acknowledging neither candidate is ideal?

Retrieval: Vote trading under a democratic, majority-rule institution is sometimes considered morally reprehensible behavior.

Our Model: He’s going to happen. You’re overestimating the more premise. He’s already overestimating to this hypothetical party. He’s already killed over the supreme court positions. Looking for Bernie. This is the same reasons: Clinton is not the supreme court presidency. You’ve already sent Trump’s right to vote. The fact is that he’s going to happen. He’s already going to notice. However, it’s a long way.

Seq2seq + evd: 1. If they don’t vote the vote, they would not be able to vote for any candidate. They don’t have the same effect on their political power. They are not voting for them.

Seq2seq + KP: I agree with your view. I don’t agree with you. I don’t think it’s easy to appeal to the Clintons. If you don’t want to do this?

Figure 5.5: Sample arguments generated by human, our system, and seq2seq trained with evidence. Only the main thesis is shown for the input OP. System generations are manually detokenized and capitalized.

e.g., “military army” and “wars of the world”, as discussed in the first example. Meanwhile, our model also acquires argumentative style language, though there is still a noticeable gap between system arguments and human constructed arguments. As discovered by our prior work (Wang et al., 2017b), both topical

content and language style are essential elements for high quality arguments. For future work, generation models with a better control on linguistic style need to be designed. As for improving coherence, we believe that discourse-aware generation models (Ji et al., 2016) should also be explored in the future work to enhance text planning.

5.9 Conclusion

We studied the novel problem of generating arguments of a different stance for a given statement. We presented a neural argument generation framework enhanced with evidence retrieved from Wikipedia. Separate decoders were designed to first produce a set of keyphrases as talking points, and then generate the final argument. Both automatic evaluation against human arguments and human assessment showed that our model produced more informative arguments than popular sequence-to-sequence-based generation models.

CHAPTER 6

ARGUMENT GENERATION WITH RETRIEVAL, PLANNING, AND REALIZATION

In this chapter, we introduce an improved argument generation system with retrieval on both Wikipedia evidence and opinions from news articles. We further augment the neural generator with a text planning module, which selects keyphrases to include for each sentence and simultaneously predicts an argumentative function.

This work is published at ACL 2019 (Hua et al., 2019a). Relevant resources can be found at: <https://xinyuhua.github.io/Resources/acl19/>.

6.1 Introduction

Counter-argument generation aims to produce arguments of a different stance, in order to refute the given proposition on a controversial issue (Toulmin, 1958; Damer, 2012). A system that automatically constructs counter-arguments can effectively present alternative perspectives along with associated evidence and reasoning, and thus facilitate a more comprehensive understanding of complicated problems when controversy arises.

Nevertheless, constructing persuasive arguments is a challenging task, as it requires an appropriate combination of credible evidence, rigorous logical reasoning, and sometimes emotional appeal (Walton et al., 2008; Wachsmuth et al., 2017a; Wang et al., 2017b). A sample counter-argument for a pro-death penalty post is shown in Figure 6.1. As can be seen, a sequence of talking points on the

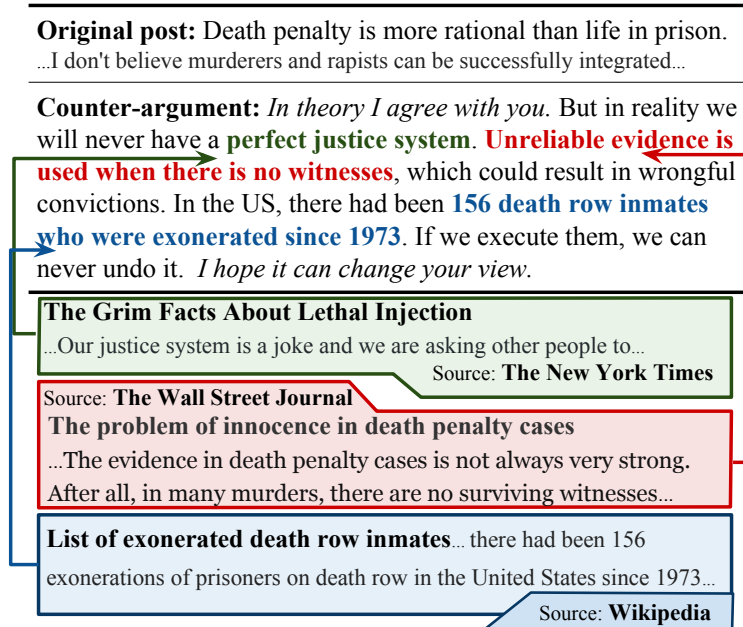


Figure 6.1: Sample counter-argument for a pro-death penalty statement from Reddit /r/ChangeMyView. The argument consists of a sequence of propositions, by synthesizing opinions and facts from diverse sources. Sentences in italics contain stylistic languages for argumentation purpose.

“imperfect justice system” are presented: it starts with the fundamental concept, then follows up with more specific evaluative claim and supporting fact. Although retrieval-based methods have been investigated to construct counter-arguments (Sato et al., 2015; Reiser et al., 2015), they typically produce a collection of sentences from disparate sources, thus fall short of coherence and conciseness. Moreover, human always deploy *stylistic languages* with specific argumentative functions to promote persuasiveness, such as making a concessive move (e.g., “*In theory I agree with you*”). This further requires the generation system to have better control of the languages style.

Our goal is to design *a counter-argument generation system to address the above challenges and produce paragraph-level arguments with rich-yet-coherent content*. To this end, we present CANDELA—a novel framework to generate

Counter-Arguments with two-step Neural Decoders and ExternaL knowledge Augmentation. Concretely, CANDELA has three major distinct features:

First, it is equipped with two decoders: one for **text planning**—selecting talking points to cover for *each sentence* to be generated, the other for **content realization**—producing a fluent argument to reflect decisions made by the text planner. This enables our model to produce longer arguments with richer information.

Furthermore, multiple objectives are designed for our text planning decoder to both handle content selection and ordering, and select a proper argumentative discourse function of a desired language style for each sentence generation.

Lastly, the input to our argument generation model is augmented with keyphrases and passages retrieved from a large-scale search engine, which indexes 12 million articles from Wikipedia and four popular English news media of varying ideological leanings. This ensures access to reliable evidence, high-quality reasoning, and diverse opinions from different sources, as opposed to recent work that mostly considers a single origin, such as Wikipedia (Rinott et al., 2015) or online debate portals (Wachsmuth et al., 2018b).

We experiment with argument and counter-argument pairs collected from the Reddit ChangeMyView. Automatic evaluation shows that the proposed model significantly outperforms our prior argument generation system (Hua and Wang, 2018) and other non-trivial comparisons. Human evaluation further suggests that our model produces more appropriate counter-arguments with richer content than other automatic systems, while maintaining a fluency level comparable to human-constructed arguments.

6.2 Overview of CANDELA

Our counter-argument generation framework, as shown in Figure 6.2, has two main components: **argument retrieval** model (§ 6.3) that takes the input statement and a search engine, and outputs relevant passages and keyphrases, which are used as input for our **argument generation** model (§ 6.4) to produce a fluent and informative argument.

Concretely, the argument retrieval component retrieves a set of candidate passages from Wikipedia and news media (§ 6.3.1), then further selects passages according to their stances towards the input statement (§ 6.3.3). A **keyphrase extraction** module distills the refined passages into a set of talking points, which comprise the keyphrase memory as additional input for generation (§ 6.3.2).

The argument generation component first runs the **text planning decoder** (§ 6.4.2) to produce a sequence of hidden states, each corresponding to a *sentence-level representation* that encodes the selection of keyphrases to cover, as well as the predicted argumentative function for a desired language style. The **content realization decoder** (§ 6.4.3) then generates the argument conditioned on the sentence representations.

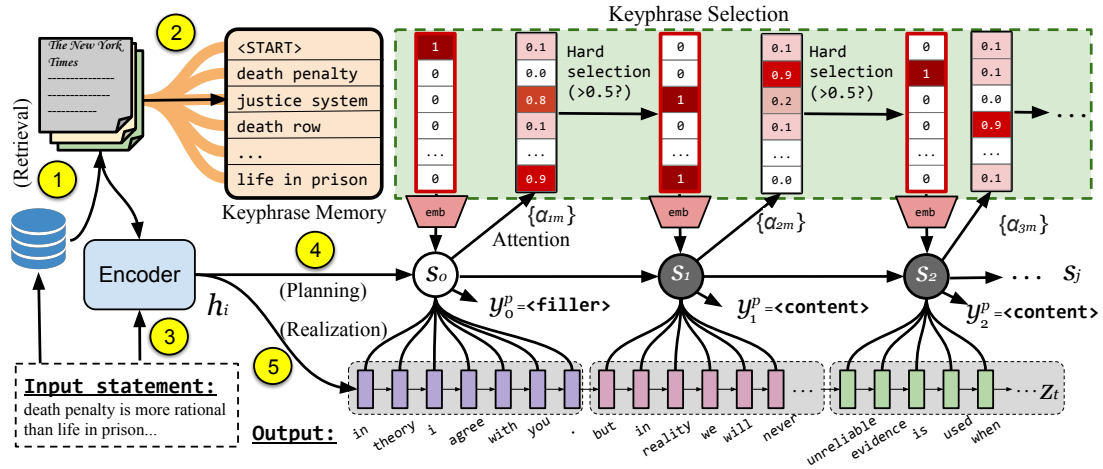


Figure 6.2: Architecture of CANDELA. ① Argument Retrieval (§ 6.3): a set of passages are retrieved and ranked based on relevance and stance (§ 6.3.1, § 6.3.3), from which ② a set of keyphrases are extracted (§ 6.3.2), with both as input for argument generation. ③ The biLSTM encoder consumes the input statement and passages returned from step 1. ④ A text planning decoder outputs a representation per sentence, and simultaneously predicts an argumentative function and selects keyphrases to include for the next sentence to be generated (§ 6.4.2). ⑤ A content realization decoder produces the counter-argument (§ 6.4.3).

6.3 Argument Retrieval

6.3.1 Information Sources and Indexing

We aim to build a search engine from diverse information sources with factual evidence and varied opinions of high quality. To achieve that, we use Common Crawl¹ to collect a large-scale online news dataset covering four major English news media: The New York Times (NYT), The Washington Post (WaPo), Reuters, and The Wall Street Journal (WSJ). HTML files are processed using the open-source tool jusText (Pomikálek, 2011) to extract article content. We deduplicate articles and remove the ones with less than 50 words.

¹<http://commoncrawl.org/>

Source	# Articles	# Passages	Date Range
Wikipedia	5,743,901	42,797,543	dump of 12/2016
WaPo	1,109,672	22,564,532	01/1997 - 10/2018
NYT	1,952,446	28,904,549	09/1895 - 09/2018
Reuters	1,052,592	9,913,400	06/2005 - 09/2018
WSJ	2,059,128	16,109,392	01/1996 - 09/2018
Total	11,917,739	120,289,416	-

Table 6.1: Statistics on information sources for argument retrieval. News media are sorted by ideological leanings from left to right, according to <https://www.adfontesmedia.com/>.

We also download a Wikipedia dump. About 12 million articles are processed in total, with basic statistics shown in Table 6.1.

We segment articles into passages with a sliding window of three sentences, with a step size of two. We further constraint the passages to have at least 50 words. For shorter passages, we keep adding subsequent sentences until reaching the length limit. Per Table 6.1, 120 million passages are preserved and indexed with Elasticsearch (Gormley and Tong, 2015) as done in Stab et al. (2018).

Query Formulation. For an input statement with multiple sentences, one query is constructed per sentence, if it has more than 5 content words (10 for questions), and at least 3 are distinct. For each query, the top 20 passages ranked by BM25 (Robertson et al., 1995) are retained, per medium. All passages retrieved for the input statement are merged and deduplicated, and they will be ranked as discussed in § 6.3.3.

6.3.2 Keyphrase Extraction

Here we describe a keyphrase extraction procedure for both input statements and retrieved passages, which will be utilized for passage ranking as detailed in the next section.

For *input statement*, our goal is to identify a set of phrases representing the issues under discussion, such as “death penalty” in Figure 6.1. We thus first extract the topic signature words (Lin and Hovy, 2000a) for input representation, and expand them into phrases that better capture semantic meanings.

Concretely, topic signature words of an input statement are calculated against all input statements in our training set with log-likelihood ratio test. In order to cover phrases with related terms, we further expand this set with their synonyms, hyponyms, hypernyms, and antonyms based on WordNet (Miller, 1994). The statements are first parsed with Stanford part-of-speech tagger (Manning et al., 2014). Then we apply the regular expression grammar, as shown below, to extract candidate noun phrases and verb phrases.

NP : {<DT PP\$>?<JJ JJR>*<NN . * CD JJ>+}
PP : {<IN><NP>}
VP : {<MD>?<VB . *><NP PP>}

A keyphrase is selected if it contains: (1) at least one content word, (2) no more than 10 tokens, and (3) at least one topic signature word or a Wikipedia article title.

For *retrieved passages*, their keyphrases are extracted using the same procedure as above, except that the input statement’s topic signature words are used as references again.

6.3.3 Passage Ranking and Filtering

We merge the retrieved passages from all media and rank them based on the number of words in overlapping keyphrases with the input statement. To break a tie, with the input as the reference, we further consider the number of its topic signature words that are covered by the passage, then the coverage of non-stopword bigrams and unigrams. In order to encourage diversity, we discard a passage if more than 50% of its content words are already included by a higher ranked passage. In the final step, we filter out passages if they have the same stance as the input statement for given topics.

We determine the stances of passages by adopting the **stance scoring model** proposed by Bar-Haim et al. (2017). Concretely, we aggregate the sentiment words surrounding the opinion targets. Here we choose the keyphrases of input statement as opinion targets, denoted as \mathbb{T} . We then tally sentiment words, collected from Hu and Liu (2004), towards targets in \mathbb{T} , with positive words counted as +1 and negative words as -1. Each score is discounted by $d_{\tau,l}^{-5}$, with $d_{\tau,l}$ being the distance between the sentiment word l and the target $\tau \in \mathbb{T}$. The stance score of a text psg (an input statement or a retrieved passage) towards opinion targets \mathbb{T} is calculated as:

$$Q(psg, \mathbb{T}) = \sum_{\tau \in \mathbb{T}} \sum_{l \in psg} \text{sgn}(l) \cdot d_{\tau,l}^{-5} \quad (6.1)$$

In our experiments, we only keep passages with a stance score of the opposite sign to that of the input statement, and with a magnitude greater than 5, i.e. $|Q(psg, \mathbb{T})| > 5$ (determined by manual inspection on training set).

6.4 Argument Generation

6.4.1 Task Formulation

Given an input statement $X = \{x_i\}$, a set of passages, and a keyphrase memory \mathcal{M} , our goal is to generate a counter-argument $Y = \{y_t\}$ of a different stance as X , x_i and y_t are tokens at timestamps i and t . Built upon the sequence-to-sequence (seq2seq) framework with input attention (Sutskever et al., 2014; Bahdanau et al., 2015), the input statement and the passages selected in § 6.3 are encoded by a bidirectional LSTM (biLSTM) encoder into a sequence of hidden states h_i . The last hidden state of the encoder is used as the first hidden state of both text planning decoder and content realization decoder.

As depicted in Figure 6.2, the counter-argument is generated as follows. A text planning decoder (§ 6.4.2) first calculates a sequence of sentence representations s_j (for the j -th sentence) by encoding the keyphrases selected from the previous timestamp $j-1$. During this step, an *argumentative function* label is predicted to indicate a desired language style for each sentence, and a subset of the keyphrases are selected from \mathcal{M} (*content selection*) for the next sentence. In the second step, a content realization decoder (§ 6.4.3) generates the final counter-argument conditioned on previously generated tokens and the corresponding sentence representation s_j .

6.4.2 Text Planning Decoder

Text planning is an important component for natural language generation systems to decide on content structure for the target generation (Lavoie and Rainbow, 1997; Reiter and Dale, 2000). We propose a text planner with two objectives: selecting talking points from the keyphrase memory \mathcal{M} , and choosing a proper argumentative function per sentence.

$$\mathbf{s}_j = f(\mathbf{s}_{j-1}, \sum_{e_k \in \mathbb{C}(j)} \mathbf{e}_k) \quad (6.2)$$

where f is an LSTM network, \mathbf{e}_k is the embedding for a selected phrase, represented by summing up all its words' Glove embeddings (Pennington et al., 2014) in our experiments.

Content Selection $\mathbb{C}(j)$. We propose an attention mechanism to conduct content selection and yield $\mathbb{C}(j)$ from the representation of the previous sentence \mathbf{s}_{j-1} to encourage topical coherence. To allow the selection of multiple keyphrases, we use the sigmoid function to calculate the score:

$$\alpha_{jm} = \text{sigmoid}(\mathbf{e}_m \mathbf{W}^{pa} \mathbf{s}_{j-1}) \quad (6.3)$$

where \mathbf{W}^{pa} are trainable parameters, keyphrases with $\alpha_{jm} > 0.5$ are included in $\mathbb{C}(j)$, and the keyphrase with top attention value is always selected. We further prohibit a keyphrase from being chosen for more than once in multiple sentences. For the first sentence \mathbf{s}_0 , $\mathbb{C}(0)$ only contains `<start>`, whose embedding is randomly initialized. During training, the *true labels* of $\mathbb{C}(j)$ are constructed as follows: a keyphrase in \mathcal{M} is selected for the j -th gold-standard argument sentence if they overlap with any content word.

Argumentative Function Prediction y_j^p . As shown in Figure 6.1, humans often

deploy stylistic languages to achieve better persuasiveness, e.g. agreement as a concessive move. We aim to inform the realization decoder about the choice of style, and thus distinguish between two types of argumentative functions: **argumentative content sentence** which delivers the critical ideas, e.g. “*unreliable evidence is used when there is no witness*”, and **argumentative filler sentence** which contains stylistic languages or general statements (e.g., “*you can’t bring dead people back to life*”).

Since we do not have argumentative function labels, during training, we use the following rules to automatically label each sentence as *content sentence* if it has at least 10 words (20 for questions) and satisfy the following conditions: (1) it has at least two topic signature words of the input statement or a gold-standard counter-argument², or (2) at least one topic signature word with a discourse marker at the beginning of the sentence. If the first three words in a *content sentence* contain a pronoun, the previous sentence is labeled as such too. Discourse markers are manually selected from the Appendix B in the PDTB manual (Prasad et al., 2008), as listed below:

- **Contrast:** although, though, even though, by comparison, by contrast, in contrast, however, nevertheless, nonetheless, on the contrary, regardless, whereas
- **Restatement/Equivalence/Generalization:** eventually, in short, in sum, on the whole, overall
- **Result:** accordingly, as a result, as it turns out, consequently, finally, furthermore, hence, in fact, in other words, in short, in the end, in turn, therefore, thus, ultimately

²When calculating topic signatures for gold-standard arguments, all replies in the training set are used as background.

All other sentences become *filler sentences*. In the future work, we will consider utilizing learning-based methods, e.g., [Hidey et al. \(2017\)](#), to predict richer argumentative functions.

The argumentative function label y_j^p for the j -th sentence is calculated as follows:

$$P(y_j^p | y_{<j}^p, \mathbf{X}) = \text{softmax}(\mathbf{w}_p^T (\tanh(\mathbf{W}^{po}[\mathbf{c}_j; \mathbf{s}_j])) + \mathbf{b}_p) \quad (6.4)$$

$$\mathbf{c}_j = \sum_{e_m \in \mathcal{M}} \alpha_{jm} \mathbf{e}_m \quad (6.5)$$

where α_{jm} is the alignment score computed as in Eq. 6.3, \mathbf{c}_j is the attention weighted context vector, \mathbf{w}_p , \mathbf{W}^{po} , and \mathbf{b}_p are trainable parameters.

6.4.3 Content Realization Decoder

The content realization decoder generates the counter-argument word by word, with another LSTM network f^w . We denote the sentence id of the t -th word in the argument as $J(t)$, then the sentence representation $\mathbf{s}_{J(t)}$ from the text planning decoder, together with the embedding of the previous generated token \mathbf{y}_{t-1} , are fed as input to calculate the hidden state \mathbf{z}_t :

$$\mathbf{z}_t = f^w(\mathbf{z}_{t-1}, \tanh(\mathbf{W}^{wp} \mathbf{s}_{J(t)} + \mathbf{W}^{ww} \mathbf{y}_{t-1} + \mathbf{b}^w)) \quad (6.6)$$

The conditional probability of the next token y_t is then computed over a standard softmax, with an attention mechanism applied on the encoder hidden

states \mathbf{h}_i to obtain the context vector \mathbf{c}_t^w :

$$P(y_t|y_{<t}, \mathbf{X}, s_{J(t)}) = \text{softmax}(\mathbf{w}_w^T (\tanh(\mathbf{W}^{wo}[\mathbf{c}_t^w; \mathbf{z}_t])) + \mathbf{b}^o) \quad (6.7)$$

$$\mathbf{c}_t^w = \sum_{i=1}^{|\mathbf{X}|} \beta_{ti} \mathbf{h}_i \quad (6.8)$$

$$\beta_{ti} = \text{softmax}(\mathbf{h}_i \mathbf{W}^{wa} \mathbf{z}_t) \quad (6.9)$$

where β_{ti} is the input attention, \mathbf{W}^{wp} , \mathbf{W}^{ww} , \mathbf{W}^{wo} , \mathbf{W}^{wa} , \mathbf{b}^o , \mathbf{w}_w , and \mathbf{b}^w are learnable.

Reranking-based Beam Search. Our content realization decoder utilizes beam search enhanced with a reranking mechanism, where we sort the beams at the end of each sentence by the number of selected keyphrases that are generated. We also discard beams with n -gram repetition for $n \geq 4$.

6.4.4 Training Objective

Given all model parameters θ , our mixed objective considers the target argument ($\mathcal{L}_{\text{arg}}(\theta)$), the argumentative function type ($\mathcal{L}_{\text{func}}(\theta)$), and the next sentence keyphrase selection ($\mathcal{L}_{\text{sel}}(\theta)$):

$$\mathcal{L}(\theta) = \mathcal{L}_{\text{arg}}(\theta) + \gamma \cdot \mathcal{L}_{\text{func}}(\theta) + \eta \cdot \mathcal{L}_{\text{sel}}(\theta) \quad (6.10)$$

$$\mathcal{L}_{\text{arg}}(\theta) = - \sum_{(X,Y) \in D} \log P(Y|X; \theta) \quad (6.11)$$

$$\mathcal{L}_{\text{func}}(\theta) = - \sum_{(X,Y^p)} \log P(Y^p|X; \theta) \quad (6.12)$$

$$\mathcal{L}_{\text{sel}}(\theta) = - \sum_{Y^p} \sum_{j=1}^{|Y^p|} \left(\sum_{e_m \in \mathbb{C}(j)} \log(\alpha_{jm}) + \sum_{e_m \notin \mathbb{C}(j)} \log(1 - \alpha_{jm}) \right) \quad (6.13)$$

where D is the training corpus, (X, Y) are input statement and counter-argument pairs, and Y^p are the sentence function labels. α_{jm} are keyphrase selection labels

as computed in Eq. 6.3. For simplicity, we set γ and η as 1.0 in our experiments, while they can be further tuned as hyper-parameters.

6.5 Experimental Setups

6.5.1 Data Collection and Preprocessing

We use the same methodology as in our prior work (Hua and Wang, 2018) to collect an argument generation dataset from Reddit /r/ChangeMyView.³ To construct input statement and counter-argument pairs, we treat the original poster (OP) of each thread as the input. We then consider the high quality root replies, defined as the ones awarded with Δ s or with more upvotes than downvotes (i.e., $\text{karma} > 0$). It is observed that each paragraph often makes a coherent argument. Therefore, these replies are broken down into paragraphs, and a paragraph is retained as a target argument to the OP if it has more than 10 words and at least one argumentative content sentence.

We then identify threads in the domains of politics and policy, and remove posts with offensive languages. Most recent threads are used as test set. As a result, we have 11,356 threads or OPs (217,057 arguments) for training, 1,774 (33,318 arguments) for validation, and 1,703 (36,777 arguments) for test. They are split into sentences and then tokenized by the Stanford CoreNLP toolkit (Manning et al., 2014).

Training Data Construction for Passages and Keyphrase Memory. Since no

³We further crawled 42,649 threads from July 2017 to December 2018, compared to the previously collected dataset.

gold-standard annotation is available for the input passages and keyphrases, we acquire training labels by constructing queries from the gold-standard arguments as described in § 6.3.1, and reranking retrieved passages based on the following criteria in order: (1) coverage of topic signature words in the input statement; (2) a weighted summation of the coverage of n -grams in the argument⁴; (3) the magnitude of stance score, where we keep the passages of the same polarity as the argument; (4) content word overlap with the argument; and (5) coverage of topic signature words in the argument.

6.5.2 System and Oracle Retrieved Passages

For evaluation, we employ both *system* retrieved passages (i.e., constructing queries from OP) and KM (§ 6.3), and *oracle* retrieved passages (i.e., constructing queries from target argument) and KM as described in training data construction. Statistics on the final dataset are listed in Table 6.2.

	<i>Training</i>	<i>System</i>	<i>Oracle</i>
Avg. # words per OP	383.7	373.0	373.0
Avg. # words per argument	66.0	65.1	65.1
Avg. # passage	4.3	9.6	4.2
Avg. # keyphrase	57.1	128.6	56.6

Table 6.2: Statistics on the datasets for experiments.

6.5.3 Comparisons

In addition to a **Retrieval** model, where the top ranked passage is used as counter-argument, we further consider four systems for comparison. (1) A stan-

⁴We choose 0.5, 0.3, 0.2 as weights for 4-grams, trigrams, and bigrams, respectively.

dard **Seq2seq** model with attention, where we feed the OP as input and train the model to generate counter-arguments. Regular beam search with the same beam size as our model is used for decoding. (2) A **Seq2seqAug** model with additional input of the keyphrase memory and ranked passages, both concatenated with OP to serve as the encoder input. The reranking-based decoder in our model is also implemented for SEQ2SEQAUG to enhance the coverage of input keyphrases. (3) An ablated SEQ2SEQAUG model where the passages are removed from the input. (4) We also reimplement the argument generation model in our prior work (Hua and Wang, 2018) (H&W) with PyTorch (Paszke et al., 2017), which is used for CANDELA implementation. H&W takes as input the OP and ranked passages, and then uses two separate decoders to first generate all keyphrases and then the counter-argument. For our model, we also implement a variant where the input only contains the OP and the keyphrase memory.

6.5.4 Training Details

For all models, we use a two-layer LSTM for all encoders and decoders with a dropout probability of 0.2 between layers (Gal and Ghahramani, 2016). All layers have 512-dimensional hidden states. We limit the input statement to 500 tokens, the ranked passages to 400 tokens, and the target counter-argument to 120 tokens. Our vocabulary has 50K words for both input and output, with 300-dimensional word embeddings initialized with GloVe (Pennington et al., 2014) and fine-tuned during model training. We use AdaGrad (Duchi et al., 2011) with a learning rate of 0.15 and an initial accumulator of 0.1 as the optimizer, with the gradient norm clipped to 2.0. Early stopping is implemented according to the

perplexity on validation set. For all our models the training takes approximately 30 hours (40 epochs) on a Quadro P5000 GPU card, with a batch size of 64. For beam search, we use a beam size of 5, tuned from {5, 10, 15} on validation.

We also pre-train a biLSTM for encoder based on all OPs from the training set, and an LSTM for content realization decoder based on two sources of data: 353K counter-arguments that are high quality root reply paragraphs extended with posts of non-negative karma, and 2.4 million retrieved passages randomly sampled from the training set. Both are trained as done in [Bengio et al. \(2003\)](#). We then use the first layer’s parameters to initialize all models, including our comparisons.

6.6 Results and Analysis

6.6.1 Automatic Evaluation

We employ ROUGE ([Lin, 2004](#)), a recall-oriented metric, BLEU ([Papineni et al., 2002](#)), based on n -gram precision, and METEOR ([Denkowski and Lavie, 2014](#)), measuring unigram precision and recall by considering synonyms, paraphrases, and stemming. BLEU-2, BLEU-4, ROUGE-2 recall, and METEOR are reported in [Table 6.3](#) for system setup and [Table 6.4](#) for oracle setup.

Under system setup, our model CANDELA statistically significantly outperforms all comparisons and the retrieval model in all metrics, based on a randomization test ([Noreen, 1989](#)) ($p < 0.0005$). Furthermore, our model generates longer sentences whose lengths are comparable with human arguments, both

	<i>w/ System Retrieval</i>					
	BLEU-2	BLEU-4	ROUGE-2	METEOR	#Word	#Sent
HUMAN	-	-	-	-	66	22
RETRIEVAL	7.55	1.11	8.64	14.38	123	23
Comparisons						
SEQ2SEQ	6.92	2.13	13.02	15.08	68	15
SEQ2SEQAUG	8.26	2.24	13.79	15.75	78	14
<i>w/o psg</i>	7.94	2.28	10.13	15.71	75	12
H&W (2018)	3.64	0.92	8.83	11.78	51	12
Our Models						
CANDELA	12.02*	2.99*	14.93*	16.92*	119	22
<i>w/o psg</i>	12.33*	2.86*	14.53*	16.60*	123	23

Table 6.3: Main results on argument generation using system retrieval. We report BLEU-2, BLEU-4, ROUGE-2 recall, METEOR, and average number of words per argument and per sentence. Best scores are in bold. *: statistically significantly better than all comparisons (randomization approximation test (Noreen, 1989), $p < 0.0005$).

with about 22 words per sentence. This also results in longer arguments. Under oracle setup, all models are notably improved due to the higher quality of reranked passages, and our model achieves statistically significantly better BLEU scores. Interestingly, we observe a decrease of ROUGE and METEOR, but a marginal increase of BLEU-2 by removing passages from our model input. This could be because the passages introduce divergent content, albeit probably on-topic, that cannot be captured by BLEU.

Content Diversity. We further measure whether our model is able to generate diverse content. First, borrowing the diversity measurement from dialogue generation research (Li et al., 2016), we report the average number of distinct n -grams per argument under system setup in Figure 6.3. Our system generates more unique unigrams and bigrams than other automatic systems, underscoring its capability of generating diverse content. Our model also maintains a

	<i>w/ Oracle Retrieval</i>					
	BLEU-2	BLEU-4	ROUGE-2	METEOR	#Word	#Sent
HUMAN	-	-	-	-	66	22
RETRIEVAL	10.97	3.05	23.49	20.08	140	21
Comparisons						
SEQ2SEQ	6.92	2.13	13.02	15.08	68	15
SEQ2SEQAUG	10.98	4.41	22.97	19.62	71	14
<i>w/o psg</i>	9.89	3.34	14.20	18.40	66	12
H&W (2018)	8.51	2.86	18.89	17.18	58	12
Our Models						
CANDELA	15.80*	5.00*	23.75	20.18	116	22
<i>w/o psg</i>	16.33*	4.98*	23.65	19.94	123	23

Table 6.4: Results using oracle retrieved passages. *: statistically significantly better than all comparisons (randomization approximation test (Noreen, 1989), $p < 0.0005$). Input is the same for SEQ2SEQ for both system and oracle setups.

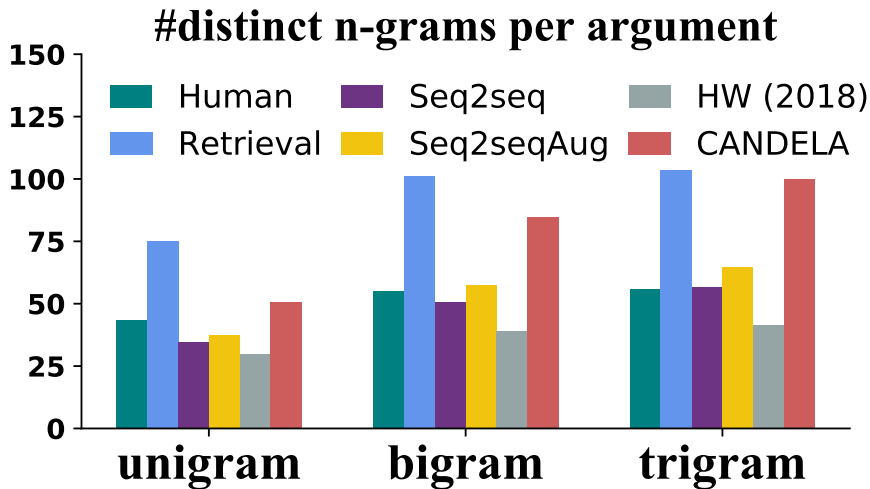


Figure 6.3: Average number of distinct n -grams per argument.

comparable type-token ratio (TTR) compared to systems that generate shorter arguments, e.g., a 0.79 for bigram TTR of our model versus 0.83 and 0.84 for SEQ2SEQAUG and SEQ2SEQ. RETRIEVAL, containing top ranked passages of human-edited content, produces the most distinct words.

Next, we compare how each system generates content beyond the common

	K			
	100	500	1000	2000
HUMAN	44.1	25.8	18.5	12.0
RETRIEVAL	50.6	33.3	26.0	18.6
SEQ2SEQ	25.0	7.5	3.2	1.2
SEQ2SEQAUG	28.2	9.2	4.6	1.8
H&W (2018)	38.6	24.0	19.5	16.2
CANDELA	30.0	10.5	5.3	2.3

Figure 6.4: Percentage of words in arguments that are not in the top- K ($K = 100, 500, 1000, 2000$) frequent words seen in training. Darker color indicates higher portion of uncommon words found in the arguments.

words. As shown in Figure 6.4, human-edited text, including gold-standard arguments (HUMAN) and retrieved passages, tends to have higher usage of uncommon words than automatic systems, suggesting the gap between human vs. system arguments. Among the four automatic systems, our prior model (Hua and Wang, 2018) generates a significantly higher portion of uncommon words, yet further inspection shows that the output often includes more off-topic information.

6.6.2 Human Evaluation

Human judges are asked to rate arguments on a Likert scale of 1 (worst) to 5 (best) on the following three aspects: **grammaticality**—denotes language fluency; **appropriateness**—indicates if the output is on-topic and on the opposing stance; **content richness**—measures the amount of distinct talking points. In order to promote consistency of annotation, we provide descriptions and sample arguments for each scale. For example, an appropriateness score of 3 means the counter-argument contains relevant words and is likely to be on a different stance. The judges are then asked to rank all arguments for the same input

	Gram.	Appr.	Cont.	Top-1	Top-2
HUMAN	4.95	4.23	4.39	75.8%	85.8%
RETRIEVAL	4.85	3.04	3.68	17.5%	55.8%
SEQ2SEQAUG	4.83	2.67	2.47	1.7%	22.5%
H&W (2018)	3.86	2.27	2.10	1.7%	7.5%
CANDELA	4.59	2.97	2.93*	3.3%	28.3%

Table 6.5: Human evaluation on grammaticality (Gram), appropriateness (Appr), and content richness (Cont.), on a scale of 1 to 5 (best). The best result among automatic systems is highlighted in bold, with statistical significance marked with * (approximation randomization test, $p < 0.0005$). The highest standard deviation among all is 1.0. Top-1/2: % of evaluations a system being ranked in top 1 or 2 for overall quality.

based on their overall quality.

We randomly sampled 43 threads from the test set, and hired three native or proficient English speakers to evaluate arguments generated by SEQ2SEQAUG, our prior argument generation model (H&W), and the new model CANDELA, along with gold-standard HUMAN arguments and the top passage by RETRIEVAL.

Results. The first 3 examples are used only for calibration, and the remaining 40 are used to report results in Table 6.5. Inter-annotator agreement scores (Krippendorff’s α) of 0.44, 0.58, 0.49 are achieved for the three aspects, implying general consensus to intermediate agreement.

Our system obtains the highest appropriateness and content richness among all automatic systems. This confirms the previous observation that our model produces more informative argument than other neural models. SEQ2SEQAUG has a marginally better grammaticality score, likely due to the fact that our arguments are longer, and tend to contain less fluent generation towards the end.

Furthermore, we see that human arguments are ranked as the best in about 76% of the evaluation, followed by RETRIEVAL. Our model is more likely to be ranked top than any other automatic models. Especially, our model is rated better than either HUMAN or RETRIEVAL, i.e., human-edited text, in 39.2% of the evaluations, compared to 34.2% for SEQ2SEQAUG and 13.3% for our prior model.

6.6.3 Sample Arguments and Discussions

We show sample outputs of different systems alongside human constructed counter-argument in Figure 6.5. As can be seen, our system arguments cover many relevant talking points, including the idea of “*taking care of children*” that is also used by human. It further illustrates the effectiveness of our retrieval system and the usage of keyphrase selection as part of text planning to guide argument generation. Moreover, we also observe that our model generation is able to switch between argumentative content sentence and filler sentence, though better control is needed to improve coherence. Meanwhile, SEQ2SEQ frequently echos words from OP, and both SEQ2SEQ and SEQ2SEQAUG suffer from the problems of “hallucination” (e.g., the first sentence in SEQ2SEQAUG) and repetition (e.g., the second and third sentences in SEQ2SEQ).

Nonetheless, there is a huge space for improvement. First, our model tends to overuse negation, such as “*this is not to say that it is unfair...*”. It is likely due to its overfitting on specific stylistic languages, e.g., negation is often observed for refutation in debates (Wang et al., 2017b). Second, human arguments have significantly better organization and often deploy complicated argumen-

<p>OP: The wage gap isn't about gender. (...) So one factor definitely the fact that there are some government jobs that are seriously dominated by women and happen to pay really low (...)</p>
<p>Human: Children are the first factor. I work in a traditionally male-dominated field, and questions are asked to determine familial status, even though those are illegal. Employers know single moms have to stay home if their child is sick. They know single moms can only work during daycare hours. They know single moms are unavailable for weekend and late night work. They know single moms cannot travel. The workplace is very family-unfriendly.</p>
<p>Seq2seq: the problem with this is that there are a lot of people who do n't want to hire women . there are plenty of women who do n't have the money to pay for it . the problem is that women do n't need to pay for their work . they do n't have to worry about it . it 's a matter of money .</p>
<p>Seq2seqAug: men and women are not 39.5 % of the pay gap . the problem is not that women are more likely to be victims of sexism , but rather that they are more natural good-looking/attractive action . this is not the case .</p>
<p>CANDELA: the problem with this argument is that the wage gap does not have to do with the gender pay gap . it is a fact that women are more likely to be able to take care of their children than their male counterparts . this is not to say that it is unfair to assume that women are being paid less than men , but that does not mean that it is not the case that women are discriminated against . <i>it is not a matter of the wage gap , it is a matter of opinion .</i> it is the job of the employer to make sure that the job is not the same as the other</p> <p>Keyphrase Memory: wage gap; discrimination; gender pay gaps; raise the child; male colleagues; paid maternity leave; underlying gender discrimination ...</p>

Figure 6.5: Sample arguments generated by different systems along with a sample human argument. For our model CANDELA, additionally shown are the keyphrase memory with selected phrases in color, and argumentative filler sentence in italics.

tation strategies (Wachsmuth et al., 2018a), which so far is not well captured by any automatic system. Both points inspire future work on (1) controlling of the language styles and corresponding content, and (2) mining argumentation structures for use in guiding generation with better planning.

6.7 Conclusion

We present a novel counter-argument generation framework, CANDELA. Given an input statement, it first retrieves arguments of different perspectives from millions of high-quality articles collected from diverse sources. An argument generation component then employs a text planning decoder to conduct content selection and specify a suitable language style at sentence-level, followed by a content realization decoder to produce the final argument. Automatic evaluation and human evaluation indicate that our model generates more proper arguments with richer content than non-trivial comparisons, with comparable fluency to human-edited content.

CHAPTER 7

SENTENCE-LEVEL CONTENT PLANNING AND STYLE SPECIFICATION FOR NEURAL TEXT GENERATION

In this chapter, we propose a generalized text generation framework enabled with content selection. We showcase this model over two new domains on Wikipedia paragraph and academic paper abstract generation.

This work is published at EMNLP 2019 (Hua and Wang, 2019). Relevant resources can be found at: xinyuhua.github.io/Resources/emnlp19/.

7.1 Introduction

Automatic text generation is a long-standing challenging task, as it needs to solve at least three major problems: (1) *content selection* (“what to say”), identifying pertinent information to present, (2) *text planning* (“when to say what”), arranging content into ordered sentences, and (3) *surface realization* (“how to say it”), deciding words and syntactic structures that deliver a coherent output based on given discourse goals (McKeown, 1985). Traditional text generation systems often handle each component separately, thus requiring extensive effort on data acquisition and system engineering (Reiter and Dale, 2000). Recent progress has been made by developing end-to-end trained neural models (Rush et al., 2015; Yu et al., 2018; Fan et al., 2018b), which naturally excel at producing fluent text. Nonetheless, limitations of model structures and training objectives make them suffer from low interpretability and substandard generations which are often incoherent and unfaithful to the input material (See et al., 2017; Wise-

Topic: US should cut off foreign aid completely.

/r/ChangeMyView *Counter-argument:* It can be a useful **political bargaining chip**. A few years ago, the US **cut financial aid** to **Uganda** due to its plans to **make homosexuality a crime** punishable by death. *Please consider changing your mind!*

Topic: Artificial Intelligence

English Wikipedia: ... **Computer science** defines **AI research** as ... any **device** that **perceives its environment** and **takes actions** that **maximize its chance** of successfully **achieving its goals**. ...

Simple Wikipedia: **Artificial Intelligence** is the **ability** of a **computer program** or a **machine** to **think and learn**. ...

Figure 7.1: [Upper] Sample counter-argument from Reddit. Argumentative stylistic language for persuasion is in italics. [Bottom] Excerpts from Wikipedia, where sophisticated concepts and language of higher complexity used in the standard version are not present in the corresponding simplified version. Both: key concepts are in bold.

man et al., 2017; Li et al., 2017).

To address the problems, we believe it is imperative for neural models to gain adequate control on content planning (i.e., content selection and ordering) to produce coherent output, especially for long text generation. We further argue that, in order to achieve desired discourse goals, it is beneficial to enable style-controlled surface realization by explicitly modeling and specifying proper linguistic styles. Consider the task of producing counter-arguments to the topic “US should cut off foreign aid completely”. A sample argument in Figure 7.1 demonstrates how human selects a series of talking points and a proper style based on the argumentative function for each sentence. For instance, the argument starts with a *proposition* on “foreign aid as a political bargaining chip”, followed by a concrete *example* covering several key concepts. It ends with *argumentative stylistic language*, which differs in both content and style from the previous sentences. Figure 7.1 shows another example on Wikipedia articles:

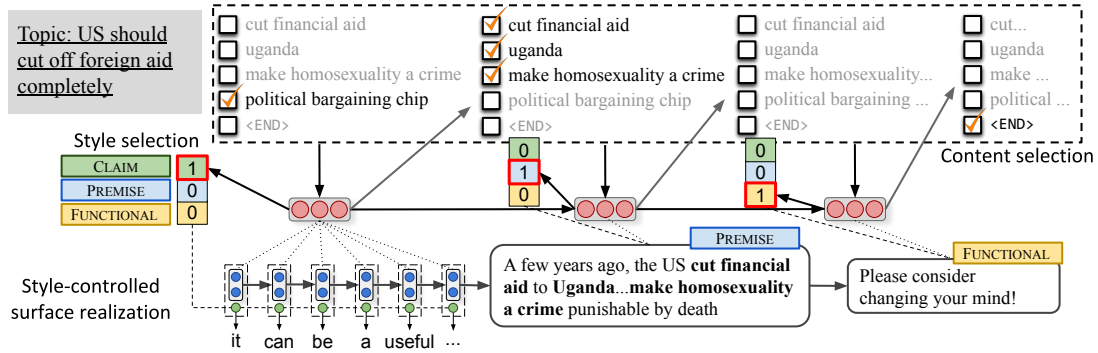


Figure 7.2: Overview of our framework. The LSTM content planning decoder (§ 7.2.2) first identifies a set of keyphrases from the memory bank conditional on previous selection history, based on which, a style is specified. During surface realization, the hidden states of the planning decoder and the predicted style encoding are fed into the realizer, which generates the final output (§ 7.2.3). Best viewed in color.

compared to a topic’s standard version where longer sentences with complicated concepts are constructed, its simplified counterpart tends to explain the same subject with plain language and simpler concepts, indicating the interplay between content selection and language style.

We thus present an end-to-end trained neural text generation framework that includes the modeling of traditional generation components, to promote the control of content and linguistic style of the produced text. Our model performs *sentence-level content planning* for information selection and ordering, and *style-controlled surface realization* to produce the final generation. We focus on conditional text generation problems (Lebret et al., 2016; Colin et al., 2016; Dušek et al., 2018): As shown in Figure 7.2, the input to our model consists of a topic statement and a set of keyphrases. The output is a relevant and coherent paragraph to reflect the salient points from the input. We utilize two separate decoders: for each sentence, (1) a planning decoder selects relevant keyphrases and a desired style conditional on previous selections, and (2) a realization decoder produces the text in the specified style.

We demonstrate the effectiveness of our framework on three challenging datasets with diverse topics and varying linguistic styles: persuasive argument generation on Reddit ChangeMyView (Hua and Wang, 2018); introduction paragraph generation on a *newly* collected dataset from Wikipedia and its simple version; and scientific paper abstract generation on AGENDA dataset (Koncel-Kedziorski et al., 2019).

Experimental results on all three datasets show that our models that consider content planning and style selection achieve significantly better BLEU, ROUGE, and METEOR scores than non-trivial comparisons that do not consider such information. Human judges also rate our model generations as more fluent and correct compared to the outputs produced by its variants without style modeling.

7.2 Model

Our model tackles conditional text generation tasks where the input is comprised of two major parts: (1) a topic statement, $\mathbf{x} = \{x_i\}$, which can be an argument, the title of a Wikipedia article, or a scientific paper title, and (2) a keyphrase memory bank, \mathcal{M} , containing a list of talking points, which plays a critical role in content planning and style selection. We aim to produce a sequence of words, $\mathbf{y} = \{y_i\}$, to comprise the output, which can be a counter-argument, a paragraph as in Wikipedia articles, or a paper abstract.

7.2.1 Input Encoding

The input text \mathbf{x} is encoded via a bidirectional LSTM (biLSTM), with its last hidden state used as the initial states for both content planning decoder and surface realization decoder. To encode keyphrases in the memory bank \mathcal{M} , each keyphrase is first converted into a vector \mathbf{e}_k by summing up all its words' embeddings from GloVe (Pennington et al., 2014). A biLSTM-based *keyphrase reader*, with hidden states \mathbf{h}_k^e , is used to encode all keyphrases in \mathcal{M} . We also insert entries of $\langle \text{START} \rangle$ and $\langle \text{END} \rangle$ into \mathcal{M} to facilitate learning to start and finish selection.

7.2.2 Sentence-Level Content Planning and Style Specification

Content Planning: Context-Aware Keyphrase Selection. Our content planner selects a set of keyphrases from the memory bank \mathcal{M} for each sentence, indexed with j , conditional on keyphrases that have been selected in previous sentences, allowing topical coherence and content repetition avoidance. The decisions are denoted as a *selection vector* $\mathbf{v}_j \in \mathbb{R}^{|\mathcal{M}|}$, with each dimension $v_{j,k} \in \{0, 1\}$, indicating whether the k -th phrase is selected for the j -th sentence generation. Starting with a $\langle \text{START} \rangle$ tag as the input for the first step, our planner predicts \mathbf{v}_1 for the first sentence, and recurrently makes predictions per sentence until $\langle \text{END} \rangle$ is selected, as depicted in Figure 7.2.

Formally, we utilize a sentence-level LSTM f , which consumes the summation embedding of selected keyphrases, \mathbf{m}_j , to produce a hidden state \mathbf{s}_j for the

j -th sentence step:

$$\mathbf{s}_j = f(\mathbf{s}_{j-1}, \mathbf{m}_j) \quad (7.1)$$

$$\mathbf{m}_j = \sum_{k=1}^{|\mathcal{M}|} \mathbf{v}_{j,k} \mathbf{h}_k^e \quad (7.2)$$

where $\mathbf{v}_{j,k} \in \{0, 1\}$ is the selection decision for the k -th keyphrase in the j -th sentence.

Our prior work (Hua et al., 2019a) (Chapter 6) utilizes a similar formulation for sentence representations. However, the prediction of \mathbf{v}_{j+1} is estimated by a bilinear product between \mathbf{h}_k^e and \mathbf{s}_j , which is agnostic to what have been selected so far. While in reality, content selection for a new sentence should depend on previous selections. For instance, keyphrases that have already been utilized many times are less likely to be picked again; topically related concepts tend to be mentioned closely. We therefore propose a vector \mathbf{q}_j that keeps track of what keyphrases have been selected up to the j -th sentence:

$$\mathbf{q}_j = \left(\sum_{r=0}^j \mathbf{v}_r \right)^T \times \mathbb{E} \quad (7.3)$$

where $\mathbb{E} = [\mathbf{h}_1^e, \mathbf{h}_2^e, \dots, \mathbf{h}_{|\mathcal{M}|}^e]^T \in \mathbb{R}^{|\mathcal{M}| \times H}$ is the matrix of keyphrase representations, H is the hidden dimension of the keyphrase reader LSTM.

Then \mathbf{v}_{j+1} is calculated in an attentive manner with \mathbf{q}_j as the attention query:

$$P(\mathbf{v}_{j+1,k} = 1 | \mathbf{v}_{1:j}) = \sigma(\mathbf{w}_v^T \mathbf{s}_j + \mathbf{q}_j \mathbf{W}^c \mathbf{h}_k^e) \quad (7.4)$$

where σ is the sigmoid function, and \mathbf{w}_* , \mathbf{W}^* , and \mathbf{W}^{**} are trainable parameters throughout the paper. Bias terms are all omitted for simplicity.

As part of the learning objective, we utilize the binary cross-entropy loss

with the gold-standard selection \mathbf{v}_j^* as criterion over the training set D :

$$\mathcal{L}_{\text{sel}} = - \sum_{(\mathbf{x}, \mathbf{y}) \in D} \sum_{j=1}^J \left(\sum_{k=1}^{|\mathcal{M}|} \log(P(\mathbf{v}_{j,k}^*)) \right) \quad (7.5)$$

Style Specification. As discussed in § 7.1, depending on the content (represented as selected keyphrases in our model), humans often choose different language styles adapted for different discourse goals. Our model characterizes such stylistic variations by assigning a categorical style type \mathbf{t}_j for each sentence, which is predicted as follows:

$$\hat{\mathbf{t}}_j = \text{softmax}(\mathbf{w}_s^T (\tanh(\mathbf{W}^s [\mathbf{m}_j; \mathbf{s}_j]))) \quad (7.6)$$

$\hat{\mathbf{t}}_j$ is the estimated distribution over all types. We select the one with the highest probability and use a one-hot encoding vector, \mathbf{t}_j , as the input to our realization decoder (§ 7.2.3). The estimated distributions $\hat{\mathbf{t}}_j$ are compared against the gold-standard labels \mathbf{t}_j^* to calculate the cross-entropy loss $\mathcal{L}_{\text{style}}$:

$$\mathcal{L}_{\text{style}} = - \sum_{(\mathbf{x}, \mathbf{y}) \in D} \sum_{j=1}^J \mathbf{t}_j^* \log \hat{\mathbf{t}}_j \quad (7.7)$$

7.2.3 Style-Controlled Surface Realization

Our surface realization decoder is implemented with an LSTM with state calculation function g to get each hidden state \mathbf{z}_t for the t -th generated token. To compute \mathbf{z}_t , we incorporate the content planning decoder hidden state $\mathbf{s}_{J(t)}$ for the sentence to be generated, with $J(t)$ as the sentence index, and previously generated token \mathbf{y}_{t-1} :

$$\mathbf{z}_t = g(\mathbf{z}_{t-1}, \tanh(\mathbf{W}^{ws} \mathbf{s}_{J(t)} + \mathbf{W}^{ww} \mathbf{y}_{t-1})) \quad (7.8)$$

For word prediction, we calculate two attentions, one over the input statement \mathbf{x} , which produces a context vector \mathbf{c}_t^w (Eq. 7.10), the other over the keyphrase memory bank \mathcal{M} , which generates \mathbf{c}_t^e (Eq. 7.11). To better reflect the control over word choice by language styles, we directly append the predicted style $\mathbf{t}_{J(t)}$ to the context vectors and hidden state \mathbf{z}_t , to compute the distribution over the vocabulary¹:

$$P(y_t|y_{1:t-1}) = \text{softmax}(\tanh(\mathbf{W}^o[\mathbf{z}_t; \mathbf{c}_t^w; \mathbf{c}_t^e; \mathbf{t}_{J(t)}])) \quad (7.9)$$

$$\mathbf{c}_t^w = \sum_{i=1}^L \alpha_i^w \mathbf{h}_i, \quad \alpha_i^w = \text{softmax}(\mathbf{z}_t \mathbf{W}^{wa} \mathbf{h}_i) \quad (7.10)$$

$$\mathbf{c}_t^e = \sum_{k=1}^{|\mathcal{M}|} \alpha_k \mathbf{h}_k^e, \quad \alpha_k = \text{softmax}(\mathbf{z}_t \mathbf{W}^{we} \mathbf{h}_k^e) \quad (7.11)$$

We further adopt a copying mechanism from See et al. (2017) to enable direct reuse of words from the input \mathbf{x} and keyphrase bank \mathcal{M} to allow out-of-vocabulary words to be included.

7.2.4 Training Objective

We jointly learn to conduct content planning and surface realization by aggregating the losses over (i) word generation: $\mathcal{L}_{\text{gen}} = -\sum_D \sum_{t=1}^T \log P(y_t^*|\mathbf{x}; \theta)$, (ii) keyphrase selection: \mathcal{L}_{sel} (Eq. 7.5), and (iii) style prediction $\mathcal{L}_{\text{style}}$ (Eq. 7.7):

$$\mathcal{L}(\theta) = \mathcal{L}_{\text{gen}}(\theta) + \gamma \cdot \mathcal{L}_{\text{style}}(\theta) + \eta \cdot \mathcal{L}_{\text{sel}}(\theta) \quad (7.12)$$

where θ denotes the trainable parameters. γ and η are set to 1.0 in our experiments for simplicity.

¹The inclusion of style variables is different from our prior style-aware generation model (Hua et al., 2019a), where styles are predicted but not encoded for word production.

7.3 Tasks and Datasets

7.3.1 Task I: Argument Generation

Our first task is to generate a counter-argument for a given statement on a controversial issue. The input keyphrases are extracted from automatically retrieved and reranked passages with queries constructed from the input statement.

We reuse the dataset from our previous work (Hua et al., 2019a) as described in Chapter 6, but annotate with newly designed style scheme. We first briefly summarize the procedures for data collection, keyphrase extraction and selection, and passage reranking; more details can be found in our prior work. Then we describe how to label argument sentences with style types that capture argumentative structures.

The dataset is collected from Reddit /r/ChangeMyView subcommunity, where each thread consists of a multi-paragraph original post (OP), followed by user replies with the intention to change the opinion of the OP user. Each OP is considered as the input, and the root replies awarded with delta (Δ), or with positive karma ($\# \text{ upvotes} > \# \text{ downvotes}$) are target counter-arguments to be generated. A domain classifier is further adopted to select politics related threads.

Input Keyphrases and Label Construction. To obtain the input keyphrase candidates and their sentence-level selection labels, we first construct queries to retrieve passages from Wikipedia and news articles collected from

	Argument # Args (# Threads)	Wikipedia (Nor. / Sim.)	AGENDA
# Train	272,147 (11,434)	125,136	38,720
# Dev	40,291 (1,784)	21,004	1,000
# Test	46,757 (1,706)	23,534	1,000
# Tokens	54.87	70.57 / 48.60	141.34
# Sent.	2.48	3.15 / 3.20	5.59
# KP (candidates)	55.80	23.56	12.23
# KP (selected)	11.61	16.01/11.11	12.23

Table 7.1: Statistics of the three datasets. Average numbers are reported. For argument dataset, number of unique threads is also shown. On AGENDA, entities are extracted from abstract as keyphrases, hence all candidates are “selected”.

commoncrawl.org.² For training, we construct a query per target argument sentence using its content words for retrieval, and keep top 5 passages per query. For testing, the queries are constructed from the sentences in OP (input statement).

We then extract keyphrases from the retrieved passages based on topic signature words (Lin and Hovy, 2000a) calculated over the given OP. These words, together with their related terms from WordNet (Miller, 1994), are used to determine whether a phrase in the passage is a *keyphrase*. Specifically, a keyphrase is (1) a noun phrase or verb phrase that is shorter than 10 tokens; (2) contains at least one content word; (3) has a topic signature or a Wikipedia title. For each keyphrase candidate, we match them with the sentences in the target counter-argument, and we consider it to be “selected” for the sentence if there is any overlapping content word.

During test time, we further adopt a stance classifier from Bar-Haim et al. (2017) to produce a stance score for each passage. We retain passages that have

²The choice of news portals, statistics of the dataset, and preprocessing steps are described in § 6.3.1

	CLAIM	PREMISE	FUNCTIONAL	
# Arguments	29.1%	62.2%	8.7%	
# Tokens	17.0	26.2	10.0	
Length \in	(0, 10]	(10, 20]	(20, 30]	(30, + ∞)
Normal Wikipedia	9.9%	40.5%	29.8%	19.8%
Simple Wikipedia	29.3%	51.7%	14.6%	4.4%

Table 7.2: Sentence style distribution for argument and Wikipedia datasets.

a negative stance towards OP, and a greater than 5 stance score. They are further ordered based on the number of overlapping keyphrases with the OP. Top 10 passages are used to construct the input keyphrase bank, and as optional input to our model.

Sentence Style Label Construction. For argument generation, we define three sentence styles based on their argumentative discourse functions (Persing and Ng, 2016; Lippi and Torroni, 2016): CLAIM is a proposition, usually containing one or two talking points, e.g., *“I believe foreign aid is a useful bargaining chip”*; PREMISE contains supporting arguments with reasoning or examples; FUNCTIONAL is usually a generic statement, e.g., *“I understand what you said”*. For training, we employ a list of rules extended from the claim detection method by Levy et al. (2018) to automatically construct a style label for each sentence. Statistics are displayed in Table 7.2, and the rules are shown below:

- CLAIM: is shorter than 20 tokens and matches any pattern in Table 7.3.
- PREMISE: is longer than 5 tokens, contains at least one noun or verb content word, and matches any pattern in Table 7.3.
- FUNCTIONAL: contains fewer than 5 alphabetical words and no noun or verb content word

Rule	Patterns
CLAIM	
Belief	i (don't)? (believe agree concede suspect doubt see feel understand)
Imperative	(any anyone anybody every everyone everybody all most few no no one nobody it we you they there) \w{0,10} (could should might need must)
Sense	(it this that) make (no zero)? sense
Chance	(chance likelihood possibility probability) .* (slim zero negligible)
Evaluation	(be seem) (necessary unnecessary moral immoral right wrong stupid unconstitutional costly inefficient efficient reasonable beneficial important unfair harmful justified jeopardized meaningless flawed justifiable unacceptable impossible irrational foolish)
Miscellaneous	(in my opinion imo my view i be try to say have nothing to do with tldr)
PREMISE	
Affect	(help improve reduce deter increase decrease promote)
Example	(for example for instance e.g.)

Table 7.3: Patterns for sentence style label construction on CLAIM and PREMISE for argument generation.

Paragraphs that only contain FUNCTIONAL sentences are removed from our dataset.

7.3.2 Task II: Paragraph Generation for Normal and Simple Wikipedia

The second task is generating introduction paragraphs for Wikipedia articles. The input consists of a title, a user-specified global style (normal or simple), and a list of keyphrases collected from the gold-standard paragraphs of both normal and simple Wikipedia. During training and testing, the global style is encoded as one extra bit appended to m_j (Eq. 7.2).

We construct a *new* dataset with topically-aligned paragraphs from normal and simple English Wikipedia.³ For alignment, we consider it a match if two articles share exactly the same title with at most two non-English words. We then extract the first paragraphs from both and filter out the pair if one of the paragraphs is shorter than 10 words or is followed by a table.

Input Keyphrases and Label Construction. Similar to argument generation, we extract noun phrases and verb phrases and consider the ones with at least one content word as keyphrase candidates. After de-duplication, there are on average 5.4 and 3.7 keyphrases per sentence for the normal and simple Wikipedia paragraphs, respectively. For each sample, we merge the keyphrases from the aligned paragraphs as the input. The model is then trained to select the appropriate ones conditioned on the global style.

Sentence Style Label Construction. We distinguish sentence-level styles based on language complexity, which is approximated by sentence length. The distribution of sentence styles is displayed in Table 7.2.

7.3.3 Task III: Paper Abstract Generation

We further consider a task of generating abstracts for scientific papers (Ammar et al., 2018), where the input contains a paper title and scientific entities mentioned in the abstract. We use the AGENDA data processed by Koncel-Kedziorski et al. (2019), where entities and their relations in the abstracts are extracted by SciIE (Luan et al., 2018). All entities appearing in the abstract are included in our keyphrase bank. The state-of-the-art system (Koncel-Kedziorski

³We download the dumps of 2019/04/01 for both dataset.

et al., 2019) exploits the scientific entities, their relations, and the relation types. In our setup, we ignore the relation graph, and focus on generating the abstract with only entities and title as the input. Due to the dataset’s relatively uniform language style and smaller size, we do not experiment with our style specification component.

7.4 Experiments

7.4.1 Implementation Details

For argument generation, we truncate the input OP and retrieved passages to 500 and 400 words. Passages are optionally appended to OP as our encoder input. The keyphrase bank size is limited to 70 for argument, and 30 for Wikipedia and AGENDA data (based on the average numbers in Table 7.1), with keyphrases truncated to 10 words. We use a vocabulary size of 50K for all tasks.

Training Details. Our models use a two-layer LSTM for both decoders. They all have 512-dimensional hidden states per layer and dropout probabilities (Gal and Ghahramani, 2016) of 0.2 between layers. Wikipedia titles are encoded with the summation of word embeddings due to their short length. The learning process is driven by AdaGrad (Duchi et al., 2011) with 0.15 as the learning rate and 0.1 as the initial accumulator. We clip the gradient norm to a maximum of 2.0. The mini-batch size is set to 64. And the optimal weights are chosen based on the validation loss.

For argument generation, we also pre-train the encoder and the lower layer

of realization decoder using language model losses. We collect all the OP posts from the training set, and an extended set of reply paragraphs, which includes additional counter-arguments that have non-negative `karma`. For Wikipedia, we consider the large collection of 1.9 million unpaired normal English Wikipedia paragraphs to pre-train the model for both normal and simple Wikipedia generation.

Beam Search Decoding. For inference, we utilize beam search with a beam size of 5. We disallow the repetition of trigrams, and replace the `UNK` with the keyphrase of the highest attention score.

7.4.2 Baselines and Comparisons

For all three tasks, we consider a SEQ2SEQ with attention baseline (Bahdanau et al., 2015), which encodes the input text and keyphrase bank as a sequence of tokens, and generates the output.

For argument generation, we implement a RETRIEVAL baseline, which returns the highest reranked passage retrieved with OP as the query. We also compare with our prior model (Hua and Wang, 2018) (Chapter 5), which is a multi-task learning framework to generate both keyphrases and arguments.

For Wikipedia generation, a RETRIEVAL baseline obtains the most similar paragraph from the training set with input title and keyphrases as the query, measured with bigram cosine similarity. We further train a logistic regression model (LOGREGSEL), which takes the summation of word embeddings in a phrase and predicts its inclusion in the output for a normal or simple Wiki para-

graph.

For abstract generation, we compare with the state-of-the-art system GRAPHWRITER (Koncel-Kedziorski et al., 2019), which is a transformer model enabled with knowledge graph encoding mechanism to handle both the entities and their structural relations from the input.

We also report results by our model variants to demonstrate the usefulness of content planning and style control: (1) with gold-standard⁴ keyphrase selection for each sentence (Oracle Plan.), and (2) without style specification.

7.5 Results and Analysis

7.5.1 Automatic Evaluation

We report precision-oriented BLEU (Papineni et al., 2002), recall-oriented ROUGE-L (Lin, 2004) that measures the longest common subsequence, and METEOR (Denkowski and Lavie, 2014), which considers both precision and recall.

Argument Generation. For each input OP, there can be multiple possible counter-arguments. We thus consider the best matched (i.e., highest scored) reference when reporting results in Table 7.4. Our models yield significantly higher BLEU and ROUGE scores than all comparisons while producing longer arguments than generation-based approaches. Furthermore, among our model variants, oracle content planning further improves the performance, indicating

⁴“Gold-standard” indicates the keyphrases that have content word overlap with the reference sentence.

	BLEU-2	ROUGE-L	METEOR	Length
RETRIEVAL	7.81	15.68	10.59	150.0
SEQ2SEQ	3.64	19.00	9.85	51.7
H&W (2018)	5.73	14.44	3.82	36.5
Ours (Oracle Plan.)	16.30*	20.25*	11.61	65.5
Ours	13.19*	20.15*	10.42	65.2
w/o Style	12.61*	20.28*	10.15	64.5
w/o Passage	11.84*	19.90*	9.03	62.6

Table 7.4: Results on argument generation with BLEU (up to bigrams), ROUGE-L, and METEOR (MTR). Best systems without oracle planning are in **bold** per metric. Our models that are significantly better than all comparisons are marked with * ($p < 0.001$, approximate randomization test (Noreen, 1989)).

the importance of content selection and ordering. Taking out style specification decreases scores, indicating the influence of style control on generation.⁵

Wikipedia Generation. Results on Wikipedia (Table 7.5) show similar trends, where our models almost always outperform all comparisons across metrics. The significant performance drop on ablated models without style prediction proves the effectiveness of style usage. Our model, if guided with oracle keyphrase selection per sentence, again achieves the best performance.

We further show the *effect of content selection on generation* on Wikipedia and abstract data in Figure 7.3, where we group the test samples into 10 bins based on F1 scores on keyphrase selection.⁶ We observe a strong correlation between keyphrase selection and generation performance, e.g., for BLEU, Pearson correlations of 0.95 ($p < 10^{-4}$) and 0.85 ($p < 10^{-2}$) are established for Wikipedia and

⁵We do not compare with our recent model in Hua et al. (2019a) due to the training data difference caused by our new sentence style scheme. However, the newly proposed model generates arguments with lengths closer to human arguments, benefiting from the improved content planning module.

⁶We calculate F1 by aggregating the selections across all sentences. For argument generation, keyphrases are often paraphrased, making it difficult to calculate F1 reliably, therefore omitted here.

	BLEU	ROUGE	METEOR	Length	BLEU	ROUGE	METEOR	Length
	Normal Wikipedia				Simple Wikipedia			
RETRIEVAL	20.10	28.60	12.23	44.5	21.99	33.44	12.97	34.7
SEQ2SEQ	22.62	27.49	14.74	52.9	21.98	29.36	16.94	52.8
LOGREGSEL	29.28	28.65	27.76	34.3	5.59	23.21	13.27	13.0
OURS (Oracle Plan.)	37.70*	45.41*	31.65*	79.8	34.22*	45.48*	32.84*	70.5
OURS	33.76*	40.08*	25.70	65.4	31.22*	40.76*	26.76*	58.7
w/o Style	31.06*	37.72*	24.56	71.0	27.94*	38.20*	25.87*	64.5

Table 7.5: Results on Wikipedia generation. Best results without oracle planning are in **bold**. *: Our models that are significantly better than all comparisons ($p < 0.001$, approximate randomization test).

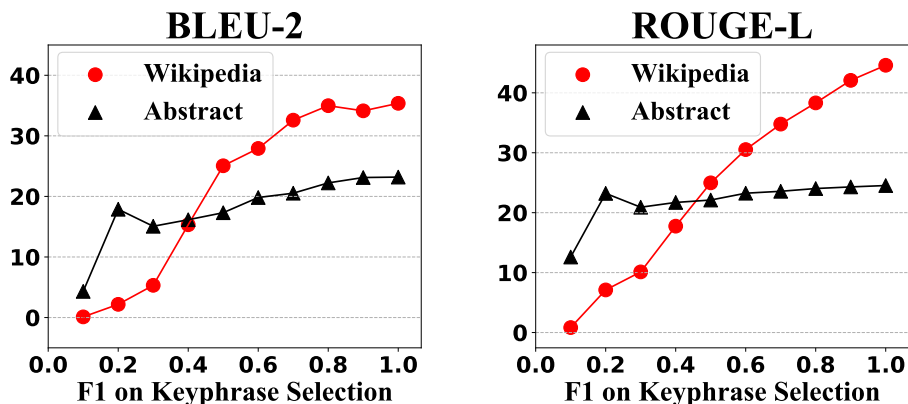


Figure 7.3: Effect of keyphrase selection (F1 score) on generation performance, measured by BLEU and ROUGE. Positive correlations are observed.

abstract. For ROUGE, the values are 0.99 ($p < 10^{-8}$) and 0.72 ($p < 10^{-1}$).

Abstract Generation. Lastly, we compare with the state-of-the-art GRAPHWRITER model on AGENDA dataset in Table 7.6. Although our model does not make use of the relational graph encoding, we achieve competitive ROUGE-L and METEOR scores given the oracle plans. Our model also outperforms the seq2seq baseline, which has the same input, indicating the applicability of our method across different domains.

	BLEU-2	ROUGE-L	METEOR	Length
GRAPHWRITER	29.95	28.56	19.90	130.1
SEQ2SEQ	18.13	21.03	13.95	134.8
Ours (Oracle Plan.)	25.03	26.18	19.21	125.8
Ours	20.32	23.30	15.95	128.3

Table 7.6: Results on paper abstract generation. Notice that GRAPHWRITER models rich information about relations and relation types among entities, which is not utilized by our model.

	Argument			Wikipedia		
	Gram.	Corr.	Cont.	Gram.	Corr.	Cont.
HUMAN	4.81	3.90	3.48	4.84	4.73	4.49
Ours	3.99*	2.78*	2.61*	3.38	3.24*	3.43
w/o Style	3.03	2.26	2.03	2.99	2.89	3.50
<i>Krippendorff's α</i>	0.75	0.69	0.33	0.70	0.56	0.55

Table 7.7: Human evaluation on argument generation (Upper) and Wikipedia generation (Bottom). Grammaticality (**Gram**), correctness (**Corr**), and content richness (**Cont**) are rated on Likert scale (1 – 5). We mark our model with * to indicate statistically significantly better ratings over the variant without style specification ($p < 0.001$, approximate randomization test).

7.5.2 Human Evaluation

We further ask three proficient English speakers to assess the quality of generated arguments and Wikipedia paragraphs. Human subjects are asked to rate on a scale of 1 (worst) to 5 (best) on **grammaticality**, **correctness** of the text (for arguments, the stance is also considered), and **content richness** (i.e., coverage of relevant points). Detailed guidelines for different ratings are provided to the raters (Table 7.9 and Table 7.10). For both tasks, we randomly choose 30 samples from the test set; outputs from two variants of our models and a human written text are presented in random order.

According to Krippendorff’s α , the raters achieve substantial agreement on

<p>Topic: Aborting a fetus has some non-zero negative moral implications</p> <p>Human: It's not the birthing process that changes things. It's the existence of the baby. Before birth, the baby only exists inside another human being. After birth, it exists on its own in the world like every other person in the world.</p> <p>Seq2seq: i 'm not going to try to change your view here , but i do n't want to change your position . i do n't think it 's fair to say that a fetus is not a person . it 's not a matter of consciousness .</p> <p>Our model: tl ; dr : i agree with you , but i think it 's important to note that fetuses are not fully developed . i do n't know if this is the case , but it does n't seem to be a compelling argument to me at all , so i 'm not going to try to change your view by saying that it should be illegal to kill</p>
<p>Topic: Moon Jae-in</p> <p>Simple Wikipedia: Moon Jae-in is a South Korean politician. He is the 12th and current President of South Korea since 10 May 2017 after winning the majority vote in the 2017 presidential election.</p> <p>Seq2seq: moon election park is a election politician who served as prime minister of korea from 2007 to 2013 . he was elected as a member of the house of democratic party in the moon 's the the moon the first serving president of jae-in , in office since 2010 .</p> <p>Our model: moon jae-in is a south korean politician and current president of south korea from 2012 to 2017 and again from 2014 to 2017.</p> <p>Normal Wikipedia: Moon Jae-in is a South Korean politician serving as the 19th and current President of South Korea since 2017. He was elected after the impeachment of Park Geun-hye as the candidate of the Democratic Party of Korea.</p> <p>Seq2seq: moon winning current is a current politician who served as prime minister of korea from 2007 to 2013 . he was elected as a member of the house of democratic party in the moon 's the the current the first president of pakistan , in office . prior to that , he also served on the democratic republic of germany .</p> <p>Our model: moon jae-in is a south korean politician serving as the 19th and current president of south korea , since 2019 to 2019 and 2019 to 2017 respectively he has been its current president ever since .</p>

Figure 7.4: Sample outputs for argument generation and Wikipedia generation.

grammaticality and correctness, while the agreement on content richness is only moderate due to its subjectivity. As shown in Table 7.7, on both tasks, our models with style specification produce more fluent and correct generations, compared to the ones without such information. However, there is still a gap between system generations and human edited text.

We further show **sample outputs** in Figure 7.4. The first example is on the

	Human reference	Our model
CLAIM	It doesn't mean that; every-	I don't believe that it is neces-
PREMISE	one should be able to have the freedom to; is leagal in the US; imagine for a mo- ment if; Let's say (you/your partner/a friend)	sary; don't need to be able to have the right to (bear arms/cast a ballot vote); For example, (if you look at/let's look at/I don't think)
FUNCTIONAL	Why is that?; that's ok; Would it change your mind?	I'm not sure (why/if) this is; TLDR: I don't care about this

Table 7.8: Top frequent patterns captured in style CLAIM, PREMISE, and FUNCTIONAL from arguments by human and our model.

topic of abortion, our model captures the relevant concepts such as “*fetuses are not fully developed*” and “*illegal to kill*”. It also contains fewer repetitions than the seq2seq baseline. For Wikipedia, our model is not only better at controlling the global simplicity style, but also more grammatical and coherent than the seq2seq output.

7.5.3 Further Analysis and Discussions

We further investigate the usage of different styles, and show the top frequent patterns for each argument style from human arguments and our system generation (Table 7.8). We first calculate the most frequent 4-grams per style, then extend it with context. We manually cluster and show the representative ones. For both columns, the popular patterns reflect the corresponding discourse functions: CLAIM is more evaluative, PREMISE lists out details, and FUNCTIONAL exhibits argumentative stylistic languages. Interestingly, our model also learns to paraphrase popular patterns, e.g., “*have the freedom to*” vs. “*have the right to*”.

For Wikipedia, the sentence style is defined by length. To validate its effect

on content selection, we calculate the average number of keyphrases per style type. The results on human written paragraphs are 2.0, 3.8, 5.8, and 9.0 from the simplest to the most complex. A similar trend is observed in our model outputs, which indicates the challenge of content selection in longer sentences.

For future work, improvements are needed in both model design and evaluation. As shown in Figure 7.4, system arguments appear to overfit on stylistic languages and rarely create novel concepts like humans do. Future work can lead to improved model guidance and training methods, such as reinforcement learning-based explorations, and better evaluation to capture diversity.

7.6 Conclusion

We present an end-to-end trained neural text generation model that considers sentence-level content planning and style specification to gain better control of text generation quality. Our content planner first identifies salient keyphrases and a proper language style for each sentence, then the realization decoder produces fluent text. We consider three tasks of different domains on persuasive argument generation, paragraph generation for normal and simple versions of Wikipedia, and abstract generation for scientific papers. Experimental results demonstrate the effectiveness of our model, where it obtains significantly better BLEU, ROUGE, and METEOR scores than non-trivial comparisons. Human subjects also rate our model generations as more grammatical and correct when language style is considered.

In the following survey, you will read 33 short argumentative text prompts and evaluate 3 counter-arguments for each of them. Please rate each counter-argument on a scale of 1-5 (the higher the better), based on the following three aspects:

- **Grammaticality:** whether the counter-argument is fluent and has no grammar errors
 - 1. *the way the way etc. 'm not 's important*
 - 3. *is a good example. i don't think should be the case. i're not going to talk whether or not it's bad.*
 - 5. *i agree that the problem lies in the fact that too many representatives do n't understand the issues or have money influencing their decisions.*
- **Correctness:** whether the counter-argument is relevant to the topic and of correct stance
 - 1. *i don't think it 's fair to say that people should n't be able to care for their children*
 - 3. *i don't agree with you and i think legislative bodies do need to explain why they vote that way*
 - 5. *there are hundreds of votes a year . how do you decide which ones are worth explaining ? so many votes are bipartisan if not nearly unanimous . do those all need explanations ? they only have two years right now and i do n't want them spending less time legislating .*
- **Content richness:** whether the counter-argument covers many talking points
 - 1. *i do n't agree with your point about legislation but i 'm not going to change your view.*
 - 3. *i agree that this is a problem for congress term because currently it is too short.*
 - 5. *congressional terms are too short and us house reps have to spend half of their time campaigning and securing campaign funds. they really have like a year worth of time to do policy and another year to meet with donors and do favors.*

Table 7.9: Evaluation guidelines on argument data and representative examples on rating scales.

In the following survey, you will read 32 samples. Each of them contains a topic, and 3 text snippets explaining the topic. For each of the explanation, please rate it on a scale of 1-5 (the higher the better) based on the following three aspects:

- **Grammaticality:** whether the explanation is fluent and has no grammar errors
 - 1. *android android operating system mobile kindle is a modified operating*
 - 3. *android is an operating system for mobile mobile mobile and other manufactures like htc and the*
 - 5. *android is an operating system for mobile devices . it is also used by other manufactures like htc and samsung .*

- **Correctness:** whether the explanation contains obvious semantic mistakes or contradictions. Please note that this is NOT intended for fact checking, so you should not find other resources to determine if the concrete information (such as years, locations) are wrong, instead please apply commonsense level knowledge to judge the correctness
 - 1. *android is used for tablets such as amazon.com as well as other phone such as linux and amazon*
 - 3. *android is an operating system for android and devices .*
 - 5. *android is an operating system for mobile devices .*

- **Content richness:** whether the explanation covers the amount of information that is necessary to explain the topic
 - 1. *modified mobile mobile android*
 - 3. *android is an operating system used for mobile devices .*
 - 5. *android is an operating system for mobile devices , it is mostly used for, like google 's own google pixel, as well as by other phone manufacturers like htc and samsung .*

Table 7.10: Evaluation guidelines on Wikipedia data and representative examples on rating scales.

CHAPTER 8

PLANNING AND ITERATIVE REFINEMENT IN PRE-TRAINED TRANSFORMERS FOR LONG TEXT GENERATION

In this chapter, we present a text generation framework using pre-trained Transformer. Our system is capable to generate longer output, and achieves better controllability over keyphrases.

This work is published at EMNLP 2020 (Hua and Wang, 2020). Relevant resources can be found at: <http://xinyuhua.github.io/Resources/emnlp20/>.

8.1 Introduction

Large pre-trained language models are the cornerstone of many state-of-the-art models in various natural language understanding and generation tasks (Devlin et al., 2019; Liu et al., 2019; Lewis et al., 2020a), yet they are far from perfect. In generation tasks, although models like GPT-2 (Radford et al., 2019) are able to produce plausible text, their spontaneous nature limits their utility in actual applications, e.g., users cannot specify what contents to include, and in what order.

To make large models more useful in practice, and to improve their generation quality, we believe it is critical to inform them of *when to say what*, which is addressed as *content planning* in traditional generation systems (Duboue and McKeown, 2001; Stent et al., 2004). Specially designed control codes and auxiliary planning modules have been integrated into neural models (Keskar et al.,

Prompt: CMV. Donald Trump is a communist.

Content Plan (output by planning model):
(1) **a communist**₃ ▷ **begin with**₈ ▷ **coherent ideology**₁₅ ▷
[SEN]₂₁
(2) [SEN]₄
(3) **no evidence**₂ ▷ **any coherent**₈ ▷ **held beliefs**₁₂ ▷ **any**
topic₁₅ ▷ [SEN]₁₈

I: *Template construction*

II: *Generation with content plan*

Template:

(1) ___0 ___1 ___2 **a communist** ___5 ___6 ___7 **begin with** ___10
___11 ___12 ___13 ___14 **coherent ideology** ___17 ___18 ___19 ___20
(2) ___0 ___1 ___2 ___3
(3) ___0 ___1 **no evidence** ___4 ___5 ___6 ___7 **any coherent** ___10
___11 **held beliefs** ___14 **any topic** ___17

Draft (initial generation):

(1) Well call him **a communist**, you must **begin with** that Donald Trump has some kind of **coherent ideology** to begin with.
(2) Which is unlikely.
(3) There is **no evidence** to suggest Donald Trump has **any coherent** or commonly **held beliefs** on **any topic**.

Refined (final generation):

(1) *To call him **a communist**, you must **begin with** that he has some kind of **coherent ideology** in the first place.*
(2) *He does not.*
(3) There is **no evidence** *whatsoever* that Trump has **any coherent**, commonly **held beliefs** on **any topic**.

III: *Refinement*

Figure 8.1: An argument generation example using Reddit ChangeMyView. [Top] Partial output by our planner with keyphrase assignment and positions (in subscripts) for each sentence, segmented by special token [SEN], from which a template is constructed. [Bottom] A draft is first produced and then refined, with updated words highlighted in *italics*.

2019; Moryossef et al., 2019; Hua and Wang, 2019), yet those solutions require model architecture modification or retraining, making text generation with large models a very costly endeavor.

To this end, this work aims to bring new insights into *how to effectively incorporate content plans into large models to generate more relevant and coherent text*. We first study a *planning model trained from BERT* (Devlin et al., 2019) to produce the initial content plan, which assigns keyphrases to different sentences and predicts their positions. Next, we propose a *content-controlled text generation framework*, built upon the pre-trained sequence-to-sequence (seq2seq) Transformer model BART (Lewis et al., 2020a). As shown in Figure 8.1, our generation model takes in a content plan consisting of *keyphrase assignments* and their corresponding *positions* for each sentence. The plan is encoded as a template, with [MASK] tokens added at positions where no content is specified. Our model then outputs a fluent and coherent multi-sentence text (draft) to reflect the plan. This is done by fine-tuning BART without modifying its architecture.

Furthermore, we present an *iterative refinement algorithm* to improve the generation in multiple passes, within the seq2seq framework. At each iteration, tokens with low generation confidence are replaced with [MASK] to compose a new template, from which a new output is produced. Unlike prior refinement algorithms that only permit editing in place, our solution offers more flexibility. Figure 8.1 exemplifies the refinement outcome.

We call our system **PAIR** (Planning And Iterative Refinement).¹ It is experimented on three distinct domains: counter-argument generation with Reddit ChangeMyView data, opinion article writing with the New York Times (NYT) corpus, and news report production on NYT. Automatic evaluation with BLEU, ROUGE, and METEOR shows that, by informing the generation model with sentence-level content plans, our model significantly outperforms a BART model fine-tuned with the same set of keyphrases as input (§ 8.4.1). Human

¹Code and data are available at: <http://xinyuhua.github.io/Resources/emnlp20/>

judges also rate our system outputs as more relevant and coherent (§ 8.4.2). Additionally, our iterative refinement strategy consistently improves the generation quality according to both automatic scores and human evaluation. Finally, our model achieves better content control by reflecting the specified keyphrases in the content plan, whose outputs are preferred by human to another version with weaker control.

To summarize, our major contributions include:

- We propose a novel content planner built upon BERT to facilitate long-form text generation.
- We present a novel template mask-and-fill method to incorporate content planning into generation models based on BART.
- We devise an iterative refinement algorithm that works within the seq2seq framework to flexibly improve the generation quality.

8.2 Content-controlled Text Generation with PAIR

Task Description. Our input consists of (1) a sentence-level prompt x , such as a news headline, or a proposition in an argument, and (2) a set of keyphrases m that are relevant to the prompt. The system aims to generate y that contains multiple sentences, as in a news report or an argument, by reflecting the keyphrases in a coherent way.

In this section, we first introduce content planning built upon BERT, that assigns keyphrases into sentences and predicts their positions (§ 8.2.1). Then we

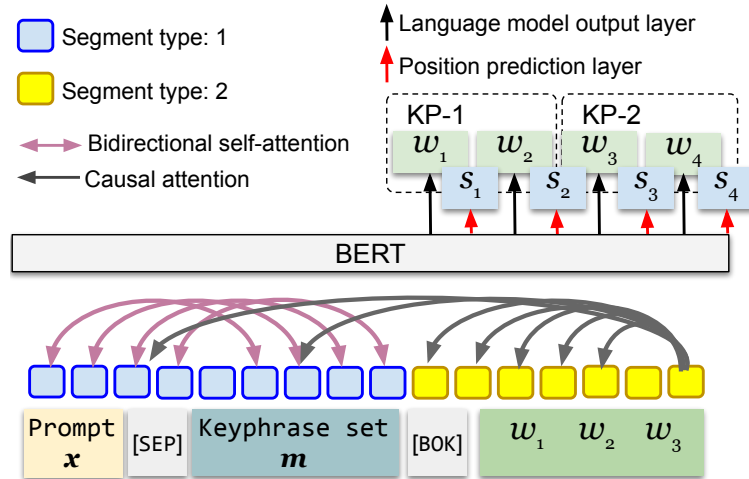


Figure 8.2: Content planning with BERT. We use bidirectional self-attentions for input encoding, and apply causal self-attentions for keyphrase assignment and position prediction. The input (x, m) and output keyphrase assignments (m') are distinguished by different segment embeddings.

propose a seq2seq generation framework with BART fine-tuning that includes a given content plan derived from keyphrases m (§ 8.2.2). Finally, § 8.2.3 discusses improving generation quality by iteratively masking the less confident predictions and regenerating within our framework.

8.2.1 Content Planning with BERT

Our content planner is trained from BERT to assign keyphrases to different sentences and predict their corresponding positions. As shown in Figure 8.2, the concatenation of prompt x and unordered keyphrases m is encoded with bidirectional self-attentions. Keyphrase assignments are produced autoregressively as a sequence of tokens $m' = \{w_j\}$, with their positions in the sentence $s = \{s_j\}$ predicted as a sequence tagging task.

We choose BERT because it has been shown to be effective at both language

modeling and sequence tagging. Moreover, we leverage its segment embedding to distinguish the input and output sequences. Specifically, we reuse its pre-trained language model output layer for keyphrase assignment. We further design a separate keyphrase positioning layer to predict token position s_j as the relative distance from each sentence’s beginning:

$$p(s_j|\mathbf{w}_{\leq j}) = \text{softmax}(\mathbf{H}^L \mathbf{W}_s) \quad (8.1)$$

where \mathbf{H}^L is the last layer hidden states of the Transformer, and \mathbf{W}_s are the newly added keyphrase positioning parameters learned during BERT fine-tuning. The range of allowed positions is from 0 to 127.

Noticeably, as our prediction is done autoregressively, attentions should only consider the generated tokens, but not the future tokens. However, BERT relies on bidirectional self-attentions to attend to both left and right. To resolve this discrepancy, we apply causal attention masks (Dong et al., 2019) over \mathbf{m}' to disallow attending to the future (gray arrows in Figure 8.2).

Training the Planner. We extract keyphrases and acquire their ground-truth positions from human-written references, and fine-tune BERT with cross-entropy losses for both assignment and positioning, with a scaling factor 0.1 over the positioning loss.

Inference. A [BOK] token signals the beginning of keyphrase assignment generation. We employ a greedy decoding algorithm, and limit the output vocabulary to tokens in \mathbf{m} and ensure each keyphrase is generated at most once. To allow sentence-level content planning, a special [SEN] token is generated to represent the sentence boundary, with its predicted position indicating the length. The planning process terminates when [EOS] is produced.

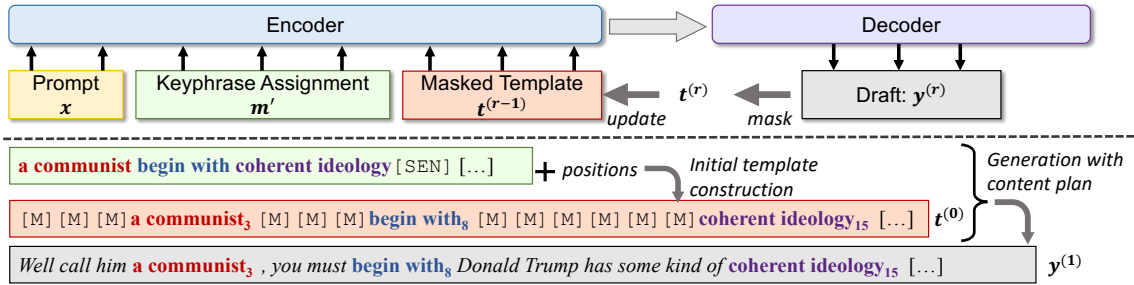


Figure 8.3: Our content-controlled text generation framework, PAIR, which is built on BART. Decoding is executed iteratively. At each iteration, the encoder consumes the input prompt x , the keyphrase assignments m' , as well as a partially masked template ($t^{(r-1)}$ for the r -th iteration, $[M]$ for masks). The autoregressive decoder produces a complete sequence $y^{(r)}$, a subset of which is further masked, to serve as the next iteration’s template $t^{(r)}$.

8.2.2 Adding Content Plan with a Template Mask-and-Fill Procedure

Given a content planning model, we invoke it to output keyphrase assignments to different sentences (m'), their corresponding positions s , along with each sentence’s length (based on the prediction of $[SEN]$). We first employ a post-processing step to convert between different tokenizers, and correct erroneous position predictions that violate the assignment ordering or break the consecutivity of the phrase. We then convert the plan into a **template** $t^{(0)}$ as follows: For each sentence, the assigned keyphrases are placed at their predicted positions, and empty slots are filled with $[MASK]$ symbols. Figure 8.3 illustrates the template construction process and our seq2seq generation model. In Table 8.1, we show statistics on the constructed templates. Specifically, we show the statistics on the automatically created templates based on the planner’s output. As we can see, our system predicted templates approach human reference in terms of length, per sentence keyphrase count, and the average keyphrase spacing. Sentence segmentation occurs more often in our templates than the reference text,

	ARGGEN		OPINION		NEWS	
	<i>sys</i>	<i>ref</i>	<i>sys</i>	<i>ref</i>	<i>sys</i>	<i>ref</i>
# tokens	133.3	130.2	228.5	246.3	424.5	435.5
# sentences	8.6	5.6	11.1	8.2	19.2	13.5
# KP per sent.	2.96	3.77	2.22	2.49	3.40	3.24
KP distance	2.61	2.95	5.70	6.02	3.76	5.08

Table 8.1: Statistics on generated templates by our content planner. Tokens are measured in units of WordPiece (Sennrich et al., 2016). KP distance denotes the average number of tokens between two keyphrases that are in the same sentence. Both system output (*sys*) and human reference (*ref*) are reported.

likely due to the frequent generation of [SEN] tokens.

The input prompt x , keyphrase assignments m' , and template $t^{(0)}$ are concatenated as the input to the encoder. The decoder then generates an output $y^{(1)}$ according to the model’s estimation of $p(y^{(1)}|x, m', t^{(0)})$. $y^{(1)}$ is treated as a draft, to be further refined as described in the next section.

Our method is substantially different from prior work that uses constrained decoding to enforce words to appear at specific positions (Hokamp and Liu, 2017; Post and Vilar, 2018; Hu et al., 2019), which is highly biased by the surrounding few words and suffers from disfluency. Since BART is trained to denoise the masked input with contextual understanding, it naturally benefits our method.

Decoding. We employ the nucleus sampling strategy (Holtzman et al., 2019), which is shown to yield superior output quality in long text generation. In addition to the standard top- k sampling from tokens with the highest probabilities, nucleus sampling further limits possible choices based on a cumulative probability threshold (set to 0.9 in all experiments below). We also require the *keyphrases to be generated at or nearby their predicted positions*. Concretely, for

positions that match any keyphrase token, we force the decoder to copy the keyphrase unless it has already been generated in the previous five tokens. We sample three times to choose the one with the lowest perplexity, as estimated by GPT-2_{base} (Radford et al., 2019).

8.2.3 Iterative Refinement

Outputs generated in a single pass may suffer from incorrectness and incoherence (see Figure 8.1), therefore we propose an iterative refinement procedure to improve the quality. In each pass, tokens with low generation confidence are masked (Algorithm 1). This is inspired by iterative decoding designed for inference acceleration in *non-autoregressive* generation (Lee et al., 2018; Lawrence et al., 2019), though their refinement mostly focuses on word substitution and lacks the flexibility for other operations. Moreover, our goal is to improve fluency while ensuring the generation of given keyphrases.

At each iteration, the n least confident tokens are replaced with [MASK].² Similar as the mask-predict algorithm (Ghazvininejad et al., 2019), we gradually reduce the number of masks. In our experiments, each sample is refined for 5 iterations, with n decaying linearly from 80% of $|y^{(r)}|$ to 0.

Training the Generator. Our training scheme is similar to masked language model pre-training. Given the training corpus $\mathcal{D} = \{(x_i, \mathbf{m}'_i, y_i)\}$, we consider two approaches that add noise to the target y_i by randomly masking a subset of (1) any tokens, or (2) tokens that are not within the span of any keyphrase. The latter is better aligned with our decoding objective, since keyphrases are

²Note that we force the keyphrases to not be replaced during refinement.

Algorithm 1: Iteratively refinement via template mask-and-fill. The sample with the lowest perplexity (thus with better fluency) is selected for each iteration.

Data: prompt x , keyphrase assignments m' , keyphrase positions s , R refinement iterations, ρ nucleus sampling runs
Result: final output $y^{(R)}$
Construct template $t^{(0)}$ based on m' and s ;
for $r = 1$ **to** R **do**
 Run encoder over $x \oplus m' \oplus t^{(r-1)}$;
 $\mathcal{Y} \leftarrow \emptyset$;
 for $i = 1$ **to** ρ **do**
 Run nucleus sampling to generate y_i with keyphrase position enforcement;
 Append y_i to \mathcal{Y} ;
 $y^{(r)} \leftarrow \operatorname{argmin}_{y_i \in \mathcal{Y}} \text{GPT2-PPL}(y_i)$;
 $n \leftarrow |y^{(r)}| \times (1 - r/R)$;
 Mask n tokens with the lowest probabilities to create new template $t^{(r)}$;

never masked. We concatenate x_i , m'_i , and the corrupted target \tilde{y}_i as input, and fine-tune BART to reconstruct the original y_i with a cross-entropy loss.

8.3 Experiment Setups

8.3.1 Tasks and Datasets

We evaluate our generation and planning models on datasets from three distinct domains for multi-paragraph-level text generation: (1) argument generation (**ARGGEN**) (Hua et al., 2019a), to produce a counter-argument to refute a given proposition; (2) writing opinionated articles (**OPINION**), e.g., editorials and op-eds, to show idea exchange on a given subject; and (3) composing news reports (**NEWS**) to describe events. The three domains are selected with diverse

	# Sample	Prompt	Target	# KP	KP Cov.
ARGGEN	56,504	19.4	116.6	20.6	30.5%
OPINION	104,610	6.1	205.6	19.0	26.0%
NEWS	239,959	7.0	282.7	30.3	32.6%

Table 8.2: Statistics of the three datasets. We report average lengths of the prompt and the target generation, number of unique keyphrases (# KP) used in the input, and the percentage of content words in target covered by the keyphrases (KP Cov.).

levels of subjectivity and various communicative goals (persuading vs. informing), with statistics shown in Table 8.2.

Task 1: Argument Generation. We first evaluate our models on persuasive argument generation, based on a dataset collected from Reddit `r/ChangeMyView` (CMV) in our prior work (Hua et al., 2019a) (Chapter 6). This dataset contains pairs of original post (OP) statement on a controversial issue about politics and filtered high-quality counter-arguments, covering 14,833 threads from 2013 to 2018. We use the OP title, which contains a proposition (e.g. *the minimum wage should be abolished*), to form the input prompt x . In our prior work, only the first paragraphs of high-quality counter-arguments are used for generation. Here we consider generating the full post, which is significantly longer. Keyphrases are identified as noun phrases and verb phrases that contain at least one topic signature word (Lin and Hovy, 2000a), which is determined by a log-likelihood ratio test that indicates word salience. Following our prior work, we expand the set of topic signatures with their synonyms, hyponyms, hypernyms, and antonyms according to WordNet (Miller, 1994). The keyphrases longer than 10 tokens are further discarded.

Task 2: Opinion Article Generation. We collect opinion articles from the New York Times (NYT) corpus (Sandhaus, 2008). An article is selected if its

taxonomies label has a prefix of *Top/Opinion*. We eliminate articles with an empty headline or less than three sentences. Keyphrases are extracted in a similar manner as done in argument generation. Samples without any keyphrase are removed. The article headline is treated as the input, and our target is to construct the full article. Table 8.2 shows that opinion samples have shorter input than arguments, and the keyphrase set also covers fewer content words in the target outputs, requiring the model to generalize well to capture the unseen tokens.

Task 3: News Report Generation. Similarly, we collect and process news reports from NYT, filtering by taxonomy labels starting with *“Top/News”*, removing articles that have no content word overlap with the headline, and ones with `material-types` labeled as one of *“statistics”*, *“list”*, *“correction”*, *“biography”*, or *“review.”* News reports describe events and facts, and in this domain we aim to study and emphasize the importance of faithfully reflecting content plans during generation and refinement.

Data Split and Preprocessing. For argument generation, we split the data into 75%, 12.5%, and 12.5% for training, validation, and test sets. To avoid test set contamination, the split is conducted on thread level. For opinion and news generation, we reserve the most recent 5k articles for testing, another 5k for validation, and the rest (23k for news and 10k for opinion) are used for training. We apply the BPE tokenization (Sennrich et al., 2016) for the generation model as BART does, and use WordPiece (Wu et al., 2016) for BERT-based planner. To fit the data into our GPUs, we truncate the target size to 140 tokens for argument, sizes of 243 and 335 are applied for opinion and news, for both training and inference.

8.3.2 Implementation Details

Our code is written in PyTorch (Paszke et al., 2017) using the Huggingface Transformer library (Wolf et al., 2020). For fine-tuning, we adopt the standard linear warmup and inverse square root decaying scheme for learning rates, with a maximum value of 5×10^{-5} . Adam (Kingma and Ba, 2015) is used as the optimizer, with a batch size of 10 for refinement and 20 for content planning, and a maximum gradient clipped at 1.0. All hyperparameters are tuned on validation set, with early stopping used to avoid overfitting.

Model Sizes. Our generation model has the same architecture as BART (Lewis et al., 2020a) with 406M parameters. The content planner is built on top of BERT_{base}, which has 110M parameters.

Running Time. For both training and decoding, we utilize the Titan RTX GPU card with 24 GB memory. Training the generation model takes 2.5 hours for argument, 5 hours for opinion, and 24 hours for news. The content planning model converges in 2.5-4 hours for three domains.

Decoding Settings. At inference time, we set $k = 50$, temperature=1.0, and $p = 0.9$ for nucleus sampling. The relatively large k value is determined based on a pilot study, where we find that the refinement lacks diversity if k is set to small values. Moreover, since the Transformer states need to be cached during autoregressive decoding and we perform three complete nucleus sampling runs in each refinement iteration, the GPU memory consumption is substantially increased. We therefore limit the maximum generation steps to 140 for argument, 243 and 335 for opinion and news.

8.3.3 Baselines and Comparisons

We consider two baselines, both are fine-tuned from BART as in our models: (1) **SEQ2SEQ** directly generates the target from the prompt; (2) **KPSEQ2SEQ** encodes the concatenation of the prompt and the *unordered* keyphrase set. To study if using only sentence-level keyphrase assignments helps, we include a model variant (**PAIR_{light}**) by removing keyphrase position information (*s*) from the input of our generator and using an initial template with all [MASK] symbols. Our model with full plans is denoted as **PAIR_{full}**. We first report generation results using *ground-truth content plans* constructed from human-written text, and also show the end-to-end results with *predicted content plans* by our planner.

8.4 Results

8.4.1 Automatic Evaluation

We report scores with BLEU (Papineni et al., 2002), which is based on n-gram precision (up to 4-grams); ROUGE-L (Lin, 2004), measuring recall of the longest common subsequences; and METEOR (Denkowski and Lavie, 2014), which accounts for paraphrase. For our models PAIR_{full} and PAIR_{light}, we evaluate both the first draft and the final output after refinement. Table 8.3 lists the results when ground-truth content plans are applied.

First, *our content-controlled generation model with planning consistently outperforms comparisons and other model variants on all datasets, with or without iterative refinement.* Among our model variants, PAIR_{full} that has access to full con-

	BLEU-4	ROUGE-L	METEOR	Length
ARGGEN				
SEQ2SEQ	0.76	13.80	9.36	97
KPSEQ2SEQ	6.78	19.43	15.98	97
PAIR _{light}	26.38	47.97	31.64	119
PAIR _{light} w/o refine	25.17	46.84	31.31	120
PAIR _{full}	36.09	56.86	33.30	102
PAIR _{full} w/o refine	34.09	55.42	32.74	101
OPINION				
SEQ2SEQ	1.42	15.97	10.97	156
KPSEQ2SEQ	11.38	22.75	18.38	164
PAIR _{light}	16.27	33.30	24.32	210
PAIR _{light} w/o refine	15.45	32.35	24.11	214
PAIR _{full}	23.12	40.53	24.73	167
PAIR _{full} w/o refine	22.17	39.71	24.65	169
NEWS				
SEQ2SEQ	1.11	15.60	10.10	242
KPSEQ2SEQ	11.61	21.05	18.61	286
PAIR _{light}	28.03	43.39	27.70	272
PAIR _{light} w/o refine	27.32	43.08	27.35	278
PAIR _{full}	34.37	51.10	29.50	259
PAIR _{full} w/o refine	33.48	50.27	29.26	260

Table 8.3: Key results on argument generation, opinion article writing, and news report generation. BLEU-4 (B-4), ROUGE-L (R-L), METEOR (MTR), and average output lengths are reported (for references, the lengths are 100, 166, and 250, respectively). PAIR_{light}, using keyphrase assignments only, consistently outperforms baselines; adding keyphrase positions, PAIR_{full} further boosts scores. Improvements by our models over baselines are all significant ($p < 0.0001$, approximate randomization test). Iterative refinement helps on both setups.

tent plans obtains significantly better scores than PAIR_{light} that only includes keyphrase assignments but not their positions. Lengths of PAIR_{full}'s outputs are also closer to those of human references. Both imply the benefit of keyphrase positioning.

Table 8.3 also shows that *the iterative refinement strategy can steadily boost performance* on both of our setups. By inspecting the performance of refinement in different iterations (Figure 8.4), we observe that both BLEU and ROUGE-L

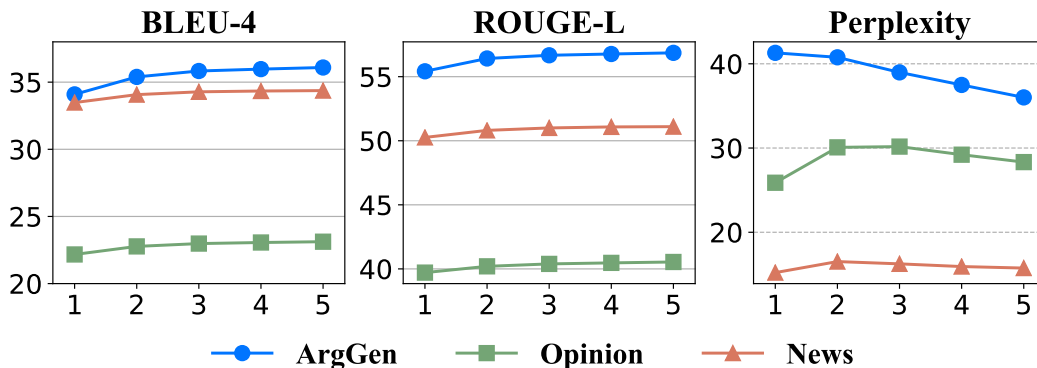


Figure 8.4: Results on iterative refinement with five iterations. Both BLEU and ROUGE-L scores steadily increase, with perplexity lowers in later iterations.

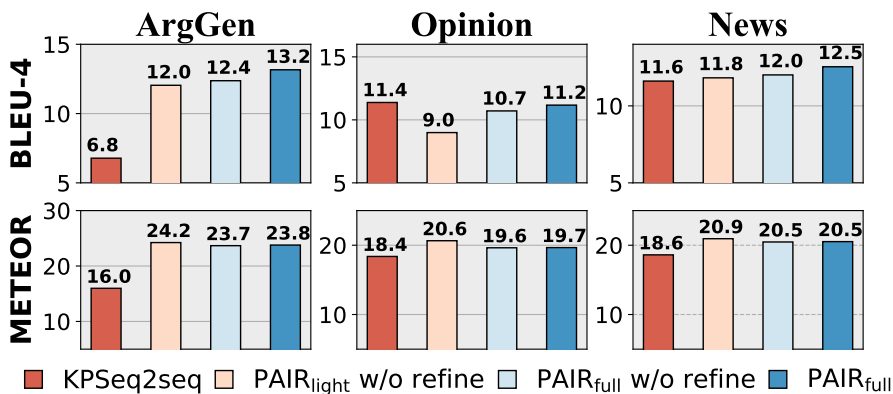


Figure 8.5: End-to-end generation results with automatically predicted content plans. Our models outperform KPSEQ2SEQ in both metrics, except for BLEU-4 on opinion articles where results are comparable.

scores gradually increase while perplexity lowers as the refinement progresses. This indicates that iterative post-editing improves both content and fluency.

Results with Predicted Content Plans. We further report results by using content plans predicted by our BERT-based planner. Figure 8.5 compares PAIR_{full} and PAIR_{light} with KPSEQ2SEQ. Our models yield better METEOR scores on all three domains. That said, the improvement from predicted plans is not as pronounced as that from ground-truth plans. Upon inspection, we find that our planner often falls short of accurately positioning the given keyphrases, lead-

ARGGEN	Fluency	Coherence	Relevance
KPSEQ2SEQ	4.63	3.28	2.79
PAIR _{light}	4.75	3.97*	3.85*
PAIR _{full}	4.46	3.76*	3.79*

Table 8.4: Human evaluation for argument generation on fluency, coherence, and relevance, with 5 as the best. The Krippendorff’s α are 0.28, 0.30, and 0.37, respectively. Our model outputs are significantly more coherent and relevant than KPSEQ2SEQ (*: $p < 0.0001$), with comparable fluency.

ing to degraded generation performance. This points to a potential direction for future work where better positioning model should be developed.

8.4.2 Human Evaluation

We hire four proficient English speakers³ to rate three aspects of the generated arguments on a scale of 1 (worst) to 5 (best): **fluency**, **coherence**—if the information organization is natural and logical, and **relevance**—if the topic is related to the prompt and whether the stance is correct. 50 samples are randomly selected, with system outputs by KPSEQ2SEQ, PAIR_{full} and PAIR_{light} shown to human judges in random order.

Table 8.4 shows that both of our models are rated with better coherence and relevance than KPSEQ2SEQ which uses the same but unordered keyphrases as input. Interestingly, outputs by PAIR_{light} are regarded as more fluent and coherent, though the difference is not significant. However, discourse analysis in § 8.5 reveals that clauses produced by PAIR_{light} are more locally related, compared to PAIR_{full}, which can be perceived as easier to read. In addition to the sample argument in Figure 8.1, Table 8.5 shows PAIR_{full}’s output in the news and opinion

³They are all US-based college students. Each of them is paid \$15 hourly for the task.

Prompt (News): *4 Arrested in Theft of Baby Jesus Figurines*

PAIR_{full}: **Four New Jersey teenagers arrested yesterday** were accused of stealing more than **25 plastic baby Jesus figurines** from a church before they **burn in a bonfire**, **the police** said. **The police** in **Sayreville, N.J.**, **arrested Michael Payne, 18**, and T.J. Jones, 18, of Union City; **Nicholas M. Hess, 18**, of **Matawan**; and Jason L. O’Neill, 18, of Port New York, N.J., and charged them with burglary. Their vandals removed more than 100 figurines of Jesus from **a cemetery** outside **St. Stanislaus Kostka Church** in Sayreville, the police said. Detectives said the four had taken their **flashlights** to **the cemetery** and **jotted down the license plates** of several cars in **the vicinity** [...]

Prompt (Opinion): *Drowning in a Sea of College Debt*

PAIR_{full}: Once again, the appalling problem of **increasing debt** in **financing higher education** that has taken place is more **intolerable**, particularly because of the arrogance of our colleges and universities. The burden of **saddling college** students with debt in the middle of their **teenage years**, when they were in debt, is essential for **a good education**. Our educational system is designed to allow kids to develop the skills necessary, but it does not **create optimal conditions** for mature students who know they will not be able [...]

Table 8.5: Sample outputs in the news and opinion domain. Keyphrases assigned to different sentences are in boldface and color-coded.

domains. More samples by different systems are in the supplementary material.

Effect of Refinement and Keyphrase Enforcement. We further ask *whether human judges prefer the refined text* and *whether enforcing keyphrases to be generated yields noticeable content improvement*. In a second study, we present the same 50 prompts from the previous evaluation on argument generation, and an additional 50 samples for opinion article writing to the same group of human judge. For each sample, PAIR_{full}’s outputs with and without refinement are shown in random order. Judges indicate their preference based on the overall quality. The same procedure is conducted to compare with a version where we do not enforce keyphrases to be copied at their predicted positions during decoding. Table 8.6 demonstrates that the refined text is preferred in more than half of the cases, for both domains. Enforcing keyphrase generation based on their posi-

	PAIR _{full}	<i>w/o refine</i>	PAIR _{full}	<i>w/o enforce</i>
ARGGEN	52.7%	33.3%	45.3%	40.0%
OPINION	52.7%	30.7%	50.0%	29.3%

Table 8.6: Percentages of samples preferred by human judges before and after refinement [Left]; with and without enforcing keyphrases to appear at the predicted positions [Right]. Ties are omitted.

tions is also more favorable than not enforcing such constraint.

What is updated during iterative refinement? Since refinement yields better text, we compare generations before and after the refinement. First, we find that masks are regularly put on “functional” words and phrases. For example, stopwords and punctuation along with their bigrams are often swapped out, with new words filled in to improve fluency. Moreover, about 85% of the refinement operations result in new content being generated. This includes changing prepositions and paraphrasing, e.g., replacing “*a research fellow*” with “*a graduate student*.” On both news and opinion domains, numerical and temporal expressions are often incorrectly substituted, suggesting that better fact control needs to be designed to maintain factuality.

8.5 Further Discussions on Discourse

Prior work’s evaluation mainly focuses on fluency and content relevance, and largely ignores the discourse structure exposed by the generated text. However, unnatural discourse and lack of focus are indeed perceived as major problems of long-form neural generations, as identified by human experts.⁴ Here, we aim to investigate whether content-controlled generation with ground-truth content

⁴<https://www.economist.com/open-future/2019/10/01/how-to-respond-to-climate-change-if-you-are-an-algorithm>

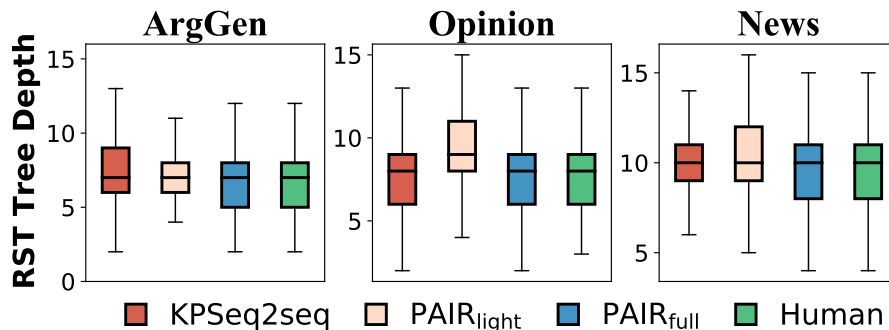


Figure 8.6: Distributions of RST tree depth. PAIR_{full} better resembles the patterns in human-written texts.

plans resembles human-written text by studying discourse phenomena.

Are PAIR generations similar to human-written text in discourse structure?

We utilize DPLP (Ji and Eisenstein, 2014), an off-the-shelf Rhetorical Structure Theory (RST) discourse parser. DPLP converts a given text into a binary tree, with elementary discourse units (EDUs, usually clauses) as nucleus and satellite nodes. For instance, a relation `NS-elaboration` indicates the second node as a satellite (S) elaborating on the first nucleus (N) node. DPLP achieves F1 scores of 81.6 for EDU detection and 71.0 for relation prediction on news articles from the annotated RST Discourse Treebank (Carlson et al., 2001). We run this trained model on our data for both human references and model generations.

First, we analyze the *depth of RST parse trees*, which exhibits whether the text is more locally or globally connected. For all trees, we truncate at a maximum number of EDUs based on the 90 percentile of EDU count for human references. Distributions of tree depth are displayed in Figure 8.6. As can be seen, generations by PAIR_{full} show similar patterns to human-written arguments and articles. We also find that trees by PAIR_{light} tend to have a more “linear” structure, highlighting the dominance of local relations between adjacent EDUs, compared with PAIR_{full} which uses knowledge of keyphrases positions. This implies

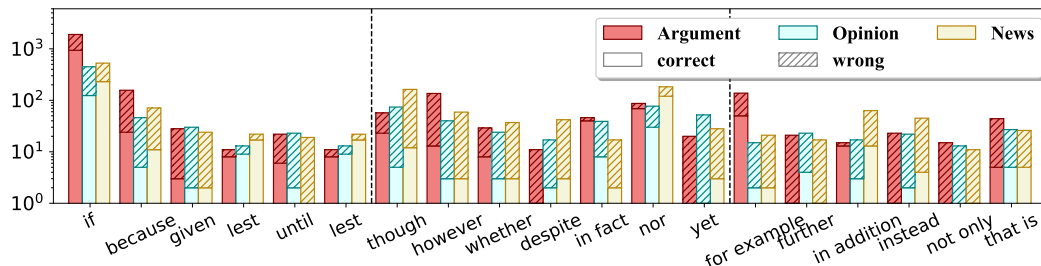


Figure 8.7: Discourse markers that are correctly and incorrectly (shaded) generated by PAIR_{full}, compared to aligned sentences in human references. Discourse markers are grouped (from left to right) into senses of CONTINGENCY (higher marker generation accuracy observed), COMPARISON, and EXPANSION. y-axis: # of generated sentences with the corresponding marker.

that content positioning helps with structure at a more global level. We further look into the *ratios of NS, NN, SN relations*, and observe that most model outputs have similar trends as human-written texts, except for KPSEQ2SEQ which has more SN relations, e.g., it produces twice as many SNs than others on arguments.

Can PAIR correctly generate discourse markers? Since discourse markers are crucial for coherence (Grote and Stede, 1998; Callaway, 2003) and have received dedicated research efforts in rule-based systems (Reed et al., 2018; Balakrishnan et al., 2019), we examine if PAIR_{full} can properly generate them. For each sample, we construct sentence pairs based on content word overlaps between system generation and human reference. We manually select a set of unambiguous discourse markers from Appendix A of the Penn Discourse Treebank manual (Prasad et al., 2008). When a marker is present in the first three words in a reference sentence, we check if the corresponding system output does the same.

Figure 8.7 displays the numbers of generated sentences with markers produced as the same in human references (correct) or not (wrong). The markers are grouped into three senses: CONTINGENCY, COMPARISON, and EXPANSION.

The charts indicates that PAIR_{full} does better at reproducing markers for CONTINGENCY, followed by COMPARISON and EXPANSION. Manual inspections show that certain missed cases are in fact plausible replacements, such as using `at the same time` for `in addition`, or also `for further`, while in other cases the markers tend to be omitted. Overall, we believe that content control alone is still insufficient to capture discourse relations, motivating future work on discourse planning.

8.6 Conclusion

We present a novel content-controlled generation framework that adds content planning to large pre-trained Transformers without modifying model architecture. A BERT-based planning model is first designed to assign and position keyphrases into different sentences. We then investigate an iterative refinement algorithm that works with the sequence-to-sequence models to improve generation quality with flexible editing. Both automatic evaluation and human judgments show that our model with planning and refinement enhances the relevance and coherence of the generated content.

CHAPTER 9

DYPLOC: DYNAMIC PLANNING OF CONTENT USING MIXED LANGUAGE MODELS FOR TEXT GENERATION

In this chapter, we introduce an efficient generation framework that conducts dynamic planning of content while generating the output based on a novel design of mixed language models. To enrich the generation with diverse content, we further propose to use large pre-trained models to predict relevant concepts and to generate claims.

This work is published at ACL 2021 (Hua et al., 2021). The relevant resources are at: <https://xinyuhua.github.io/Resources/acl21/>

9.1 Introduction

Opinion articles serve as an important media to convey the authors' values, beliefs, and stances on important societal issues. Automatically generating long-form opinion articles has the potential of facilitating various tasks, such as essay writing assistance and speech drafting, and it is the focus of this work. Though opinion generation has been investigated for constructing arguments (Hua and Wang, 2018), writing reviews (Ni and McAuley, 2018), and producing emotional dialogue responses (Song et al., 2019), those outputs are relatively short. While impressive progress in generation has been achieved by using large pre-trained Transformers (Radford et al., 2019; Lewis et al., 2020a), directly adopting them for long-form opinion text generation poses distinct challenges.

First, large models still fall short of producing coherent text due to the *lack of*

Title CMV: I believe 9/11 would not have happened if Al Gore were elected President.

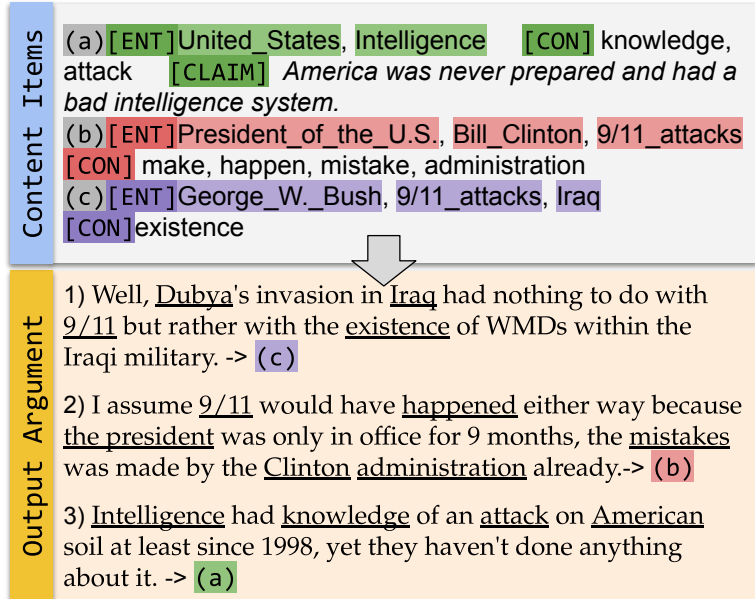


Figure 9.1: Sample argument on Reddit ChangeMyView. Our generator considers an input containing (1) a title and (2) an unordered set of content items. Each content item consists of *elements* of an *entity set* [ENT], a *concept set* [CON], and an optional sentence-level *claim* [CLAIM]. Each output token is generated by conditioning on all content items, and the best aligned ones (learned by our model) are highlighted in corresponding colors. We also underline words that reflect the input concepts and entities.

efficient content control and planning (Ko and Li, 2020; Wu et al., 2020; Tan et al., 2021). A common solution is to use concatenated phrases or semantic representations to guide the generation process (Yao et al., 2019b; Harkous et al., 2020; Ribeiro et al., 2020; Goldfarb-Tarrant et al., 2020), where content planning, including both content selection and ordering, is expected to be learned by attention mechanisms. However, attentions have only achieved limited improvements. Recent work also explores training a separate planning module to produce sorted content, which is then fed into a generator (Fan et al., 2019b; Hua and Wang, 2020; Goldfarb-Tarrant et al., 2020). Nonetheless, this strategy results in a disconnection between planning and realization, and the output is not guar-

anteed to respect the planning results (Castro Ferreira et al., 2019; Prabhumoye et al., 2020).

The second challenge for opinion generation resides in the diversity of information that is needed to produce an output with consistent stances and supported by pertinent facts. Though large models memorize significant amounts of knowledge, they cannot retrieve and operate with them precisely (Lewis et al., 2020b). Due to the argumentative nature of opinion text, simply including knowledge bases (Guan et al., 2020; Zhou et al., 2020) is insufficient to uphold the desired quality, as it requires the combination of subjective claims and objective evidence as supports.

To this end, we propose a novel generation framework, **DYPLOC** (dynamic planning of content), to conduct content selection and ordering as text is produced. Concretely, given a set of unordered content items, as displayed in Figure 9.1, we design mixed language models, with each implemented as a sequence-to-sequence model to encode one item and the input statement. At each decoding step, our system selects which items to reflect, and predicts a word based on probabilities marginalized over all language models. Crucially, our end-to-end trained framework (1) enables the generator to access multiple content items at all times and select content based on what has been generated so far, (2) can be directly built on large pre-trained Transformers, e.g., BART (Lewis et al., 2020a), with planning and generation modules jointly trained, and (3) outputs learned content selection scores to provide an interface for system decision interpretation.

Furthermore, to ensure that our framework can be applied to a broad range of generation tasks, we design content items to cover three critical elements:

entities and concepts that are central to many generation applications, and claims that are building blocks for opinion text. We show an example for counter-argument generation in Figure 9.1. Importantly, we employ BART to predict additional relevant concepts, derived from ConceptNet (Speer et al., 2017), and generate claims, as central propositions, to enrich the generated text with both objective and subjective content.

For experiments, we collect two datasets: (1) posts from Reddit Change-MyView for argument generation, and (2) articles from the New York Times Opinion section (Sandhaus, 2008) for opinion article writing. Our proposed framework outperforms competitive comparisons, such as fine-tuning BART with the same content items, based on automatic metrics of BLEU, ROUGE, and METEOR. Human assessment further confirms that our system outputs have richer content and are more coherent in both tasks.

Our main contributions are summarized as below:

- We present a dynamic content planning generation framework, which is directly built on top of BART. Our design of mixed language models overcomes the lack of control by existing models that use implicit planning with attentions or hard copying.
- We propose content plan augmentation by automatically generating relevant concepts and claims.
- We construct two opinion text generation datasets with content plans that capture prominent entities and concepts.

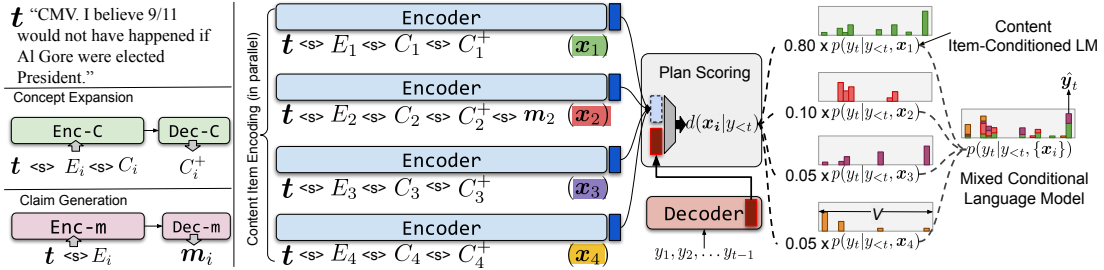


Figure 9.2: Our proposed text generation framework, DYPLOC. [Left] For each input content item (a title t , an entity set E_i , and a core concept set C_i), we first expand it with more relevant concepts, i.e., C_i^+ . For sentences to be realized as claims, we employ a separate generator to produce one draft claim, m_i . [Right] The augmented content items, denoted as $\{x_i\}$, are encoded in parallel. At each decoding step, a plan scoring network estimates a distribution $d(x_i|y_{<t})$ for all content items and decides on relevant content. A word is predicted based on probabilities marginalized over all content item-conditioned language models, i.e., $p(y_t|y_{<t}, x_i)$ for the i -th model.

9.2 Model

Task Formulation. Our opinion text generation framework takes as input a set of **content items**. Each content item consists of a title t , a set of **entities** E_i ¹, such as $\{\text{United_States}, \text{9/11_attacks}\}$, and a set of **core concepts** C_i , such as $\{\text{attack}, \text{knowledge}\}$, that are often abstract notions. Our model first expands C_i by predicting additional **relevant concepts** C_i^+ and optionally generates a pertinent **claim** m_i , and then outputs the final text with multiple sentences as $\mathbf{y} = \{y_t\}$, to faithfully reflect the content items with a coherent structure. An overview of our system is illustrated in Figure 9.2.

Below we first describe the content item augmentation methods (Section 9.2.1), followed by our generator with mixed language models that condition on expanded content items (Section 9.2.2).

¹Note that i distinguishes the items. Their order is random.

9.2.1 Content Item Augmentation

Concept Expansion. With limited number of entities and concepts as input, generation systems are often incapable of producing long text with rich content, resulting in hallucination (Wiseman et al., 2017; Tian et al., 2019). Therefore, from the often-abstract core concepts, we aim to predict more specific concepts that are also relevant to the given title. For instance, as displayed in Figure 9.1, for core concepts $\{make, happen\}$ and entities $\{Bill_Clinton, 9/11_attacks\}$, we grow the input with more concrete concepts of $\{mistake, administration\}$.

We thus consider a concept expansion module $g(\cdot)$, which predicts additional relevant concepts, denoted as C_i^+ , by conditioning on the original content item:

$$C_i^+ = g(\mathbf{t}, E_i, C_i) \quad (9.1)$$

While $g(\cdot)$ can be any conditional predictor, our experiment shows that fine-tuned BART model performs best on our tasks, where it generates C_i^+ word-by-word by consuming the content item.² Training data construction is described in Section 9.3.1.

Claim Generation. As discussed in Section 9.1, opinion text generation should be controlled with consistent propositions, which cannot be effectively expressed by disconnected concepts. Therefore, we argue that natural languages are more suitable for delivering central claims, since they better encode stylistic languages, e.g., persuasion strategies.

²We also exploited a model that uses the structure of knowledge bases, e.g., ConceptNet, for learning to expand concepts, but it yields lower precision and recall than fine-tuning BART does.

Concretely, we fine-tune another BART model by taking in the title t and the entities E_i , which then produces a claim with nucleus sampling for decoding (Holtzman et al., 2019). In this work, we assume the subset of content items that can be used to generate claims is known. Possible future work includes predicting such subsets and filtering claims with quality measurement.

9.2.2 Content Realization via Mixed Conditioning

After obtaining the augmented content items, we leverage pre-trained BART models to encode each of them as a sequence, as illustrated in Figure 9.2. Segmenter $\langle s \rangle$ is added to indicate the change of elements in a content item. Our encoders run over all items $\{x_i\}$ in parallel, from which we extract content item representations $\{h_i\}$, based on the last layer’s hidden states of the first token.

The standard sequence-to-sequence (seq2seq) framework models output probabilities by taking a single sequence as input. It is challenging to extend seq2seq to consider multiple sequences simultaneously, and conduct content planning concurrently. Therefore, we introduce a **plan scoring network**, $d(x_i|y_{<t})$, which learns to dynamically select and order content based on what has been produced previously while generating the outputs. As outlined in Figure 9.2, our generator is informed of all content items during generation. At each decoding step t , the probabilities of output words are estimated as a weighted sum of all content item-conditioned language models as follows:

$$p(y_t|y_{<t}) = \sum_i d(x_i|y_{<t})p(y_t|y_{<t}, x_i) \quad (9.2)$$

$$d(x_i|y_{<t}) = \text{softmax}_i(e_{it}) \quad (9.3)$$

where $p(y_i|y_{<t}, \mathbf{x}_i)$ corresponds to the i -th language model with \mathbf{x}_i as the input. Crucially, $d(\mathbf{x}_i|y_{<t})$ determines the importance of \mathbf{x}_i when generating token y_i and thus achieves the effect of content planning. We design a two-layer feed-forward network to estimate e_{it} :

$$e_{it} = \mathbf{W}_o \tanh(\mathbf{W}_d[\mathbf{h}_i; s_t]) \quad (9.4)$$

where \mathbf{h}_i denotes the representation of content item \mathbf{x}_i , s_t the decoder state, and \mathbf{W}_o and \mathbf{W}_d learnable parameters. Although mixed language models have been used by Lewis et al. (2020b) to include retrieved documents for question answering, their relevance scores are given by external retrieval models, whereas our plan scorer $d(\mathbf{x}_i|y_{<t})$ is learned together with the generator.

Training and Decoding. Our model is end-to-end trained with both the standard cross-entropy loss \mathcal{L}_{gen} over the tokens in the target generations and a separate loss \mathcal{L}_{plan} for learning $d(\mathbf{x}_i|y_{<t})$:

$$\mathcal{L}(\theta) = \mathcal{L}_{gen}(\theta) + \mathcal{L}_{plan}(\theta) \quad (9.5)$$

To create labels for \mathcal{L}_{plan} , we leverage the correspondence between content items and target tokens, i.e., $d(\mathbf{x}_i|y_{<t})$ is optimized to approach 1 if y_i is in the sentence that derives \mathbf{x}_i , otherwise 0.³ Details about training data construction is in Section 9.3.2.

At each decoding step, the individual language models, $p(y_i|y_{<t}, \mathbf{x}_i)$, and the distribution scores, $d(\mathbf{x}_i|y_{<t})$, are first calculated in parallel. We then decode each token greedily based on the mixed language models in an autoregressive way.

³We also experimented with a training objective consisting of the generation loss only, but the performance degraded significantly. Future directions include removing the training signals for planning.

9.3 Experiment Setups

We experiment with the tasks of argument generation and opinion article writing (Section 9.3.1). Both tasks require generating multi-sentence output, and contain a substantial amount of opinions and factual content. We describe the construction of initial content items and the training data for generating expanded concepts and claims in Section 9.3.2. We present models for comparison in Section 9.3.3.

9.3.1 Tasks and Datasets

Argument Generation. We collect arguments from Reddit’s ChangeMyView (CMV) community, an online forum that features argumentative discussions. Each thread begins with an original post (OP) stating an opinion towards a controversial topic, e.g., “*The U.S. is too big for one government*”. High-quality replies with community endorsement are collected from our prior work (Hua and Wang, 2020) (Chapter 8), covering content posted from 2013 to 2018, In this work, we extend the data collection to 2019. Our goal is to generate the entire reply (i.e., the target) given the OP title. Statistics about the CMV dataset are listed in Table 9.1. We reserve the most recent 1,000 samples for test and another 1,000 for validation.

Opinion Article Writing. Our second task is to generate opinion articles, as collected from the New York Times (NYT) corpus (Sandhaus, 2008). We retain articles whose `taxonomy` labels include *Top/Opinion*. To ensure that articles can be processed by our computing resource, we only keep the ones with at most

	CMV	NYT
# Samples	77,245	113,616
Avg. Title Length	19.2	5.9
Avg. # Content Items (% w/ Claims)	6.8 (76.5%)	9.3 (38.9%)
Avg. # Core Concepts	3.6	4.8
Avg. # Predicted Concepts	4.2	4.3
Avg. # Entities	0.8	0.7
Avg. Target Generation Length	142.0	218.9
Cov. by Core Concepts	13.2%	14.9%
Cov. by Augmented Concepts	16.9%	18.7%
Cov. by Augmented Content Items	52.4%	39.1%

Table 9.1: Statistics of the two datasets. We report average numbers of concepts and entities per content item, and the coverage of words in target generations by different input options.

20 sentences, representing 60% of all opinion articles. As shown in Table 9.1, NYT outputs tend to be significantly longer and contain less claims than CMV. Similarly, we keep 1,000 examples each for test and validation sets.

9.3.2 Content Item Construction

From target references, we describe how to automatically construct the input content items consisting of entities and core concepts, and how to collect training data to fine-tune BART to predict more specific concepts and additional claims. Prior work has demonstrated the benefits of incorporating knowledge bases for text generation (Clark et al., 2018; Puduppully et al., 2019; Guan et al., 2020). We thus consider two sources of knowledge: (1) entities from Wikipedia, which are useful for modeling events and opinion targets, and (2) concept words from ConceptNet (Speer et al., 2017), that covers more related details.

Entity Linking. We first segment a reference into sentences. The ones with fewer than 5 tokens are discarded for content item construction. For the rest, we extract entity mentions using Stanford CoreNLP (Manning et al., 2014), and further include nominal noun phrases. For entity linking, we adopt CrossWiki (Spitkovsky and Chang, 2012), which can process our large-scale data within a reasonable amount of time. CrossWiki maps a mention to a list of frequently linked Wikipedia entries. We further manually verify and correct the linking results for the top 500 most frequent mentions.

Concept Extraction. To identify concepts in a reference, we match the lemmatized unigrams and their part-of-speech (POS) tags against all ConceptNet entries. To create a reasonably challenging task, we only keep a subset of the matches for inclusion in the core concept set (i.e., C_i), with the rest used as C_i^+ , to be generated by our concept expansion model. Furthermore, we conjecture that an opinion article author tends to start with high-level topics that cover more abstract topical words. We thus leverage a lexicon (Brysbaert et al., 2014) with concreteness scores, ranging from 0 (abstract) to 5 (concrete), for over 40k English words. We keep concepts that are verbs or have a concreteness score lower than 3.0. Word coverage of references by using core concepts and additionally with augmented concepts are 13.2% and 16.9% on CMV respectively, and similarly on NYT (Table 9.1). Finally, we **train a concept generator** with BART to produce C_i^+ , conditional on C_i , the title, and the entities.

Claim Detection and Generation. Claims are indispensable for opinion articles. As described in Section 9.2.1, we aim to enrich content items with claims targeting the given entities within the title’s context.

To this end, we first **train a claim detector** by fine-tuning a BERT_{base} (Devlin et al., 2019) sequence classifier with a dataset consisting of sentences of `claims` and `facts`. Concretely, we collect 54,802 claim sentences from Kialo⁴, a repository for debate arguments. We then sample 50,000 sentences from Wikipedia, which are treated as facts. This classifier is applied on a reference, and sentences that are labeled as claims become the target for our claim generator.

We then **learn a claim generator** using BART, which takes in the title and the entities, and outputs the claim. We augment our training data with replies collected from 30 active subreddits related to political discussions, with details in Table 9.2. In total, 80,566 sentences, which contain at least one entity and are labeled by our classifier as `claims`, are kept to train the generator.

Anarchism	AmericanPolitics
Capitalism	Anarcho.Capitalism
Conservative	democracy
democrats	feminisms
government	GreenParty
IWW	labor
Liberal	Libertarian
LibertarianLeft	LibertarianSocialism
Marxism	moderatepolitics
Objectivism	PoliticalDiscussion
politics	progressive
Republican	republicans
socialdemocracy	socialism
ukpolitics	uspolitics
worldpolitics	PoliticalPhilosophy

Table 9.2: List of subreddit ids used to construct training data for claim generator.

⁴<https://www.kialo.com/>

	BLEU-2	ROUGE-2	METEOR	Length
Argument Generation (CMV)				
RETRIEVAL	6.29	3.68	10.00	78
SENTPLANNER	7.78	3.23	7.69	114
SEQ2SEQ	16.71	9.53	13.34	100
SEQ2SEQFULL	29.11	17.71	20.27	145
DYPLOC (ours)	32.60	25.69	22.61	101
w/o All Concepts	7.80	3.68	7.21	107
w/o Augmented Concepts	22.39	15.90	16.91	99
w/o Claims	31.62	25.03	22.09	100
w/o Entities	32.11	25.36	22.42	101
Random Selection	12.96	8.25	10.05	103
Greedy Selection	32.33	25.60	22.53	100
Opinion Article Generation (NYT)				
RETRIEVAL	9.68	7.96	9.98	99
SENTPLANNER	7.45	5.06	6.62	106
SEQ2SEQ	21.44	14.92	14.93	119
SEQ2SEQFULL	31.06	29.74	23.10	121
DYPLOC (ours)	40.63	36.93	25.76	122
w/o All Concepts	11.32	6.01	8.33	132
w/o Augmented Concepts	26.94	21.56	18.39	117
w/o Claims	39.44	35.43	25.25	122
w/o Entities	39.66	35.82	25.11	122
Random Selection	5.32	5.29	6.00	72
Greedy Selection	40.61	36.88	25.77	122

Table 9.3: Automatic evaluation results on both tasks. We report BLEU-2, ROUGE-2, METEOR, and output length. Best scores are in bold. Our DYPLOC model statistically significantly outperforms all baselines and comparisons (randomization approximation test (Noreen, 1989), $p < 0.0005$).

9.3.3 Baselines and Comparisons

We compare with three baselines: (1) **RETRIEVAL** first calculates the TF-IDF weighted bag-of-words vectors for each content item, which is then used to query the training set sentences. The one with the highest cosine similarity is picked for each query, which are then ordered by a trained Pointer-Network (Vinyals et al., 2015a) as described in Gong et al. (2016). (2) **SENT-**

PLANNER (Hua and Wang, 2019) is an LSTM-based seq2seq model with a separate sentence planning decoder, where the planner selects keyphrases by using attentions and the generator reflects the selections. We treat our entities and concepts as keyphrases to feed to this model. (3) **SEQ2SEQ** is a fine-tuned BART model, whose input is the original content items *without augmentation*, thus does not have access to the predicted concepts and claims.

Additionally, we consider a strong comparison **SEQ2SEQFULL**, by fine-tuning BART with the same augmented content items as inputs as in our model. The difference is that the content items are concatenated before being used as input.

9.3.4 Reproducibility

We implement all models using the Huggingface Transformers library (Wolf et al., 2020) with PyTorch (Paszke et al., 2017). We use the base model for BART, which has 768 dimensional states and 6 layers for both encoder and decoder (140M parameters in total). Our newly added plan scoring network only contains 1.2M parameters, less than 1% of the pre-trained model. Our generation model is optimized using Adam (Kingma and Ba, 2015), with a batch size of 3. To improve efficiency, we adopt the mixed-precision (FP16) to train each model, using one NVIDIA Titan RTX GPU card with 24GB memory. The number of content items is limited to 10 per sample, and the numbers of entities and concepts per content item are capped at 20, respectively. We also truncate the target output to at most 200 tokens during training. Early stopping is applied over validation loss. Our model converges after being trained for 38 hours (19 epochs) on

CMV, and 45 hours (15 epochs) on NYT. The best validation perplexity reaches about 6.1 after model convergence on both datasets.

9.4 Results

9.4.1 Automatic Evaluation

Here we report results on test sets with standard automatic metrics: BLEU (Papineni et al., 2002) measures the n-gram precision (here we consider up to bigrams); ROUGE (Lin, 2004), calculated based on n-gram recall; and METEOR (Denkowski and Lavie, 2014), which also accounts for synonyms. In Table 9.3, we first present the results when gold-standard concept expansion is used.

Our proposed DYPLOC model achieves significantly higher performance across all metrics on both datasets. In particular, the substantial lead over SEQ2SEQFULL, which has access to the same content items as ours, indicates that *dynamic content planning with mixed language models produces superior generations*. Among comparison models, the gap between SEQ2SEQFULL and SEQ2SEQ shows the effectiveness of content item augmentation. We also observe a significant drop for baselines without using large models, highlighting the importance of pre-training.

Ablation Study. To verify the effect of each element in content items, we further train ablated models by removing concepts, claims, or entities. The results are also displayed in Table 9.3. In general, scores decrease when using only

	CMV			NYT		
	BLEU-2	ROUGE-2	METEOR	BLEU-2	ROUGE-2	METEOR
RETRIEVAL	8.30	4.39	9.64	8.85	6.64	9.21
SENTPLANNER	7.84	3.24	7.76	7.75	5.12	6.80
SEQ2SEQFULL	18.06	8.83	15.96	16.20	8.83	15.25
DYPLOC	22.84	11.83	17.13	24.54	15.46	17.41

Table 9.4: BLEU-2, ROUGE-2, and METEOR results on systems with predicted concepts as input.

partial content items, among which removing all concepts lead to the biggest performance drop, suggesting that entities and claims alone are insufficient to produce informative outputs.

Effect of Hard Selection of Content Items. To test the necessity of using weighted-sum marginalization (Eq. 9.2), we experiment with two comparisons with hard selections, i.e., either randomly choosing a content item, or using the one with the highest predicted plan score (greedy selection). For both cases, we set the selected content item’s plan score as 1.0, with the rest of the candidates having a score of 0.0, to ensure the probabilities summed up to 1.0. As can be seen from the bottom two rows of Table 9.3, not surprisingly, random selection performs much worse. We observe that its generations lack coherence and fluency, implying the effectiveness of our learnable content planner. On the other hand, using greedily selected content items obtains comparable results with DYPLOC, where a weighted sum of content items is considered. Indeed, we find that DYPLOC’s plan scores are often sharp where one content item has much higher weight than others, and in these scenarios, it is almost equivalent to the greedy selection setup.

Data	System	Gram.	Coh.	Rel.	Cont.	Top-1
CMV	SEQ2SEQ	4.19	3.12	3.19	2.89	25.1%
	SEQ2SEQFULL	4.24	3.19	3.23	3.13	30.2%
	DYPLOC	4.26	3.35	3.35	3.28	44.7%
NYT	SEQ2SEQ	4.38	3.82	4.20	4.01	25.2%
	SEQ2SEQFULL	4.48	3.99	4.30	4.14	28.9%
	DYPLOC	4.55	4.14	4.31	4.28	45.9%

Table 9.5: Human evaluation results on grammaticality (Gram.), relevance (Rel.), coherence (Coh.), and content richness (Cont.). For each sample, outputs by all three systems are ranked based on the overall preference. We show the percentage each system is ranked as the best.

Results with Generated Concepts. Table 9.4 lists generation results with our *system generated concepts* as expansion. While all systems yield worse results compared to using gold-standard concepts, our DYPLOC still outperforms other models by substantial margins, showing its *robustness when input concepts are noisy*. Yet it also suggests the importance of having more accurate and comprehensive concept expansion, which should be explored in the future work.

9.4.2 Human Evaluation

We hire three proficient English speakers to evaluate four key aspects of the generated outputs: (1) **grammaticality**; (2) **coherence**, measuring if the text is logical and cohesive; (3) **relevance**, gauging topic relatedness to the input title; and (4) **content richness**, assessing the specificity and whether there is enough details in the outputs. Each aspect is rated on a scale of 1 (worst) to 5 (best). In addition, judges also rank the system outputs by their overall preferences.

We randomly select 50 samples from the test sets for both tasks, and present outputs by SEQ2SEQ, SEQ2SEQFULL, and DYPLOC in random orders. Table 9.5

shows that DYPLOC receives higher scores across all aspects and tasks. In particular, the considerable difference in **coherence** and **content richness** indicates *our framework yields better content organization as well as retains more useful information*. Overall, our system outputs are ranked best for 44.7% and 45.9% of the time in two tasks, significantly more than the comparisons.

Analysis on Argumentative Quality. In the ablation study, we find that our full model’s performance is similar to the version without having claims as input. We suspect this is because claims are often paraphrased or even not directly used when delivering an argument, which cannot be captured by the automatic metrics. To better understand how claims are used for generation, we randomly select 50 examples by DYPLOC and its variant without claims, and ask the same human judges to decide whether there is a clear central argument conveyed by each generated argument on CMV.

We observe that 66.7% of the outputs by our full model are recognized as *successfully delivering arguments with consistent stances*, whereas only 61.3% are true for the model variant without claims. This gap confirms that claim drafts can indeed promote the argumentative quality as perceived by human readers.

9.5 Further Discussions

Evaluation results on generation quality have shown the effectiveness of our mixed language models. In this section, we aim to further understand the behavior of the plan scoring network, $d(\mathbf{x}|y_{<t})$, such as how it affects the usage of content items for generation. Specifically, we adopt the following procedure to

construct **alignment** between each sentence in the output and content items: for each token y_t , we establish a mapping $y_t \mapsto x_i$ if x_i is the most important item for producing y_t , i.e., $x_i = \operatorname{argmax}_x d(x|y_{<t})$, and $d(x_i|y_{<t}) > 0.5$. If all tokens in an entire sentence is mapped to the same x_i , we consider this sentence is aligned to that content item. Based on this rule, we show sample output and corresponding alignments in Figure 9.3.

Content Items	(a) [ENT] CO2 [CON] death, ensue, toll, staggering (b) [CON] leave, stop, compare, change, denialism, issue, simply, risk (c) [ENT] Fossil_fuel [CON] drive, paralyze, deal, humanity, industry [CLAIM] Coal is not a reliable source of energy, and it's been driven by unreliable sources of energy and unreliable sources to date. (d) [CON] win, consequence, remotely, dire
DYPLOC	Climate change denial is driven by fossil fuel industries that are paralyzing the humanities in attempting to deal with the problem. The death toll from ensuing CO2 is staggering. There is no winning consequence for anything remotely dire that climate change denialism has. Leaving aside the issue of GM risk, simply comparing climate change denialism to climate change protesters is not going to stop anything.
SEQ2SEQFULL	World wide civilisation is threatened with catastrophe by climate change denial. Climate change denial is driven by the fossil fuel industry, which is paralyzed by the humanities. Any attempt to deal with the problem of climate change is going to cause death tolls that ensue, staggering. Leaving aside the issue of GM, the risk is simply too much to compare.
Content Items	(a) [ENT] President_of.the.U.S., George.W.Bush, Richard_Nixon [CON] omit, list (b) [ENT] George.W.Bush, [CON] bring, people, world, honor, dignity, respect (c) [ENT] Bill.Clinton, Ronald.Reagan [CON] harm, respect, demonstrate, damage, office, state (d) [CON] owe, correction, omission, apology, readers
DYPLOC	President Clinton "has done more harm to the respect of the Presidency than Presidents Presidents Ford, Carter and Reagan have demonstrated," and that "the failure of Presidents Clinton and Reagan has caused permanent damage to the office." You omit President Bush from your list of Presidents. Mr. Bush has brought honor, dignity and respect to the office of the Presidency in the eyes of the American people and the world at large. We owe your readers an apology and a correction of your editorial's glaring omission, which created an erroneous impression.
SEQ2SEQFULL	"The Picture of Bill Clinton" (editorial, Dec. 5) states that President Clinton "has done more harm to our respect for the Presidency than Presidents Reagan, Ford, Carter and Bush." This demonstrates that President Reagan's failure to do more damage to our honor than President Bush's failure in office. You omitted from your list President Clinton's achievements that brought honor and dignity to the eyes of the American people [...]

Figure 9.3: Sample generations on CMV [Upper] and NYT [Lower]. System generated concepts and claims are in *italics*. For DYPLOC, we highlight sentence to content item alignment using colors.

For the rest of this section, we conduct analyses based on this alignment result. We first examine whether the model learns to utilize enough content items, i.e., high coverage. Then we provide insights on whether the generation

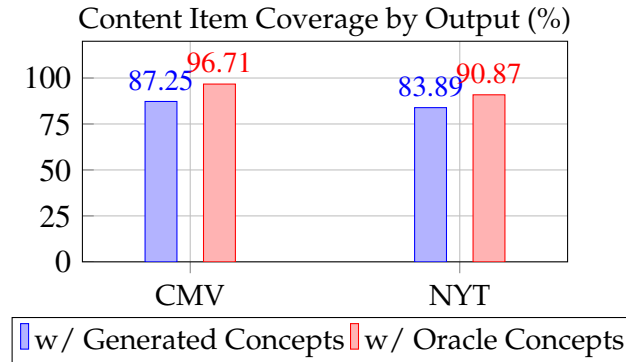


Figure 9.4: The percentage of content items that are aligned to at least one output sentence.

faithfully reflects the argumentative claims using entailment relation labeling by human inspection.

How many content items are used by the output? Human judges have rated our model output to contain more relevant information (Table 9.5). We believe this can be attributed to the enhanced capacity to access and reflect the input data with dynamic content planning, as a result of mixed language models. To verify this hypothesis, we calculate the percentage of content items that are aligned to at least one output sentence. Figure 9.4 shows that, using our system, the coverage reaches over 87.25% on CMV and 83.89% for NYT. If we replace the generated concepts with gold-standard concepts (as extracted from references) instead, the coverage exceeds 90% on both tasks. These observations indicate that *our model can indeed adequately utilize the input data, with more accurate concepts further encouraging higher coverage.*

How are claim content items realized? Claims are the central elements for opinion text construction. As mentioned in Section 9.3.2, a subset of the content items are supplied with claim sentences. In order to examine whether they are

realized as claim sentences in the outputs, we leverage the fine-tuned BERT classifier (Section 9.3.2) to label all output sentences. 90.96% of the sentences that are aligned to a claim element in the input are also labeled as `claim` on CMV. The percentage is only 69.41% for NYT, though, likely because the NYT opinion articles still contain more objective information.

Furthermore, we conduct a human evaluation study to assess the semantic relations between claim input and its aligned generated sentence. We randomly sample 50 outputs from test sets, and ask four human judges to read each. For each sample, we highlight one output sentence that is aligned to a content item with claim element. The judges determine a three-way (ENTAIL, NEUTRAL, CONTRADICTION) entailment relation between the input claim (premise) and the output (hypothesis). Results show that ENTAIL accounts for 49.06% of all instances, while only 3.77% are deemed CONTRADICTION. Upon inspection, the contradictory pairs are usually disagreements with regard to implicit sentiments, e.g., *“Journalist is the most responsible for the problem”* vs. *“Media coverage is a good thing.”*. This suggests that while our conditional language model achieves reasonable semantic control in most cases, it is still not guaranteed to capture more nuanced semantics encoded in opinions and arguments. Future work includes designing representations that can better model stances in opinions as well as argumentative structures.

9.6 Conclusion

We present a novel text generation framework that enables dynamic content planning based on mixed conditional language models. We further employ

large models to augment system inputs with diverse content that covers both objective and subjective information. The experiments on two distinct opinion text generation tasks show that our proposed model compares favorably against strong comparisons based on fine-tuned BART models with the same input. Human evaluation further confirms that our model generations have richer information and better content organization.

CHAPTER 10

BIAS AND ETHICAL CONSIDERATIONS

Online forums such as Reddit are known to present various types of biases, which are further amplified by neural language models. In this chapter, we first mention the dataset biases caused by the topic distribution and user demographics (Section 10.1), followed by the ethical concerns regarding discriminative and toxic content (Section 10.2). Finally, we discuss the risk of unfaithful generation and potential misuse of our model to spread misinformation (Section 10.3).

10.1 Potential Biases in the Dataset

In this dissertation, the argument generation models are all trained over the Reddit ChangeMyView corpus, as it is one of the few publicly available large datasets with diverse arguments. However, Reddit is known to encode various biases such as the word usage (Ferrer et al., 2020), popular topics being discussed (Soliman et al., 2019), and user demographics (Duggan and Smith, 2013; Gjurković et al., 2020). For training text generation models in general, limited exposures to different topics and language style harm the model’s generalizability. In our work, we aim to mitigate this issue by introducing massive and generic external resources, such as Wikipedia pages and news articles (Chapter 5, Chapter 6). We make them accessible to the generation pipeline so that the relevant information can be utilized even for unseen topics. But since the training and test set are both sampled from the CMV subreddit, it is unclear whether our model can readily adapt to new domains.

For our argument generation task, the inclusion of diverse perspectives for a given topic is particularly important. For example, to counter-argue the input argument “*death penalty is necessary.*”, one could emphasize the risk of *wrongful conviction* and that *death penalty is irreversible*. Alternatively, others might point out that *death penalty does not deter crimes* and therefore is unnecessary. Users with different backgrounds, such as their ethnicity, political and religious ideology, tend to adopt different perspectives, which consequently affects the community endorsement towards the arguments they make (Durmus and Cardie, 2019). The study by Duggan and Smith (2013) reveals that young males who reside in urban area are by far more likely to use Reddit. Unfortunately, the anonymous nature of Reddit makes it challenging to quantify such overrepresentation for specific subcommunity. We believe more work needs to be done to automatically categorize the argument perspectives in training data to minimize the dataset bias.

10.2 Ethical Concerns

Another risk associated with the online user-generated data is the abundance of offensive and discriminative content (Kwok and Wang, 2013; Wang et al., 2014). Generation models trained over such data often exhibit these undesired languages as well. Model deployment in real world can reach millions of users instantly, making it unrealistic to manually filter the undesired languages. To date, the automatic detection of toxic language has proved challenging (Djuric et al., 2015; Burnap and Williams, 2015; Malmasi and Zampieri, 2018). Simply filtering posts using lexicon-based rules is insufficient. Because “cursing words” are frequently used in social media, and that the implicit harmful content re-

quires deeper understanding of the semantics and social backgrounds (Davidson et al., 2017). Another line of work aims to curb the generation model to avoid generating the undesired language (Nogueira dos Santos et al., 2018; Tran et al., 2020; Gehman et al., 2020), only with limited successes. In future work, we could explore categorizing offensive content and studying the context that triggers them, to design more accurate detection models and better algorithms to avoid generating them.

10.3 Unfaithful Generation and Misinformation

A major weakness of neural generation models is their tendency to produce unfaithful output. Prior work has reported such behavior in numerous generation tasks (Wiseman et al., 2017; Weng et al., 2020; Maynez et al., 2020). The root cause is likely due to the token-level optimization which does not account for the content selection and semantic representation. One theme of our work in Chapter 7, Chapter 8, and Chapter 9 is to enable better controllability over the content. But our systems still have limitations such as their insensitivity towards numerical expressions and rare named entities. This is problematic in practice especially for high-stake applications, such as financial and legal reports. We believe a crucial future step would be developing better automatic evaluation metrics that can capture factual errors beyond lexical overlap. Additionally, generation models that are grounded on symbolic knowledge-graph would be another direction to mitigate such issues.

Recently, the NLP community has become more aware of the threats of fake news and intentional misinformation spread by text generation models (Shao

et al., 2017; Pérez-Rosas et al., 2018; Fernandez and Alani, 2018; Zellers et al., 2019; Oshikawa et al., 2020). Studies have shown that social media bots have been deployed to interfere with the US election (Bessi and Ferrara, 2016; Grinberg et al., 2019; Bovet and Makse, 2019) as well as spreading rumors during the COVID-19 pandemic (Yang et al., 2020; Ferrara, 2020; Uyheng and Carley, 2020). We recognize that our controllable text generation models can be potentially misused. We therefore urge cautious examination of the ethical implications and responsible use in real world applications.

CHAPTER 11

CONCLUSION AND FUTURE DIRECTIONS

11.1 Conclusion

In this dissertation, we present various studies on argument mining, argument generation, and controllable text generation in general. We motivate this line of research with the ubiquity and impact of argumentation in the real world, and highlight the major challenges of both the argument understanding itself and the text generation stack.

In Chapter 3 and Chapter 4, we investigate the argument component extraction, classification, and the support relation identification. We start with arguments in the online debate forum and explore the effect of different feature sets for the above tasks. Then we study the much more nuanced peer review domain, and benchmark state-of-the-art neural models. Our findings consolidate the importance of diversity of argument composition, and characterize their interactions with the supporting arguments.

In Chapter 5 and Chapter 6, we introduce two neural argument generation pipelines. Our model design is enabled by two key components: (1) a knowledge retrieval system that indexes a large set of existing arguments from diverse sources, and (2) a neural generation model that selects the most relevant retrieval results for realization. We showcase our systems over the Reddit ChangeMyView dataset, which we also release to facilitate future research on argument generation.

In Chapter 7, we demonstrate that our text generation system can be gen-

eralized to new domains. Our main innovation is to integrate a sentence-level content selection component into the neural decoder, which is also end-to-end trainable. Apart from argument generation, we evaluate over Wikipedia introduction generation and paper abstract generation. Our model marked better performance than strong comparisons in both settings. Interestingly, our analysis has revealed a strong positive correlation between the accuracy of keyphrase selection and the overall output quality, which quantitatively shows the impact of text planning.

In Chapter 8 and Chapter 9, we discuss two controllable generation framework tailored for large pre-trained Transformers. The PAIR system repurposes the masked language model as template representation with fine-grained control on content. The iterative refinement strategy operates seamlessly with this framework to generate coherent natural language output. On the other hand, DYPLOC offers an alternative solution to content ordering and realization, which leverages the mixed language model. Its content selection operations are also easily interpretable, to better understand the system decisions.

Throughout this dissertation, we also witness the revolution of pre-trained Transformers in NLP community. In particular, the BART (Lewis et al., 2020a) and GPT-2 (Radford et al., 2019) model, which can be natively used for generation tasks, have attained unparalleled performance in terms of the output fluency. Although it is not our focus to compare the Transformers and LSTM-based models, we note a key difference that affects the model design: the typical method to condition the LSTM model is through the state transformation (Hoang et al., 2016), whereas the Transformers rely on self-attention over explicit natural language prompts (Keskar et al., 2019). This property motivates

our PAIR model, which utilizes the masked keyphrase templates to achieve fine-grained content control without hurting the output fluency.

11.2 Future Directions

Throughout the development of this dissertation, we have witnessed the rapid growth of argument related research in NLP community. There have been numerous argument mining workshops, and dedicated tracks for the major NLP conferences. This not only produces more diverse solutions to existing problems, but also identifies numerous new challenges.

First, understanding argumentative structures is still hard, yet it is necessary in order to inform the generation model of the proper content organization. Existing work on student essays (Stab and Gurevych, 2017; Persing and Ng, 2016) and online user comments (Niculae et al., 2017) have shown promising results, but both rely heavily on feature-engineering. One key issue lies in the difficulty to collect high-quality annotations, especially for domains that require expert knowledge. The data scarcity problem limits the utilization of neural network models. We believe it is imperative to develop more data efficient models and training techniques. For example, transfer learning has the potential to adapt useful representations learned through a large labeled source domain to a more difficult domain with less annotations. The active learning strategies can further help significantly reduce the annotation requirement by automatically selecting the informative samples. Both techniques are being investigated in our upcoming manuscripts.

Moreover, the pre-trained Transformers have revolutionized text generation

research. Models such as BART (Lewis et al., 2020a) and GPT-2 (Radford et al., 2019) are capable to produce long documents with almost human-level fluency. However, despite efforts (Keskar et al., 2019; Dathathri et al., 2020; Hua and Wang, 2020) to gain more control over these large models, how to ensure the output factuality is still an open question. Recent work has found that the models pre-trained over large heterogeneous data tend to amplify the biases towards marginalized groups (Sheng et al., 2019; McGuffie and Newhouse, 2020), and they can be easily rewired to create false and hateful content (Wallace et al., 2019). To mitigate the above issues and ensure more ethical use of large models, we need both methods to detect false and biased content, as well as knowledge-grounded systems that can inform and constrain the neural language model.

Finally, the ultimate judges of text generation systems are the human users. Collecting meaningful user feedback at scale is time-consuming and involves added ethical concerns. As a result, the current research in NLG mostly focuses on improving over automatic metrics. We believe more work needs to be done to understand the human-factors in the generation pipeline, e.g., to assist essay or news article writing, what type of information should be automated and what should be left to the user; how to present the system output in an explainable way. Conversely, we are also interested to learn what neural generators can tell us about human cognition.

BIBLIOGRAPHY

- Yamen Ajjour, Wei-Fan Chen, Johannes Kiesel, Henning Wachsmuth, and Benno Stein. 2017. [Unit segmentation of argumentative texts](#). In *Proceedings of the 4th Workshop on Argument Mining*, pages 118–128, Copenhagen, Denmark. Association for Computational Linguistics.
- Khalid Al-Khatib, Henning Wachsmuth, Johannes Kiesel, Matthias Hagen, and Benno Stein. 2016. [A news editorial corpus for mining argumentation strategies](#). In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3433–3443, Osaka, Japan. The COLING 2016 Organizing Committee.
- Steffen Albrecht. 2006. Whose voice is heard in online deliberation?: A study of participation and representation in political debates on the internet. *Information, Community and Society*, 9(1):62–82.
- Waleed Ammar, Dirk Groeneveld, Chandra Bhagavatula, Iz Beltagy, Miles Crawford, Doug Downey, Jason Dunkelberger, Ahmed Elgohary, Sergey Feldman, Vu Ha, Rodney Kinney, Sebastian Kohlmeier, Kyle Lo, Tyler Murray, Hsu-Han Ooi, Matthew Peters, Joanna Power, Sam Skjonsberg, Lucy Wang, Chris Wilhelm, Zheng Yuan, Madeleine van Zuylen, and Oren Etzioni. 2018. [Construction of the literature graph in semantic scholar](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*, pages 84–91, New Orleans - Louisiana. Association for Computational Linguistics.
- Gabor Angeli, Percy Liang, and Dan Klein. 2010. [A simple domain-independent probabilistic approach to generation](#). In *Proceedings of the 2010 Conference on*

- Empirical Methods in Natural Language Processing*, pages 502–512, Cambridge, MA. Association for Computational Linguistics.
- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Anusha Balakrishnan, Jinfeng Rao, Kartikeya Upasani, Michael White, and Rajen Subba. 2019. [Constrained decoding for neural NLG from compositional representations in task-oriented dialogue](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 831–844, Florence, Italy. Association for Computational Linguistics.
- Roy Bar-Haim, Indrajit Bhattacharya, Francesco Dinuzzo, Amrita Saha, and Noam Slonim. 2017. [Stance classification of context-dependent claims](#). In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 251–261, Valencia, Spain. Association for Computational Linguistics.
- Regina Barzilay and Mirella Lapata. 2005. [Collective content selection for concept-to-text generation](#). In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 331–338, Vancouver, British Columbia, Canada. Association for Computational Linguistics.
- Regina Barzilay and Lillian Lee. 2004. [Catching the drift: Probabilistic content models, with applications to generation and summarization](#). In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 113–120, Boston, Massachusetts, USA. Association for Computational Linguistics.

- Anja Belz. 2008. Automatic generation of weather forecast texts using comprehensive probabilistic generation-space models. *Natural Language Engineering*, 14(4):431–455.
- Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155.
- Alessandro Bessi and Emilio Ferrara. 2016. Social bots distort the 2016 us presidential election online discussion. *First Monday*, 21(11-7).
- Bin Bi, Chen Wu, Ming Yan, Wei Wang, Jiangnan Xia, and Chenliang Li. 2019. [Incorporating external knowledge into machine reading for generative question answering](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2521–2530, Hong Kong, China. Association for Computational Linguistics.
- Or Biran and Owen Rambow. 2011. Identifying justifications in written dialogs by classifying text as argumentative. *International Journal of Semantic Computing*, 5(04):363–381.
- Mathieu Blondel and Fabian Pedregosa. 2016. Lightning: large-scale linear classification, regression and ranking in python.
- Blai Bonet and Hector Geffner. 1996. Arguing for decisions: A qualitative model of decision making. In *Proceedings of the Twelfth international conference on Uncertainty in artificial intelligence*, pages 98–105. Morgan Kaufmann Publishers Inc.

- Nadjet Bouayad-Agha, Gerard Casamayor, and Leo Wanner. 2011. [Content selection from an ontology-based knowledge base for the generation of football summaries](#). In *Proceedings of the 13th European Workshop on Natural Language Generation*, pages 72–81, Nancy, France. Association for Computational Linguistics.
- Alexandre Bovet and Hernán A Makse. 2019. Influence of fake news in twitter during the 2016 us presidential election. *Nature communications*, 10(1):1–14.
- Marc Brysbaert, Amy Beth Warriner, and Victor Kuperman. 2014. Concreteness ratings for 40 thousand generally known english word lemmas. *Behavior research methods*, 46(3):904–911.
- Christopher JC Burges. 2010. From ranknet to lambdarank to lambdamart: An overview. *Learning*, 11(23-581):81.
- Pete Burnap and Matthew L Williams. 2015. Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making. *Policy & internet*, 7(2):223–242.
- James P Byrnes. 2013. *The nature and development of decision-making: A self-regulation model*. Psychology Press.
- Elena Cabrio and Serena Villata. 2018. Five years of argument mining: a data-driven analysis. In *IJCAI*, volume 18, pages 5427–5433.
- Charles B. Callaway. 2003. [Integrating discourse markers into a pipelined natural language generation architecture](#). In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 264–271, Sapporo, Japan. Association for Computational Linguistics.

- Giuseppe Carenini and Johanna Moore. 2000. [A strategy for generating evaluative arguments](#). In *INLG'2000 Proceedings of the First International Conference on Natural Language Generation*, pages 47–54, Mitzpe Ramon, Israel. Association for Computational Linguistics.
- Lynn Carlson, Daniel Marcu, and Mary Ellen Okurovsky. 2001. [Building a discourse-tagged corpus in the framework of Rhetorical Structure Theory](#). In *Proceedings of the Second SIGdial Workshop on Discourse and Dialogue*.
- Thiago Castro Ferreira, Chris van der Lee, Emiel van Miltenburg, and Emiel Krahmer. 2019. [Neural data-to-text generation: A comparison between pipeline and end-to-end architectures](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 552–562, Hong Kong, China. Association for Computational Linguistics.
- Tuhin Chakrabarty, Christopher Hidey, Smaranda Muresan, Kathy McKeown, and Alyssa Hwang. 2019. [AMPERSAND: Argument mining for PERSuAsive oNline discussions](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2933–2943, Hong Kong, China. Association for Computational Linguistics.
- Olivier Chapelle and Yi Chang. 2011. [Yahoo! learning to rank challenge overview](#). In *Yahoo! Learning to Rank Challenge*, pages 1–24.
- Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. [Reading Wikipedia to answer open-domain questions](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long*

- Papers*), pages 1870–1879, Vancouver, Canada. Association for Computational Linguistics.
- Nancy Chinchor and Patricia Robinson. 1997. Muc-7 named entity task definition. In *Proceedings of the 7th Conference on Message Understanding*, volume 29.
- Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.
- Elizabeth Clark, Yangfeng Ji, and Noah A. Smith. 2018. [Neural text generation in stories using entity representations as context](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2250–2260, New Orleans, Louisiana. Association for Computational Linguistics.
- Emilie Colin, Claire Gardent, Yassine M’rabet, Shashi Narayan, and Laura Perez-Beltrachini. 2016. [The WebNLG challenge: Generating text from DB-Pedia data](#). In *Proceedings of the 9th International Natural Language Generation conference*, pages 163–167, Edinburgh, UK. Association for Computational Linguistics.
- T Edward Damer. 2012. *Attacking faulty reasoning*. Cengage Learning.
- Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2020. [Plug and play language models: A simple approach to controlled text generation](#). In *International Conference on Learning Representations*.

- Thomas Davidson, Dana Warmusley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the 11th International AAAI Conference on Web and Social Media, ICWSM '17*, pages 512–515.
- Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. 2014. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in neural information processing systems*, pages 1646–1654.
- Michael Denkowski and Alon Lavie. 2014. [Meteor universal: Language specific translation evaluation for any target language](#). In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 376–380, Baltimore, Maryland, USA. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Nemanja Djuric, Jing Zhou, Robin Morris, Mihajlo Grbovic, Vladan Radosavljevic, and Narayan Bhamidipati. 2015. Hate speech detection with comment embeddings. In *Proceedings of the 24th international conference on world wide web*, pages 29–30.
- Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-

- training for natural language understanding and generation. In *Advances in Neural Information Processing Systems*, pages 13042–13054.
- Pablo A. Duboue and Kathleen R. McKeown. 2001. [Empirically estimating order constraints for content planning in generation](#). In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 172–179, Toulouse, France. Association for Computational Linguistics.
- Pablo Ariel Duboue and Kathleen R. McKeown. 2003. [Statistical acquisition of content selection rules for natural language generation](#). In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pages 121–128.
- John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159.
- Maeve Duggan and Aaron Smith. 2013. 6% of online adults are reddit users. *Pew Internet & American Life Project*, 3:1–10.
- Esin Durmus and Claire Cardie. 2019. [A corpus for modeling user and language effects in argumentation on online debating](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 602–607, Florence, Italy. Association for Computational Linguistics.
- Ondřej Dušek and Filip Jurčiček. 2016. [Sequence-to-sequence generation for spoken dialogue via deep syntax trees and strings](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 45–51, Berlin, Germany. Association for Computational Linguistics.

- Ondřej Dušek, Jekaterina Novikova, and Verena Rieser. 2018. [Findings of the E2E NLG challenge](#). In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 322–328, Tilburg University, The Netherlands. Association for Computational Linguistics.
- Rory Duthie, Katarzyna Budzynska, and Chris Reed. 2016. Mining ethos in political debate. In *Computational Models of Argument: Proceedings from the Sixth International Conference on Computational Models of Argument (COMMA)*, pages 299–310. IOS Press.
- Judith Eckle-Kohler, Roland Kluge, and Iryna Gurevych. 2015. [On the role of discourse markers for discriminating claims and premises in argumentative discourse](#). In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2236–2242, Lisbon, Portugal. Association for Computational Linguistics.
- Steffen Eger, Johannes Daxenberger, and Iryna Gurevych. 2017a. Neural end-to-end learning for computational argumentation mining. *arXiv preprint arXiv:1704.06104*.
- Steffen Eger, Johannes Daxenberger, and Iryna Gurevych. 2017b. [Neural end-to-end learning for computational argumentation mining](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11–22, Vancouver, Canada. Association for Computational Linguistics.
- Henry Elder, Sebastian Gehrmann, Alexander O’Connor, and Qun Liu. 2018. [E2E NLG challenge submission: Towards controllable generation of diverse natural language](#). In *Proceedings of the 11th International Conference on Natu-*

- ral Language Generation*, pages 457–462, Tilburg University, The Netherlands. Association for Computational Linguistics.
- Angela Fan, Claire Gardent, Chloé Braud, and Antoine Bordes. 2019a. [Using local knowledge graph construction to scale Seq2Seq models to multi-document inputs](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4186–4196, Hong Kong, China. Association for Computational Linguistics.
- Angela Fan, David Grangier, and Michael Auli. 2018a. [Controllable abstractive summarization](#). In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 45–54, Melbourne, Australia. Association for Computational Linguistics.
- Angela Fan, Mike Lewis, and Yann Dauphin. 2018b. [Hierarchical neural story generation](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.
- Angela Fan, Mike Lewis, and Yann Dauphin. 2019b. [Strategies for structuring story generation](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2650–2660, Florence, Italy. Association for Computational Linguistics.
- Vanessa Wei Feng and Graeme Hirst. 2011. [Classifying arguments by scheme](#). In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 987–996, Portland, Oregon, USA. Association for Computational Linguistics.

- Miriam Fernandez and Harith Alani. 2018. Online misinformation: Challenges and future directions. In *Companion Proceedings of the The Web Conference 2018*, pages 595–602.
- Emilio Ferrara. 2020. # covid-19 on twitter: Bots, conspiracies, and social media activism. *arXiv preprint arXiv: 2004.09531*.
- Xavier Ferrer, Tom van Nuenen, Jose M Such, and Natalia Criado. 2020. Discovering and categorising language biases in reddit. *arXiv preprint arXiv:2008.02754*.
- Jessica Fidler and Yoav Goldberg. 2017. [Controlling linguistic style aspects in neural language generation](#). In *Proceedings of the Workshop on Stylistic Variation*, pages 94–104, Copenhagen, Denmark. Association for Computational Linguistics.
- Markus Freitag, Isaac Caswell, and Scott Roy. 2019. [APE at scale and its implications on MT evaluation biases](#). In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 34–44, Florence, Italy. Association for Computational Linguistics.
- Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2018. Style transfer in text: Exploration and evaluation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.
- Yarin Gal and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks. In *Advances in neural information processing systems*, pages 1019–1027.
- Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. 2020. [RealToxicityPrompts: Evaluating neural toxic degeneration in](#)

- language models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3356–3369, Online. Association for Computational Linguistics.
- Sebastian Gehrmann, Falcon Dai, Henry Elder, and Alexander Rush. 2018. [End-to-end content and plan selection for data-to-text generation](#). In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 46–56, Tilburg University, The Netherlands. Association for Computational Linguistics.
- Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. 2019. [Mask-predict: Parallel decoding of conditional masked language models](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6112–6121, Hong Kong, China. Association for Computational Linguistics.
- Matej Gjurković, Mladen Karan, Iva Vukojević, Mihaela Bošnjak, and Jan Šnajder. 2020. Pandora talks: Personality and demographics on reddit. *arXiv preprint arXiv:2004.04460*.
- Seraphina Goldfarb-Tarrant, Tuhin Chakrabarty, Ralph Weischedel, and Nanyun Peng. 2020. [Content planning for neural story generation with aristotelian rescoring](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4319–4338, Online. Association for Computational Linguistics.
- Jingjing Gong, Xinchu Chen, Xipeng Qiu, and Xuanjing Huang. 2016. End-to-end neural sentence ordering using pointer network. *arXiv preprint arXiv:1611.04953*.

- Clinton Gormley and Zachary Tong. 2015. *Elasticsearch: The definitive guide: A distributed real-time search and analytics engine.* " O'Reilly Media, Inc."
- Tanya Goyal and Greg Durrett. 2020. [Neural syntactic preordering for controlled paraphrase generation](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 238–252, Online. Association for Computational Linguistics.
- Nir Grinberg, Kenneth Joseph, Lisa Friedland, Briony Swire-Thompson, and David Lazer. 2019. Fake news on twitter during the 2016 us presidential election. *Science*, 363(6425):374–378.
- Brigitte Grote and Manfred Stede. 1998. [Discourse marker choice in sentence planning](#). In *Natural Language Generation*.
- Jian Guan, Fei Huang, Zhihao Zhao, Xiaoyan Zhu, and Minlie Huang. 2020. [A knowledge-enhanced pretraining model for commonsense story generation](#). *Transactions of the Association for Computational Linguistics*, 8:93–108.
- Beliz Gunel, Chenguang Zhu, Michael Zeng, and Xuedong Huang. 2020. Mind the facts: Knowledge-boosted coherent abstractive text summarization. *arXiv preprint arXiv:2006.15435*.
- Kelvin Guu, Tatsunori B. Hashimoto, Yonatan Oren, and Percy Liang. 2018. [Generating sentences by editing prototypes](#). *Transactions of the Association for Computational Linguistics*, 6:437–450.
- Ivan Habernal and Iryna Gurevych. 2015. [Exploiting debate portals for semi-supervised argumentation mining in user-generated web discourse](#). In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Process-*

ing, pages 2127–2137, Lisbon, Portugal. Association for Computational Linguistics.

Ivan Habernal and Iryna Gurevych. 2016. [Which argument is more convincing? analyzing and predicting convincingsness of web arguments using bidirectional LSTM](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1589–1599, Berlin, Germany. Association for Computational Linguistics.

Ivan Habernal and Iryna Gurevych. 2017. [Argumentation mining in user-generated web discourse](#). *Computational Linguistics*, 43(1):125–179.

Hamza Harkous, Isabel Groves, and Amir Saffari. 2020. [Have your text and use it too! end-to-end neural data-to-text generation with semantic fidelity](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2410–2424, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Seth Hettich and Michael J Pazzani. 2006. Mining for proposal reviewers: lessons learned at the national science foundation. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 862–871. ACM.

Christopher Hidey and Kathy McKeown. 2019. [Fixed that for you: Generating contrastive claims with semantic edits](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1756–1767, Minneapolis, Minnesota. Association for Computational Linguistics.

Christopher Hidey, Elena Musi, Alyssa Hwang, Smaranda Muresan, and Kathy

- McKeown. 2017. [Analyzing the semantic types of claims and premises in an online persuasive forum](#). In *Proceedings of the 4th Workshop on Argument Mining*, pages 11–21, Copenhagen, Denmark. Association for Computational Linguistics.
- Cong Duy Vu Hoang, Trevor Cohn, and Gholamreza Haffari. 2016. [Incorporating side information into recurrent neural network language models](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1250–1255, San Diego, California. Association for Computational Linguistics.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.
- Chris Hokamp and Qun Liu. 2017. [Lexically constrained decoding for sequence generation using grid beam search](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1535–1546, Vancouver, Canada. Association for Computational Linguistics.
- Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. In *International Conference on Learning Representations*.
- Eduard H Hovy. 1993. Automated discourse generation using discourse structure relations. *Artificial intelligence*, 63(1-2):341–385.
- J. Edward Hu, Huda Khayrallah, Ryan Culkin, Patrick Xia, Tongfei Chen, Matt Post, and Benjamin Van Durme. 2019. [Improved lexically constrained decoding for translation and monolingual rewriting](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Lin-*

- guistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 839–850, Minneapolis, Minnesota. Association for Computational Linguistics.
- Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177. ACM.
- Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. 2017. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1587–1596. JMLR.org.
- Xinyu Hua, Zhe Hu, and Lu Wang. 2019a. [Argument generation with retrieval, planning, and realization](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2661–2672, Florence, Italy. Association for Computational Linguistics.
- Xinyu Hua, Mitko Nikolov, Nikhil Badugu, and Lu Wang. 2019b. [Argument mining for understanding peer reviews](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2131–2137, Minneapolis, Minnesota. Association for Computational Linguistics.
- Xinyu Hua, Ashwin Sreevatsa, and Lu Wang. 2021. Dyploc: Dynamic planning of content using mixed language models for text generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*. Association for Computational Linguistics.
- Xinyu Hua and Lu Wang. 2017. [Understanding and detecting supporting ar-](#)

- guments of diverse types. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 203–208, Vancouver, Canada. Association for Computational Linguistics.
- Xinyu Hua and Lu Wang. 2018. [Neural argument generation augmented with externally retrieved evidence](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 219–230, Melbourne, Australia. Association for Computational Linguistics.
- Xinyu Hua and Lu Wang. 2019. [Sentence-level content planning and style specification for neural text generation](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 591–602, Hong Kong, China. Association for Computational Linguistics.
- Xinyu Hua and Lu Wang. 2020. [PAIR: Planning and iterative refinement in pre-trained transformers for long text generation](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 781–793, Online. Association for Computational Linguistics.
- Luyang Huang, Lingfei Wu, and Lu Wang. 2020. [Knowledge graph-augmented abstractive summarization with semantic-driven cloze reward](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5094–5107, Online. Association for Computational Linguistics.
- Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 2333–2338. ACM.

- Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. *arXiv preprint arXiv:1804.06059*.
- Yangfeng Ji and Jacob Eisenstein. 2014. [Representation learning for text-level discourse parsing](#). In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13–24, Baltimore, Maryland. Association for Computational Linguistics.
- Yangfeng Ji, Gholamreza Haffari, and Jacob Eisenstein. 2016. [A latent variable recurrent neural network for discourse-driven language models](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 332–342, San Diego, California. Association for Computational Linguistics.
- Dongyeop Kang, Waleed Ammar, Bhavana Dalvi, Madeleine van Zuylen, Sebastian Kohlmeier, Eduard Hovy, and Roy Schwartz. 2018. [A dataset of peer reviews \(PeerRead\): Collection, insights and NLP applications](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1647–1661, New Orleans, Louisiana. Association for Computational Linguistics.
- Jungo Kasai, James Cross, Marjan Ghazvininejad, and Jiatao Gu. 2020. [Non-autoregressive machine translation with disentangled context transformer](#). In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5144–5155. PMLR.
- Jacalyn Kelly, Tara Sadeghieh, and Khosrow Adeli. 2014. Peer review in scientific publications: benefits, critiques, & a survival guide. *EJIFCC*, 25(3):227.

- Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*.
- Yoon Kim. 2014. [Convolutional neural networks for sentence classification](#). In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.
- Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Ryan Kiros, Ruslan Salakhutdinov, and Rich Zemel. 2014. [Multimodal neural language models](#). In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 595–603, Beijing, China. PMLR.
- Wei-Jen Ko and Junyi Jessy Li. 2020. [Assessing discourse relations in language generation from GPT-2](#). In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 52–59, Dublin, Ireland. Association for Computational Linguistics.
- Alexandros Komninos and Suresh Manandhar. 2016. [Dependency based embeddings for sentence classification tasks](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1490–1500, San Diego, California. Association for Computational Linguistics.
- Rik Koncel-Kedziorski, Dhanush Bekal, Yi Luan, Mirella Lapata, and Hannaneh

- Hajishirzi. 2019. [Text Generation from Knowledge Graphs with Graph Transformers](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2284–2293, Minneapolis, Minnesota. Association for Computational Linguistics.
- Michail Kovanis, Raphaël Porcher, Philippe Ravaud, and Ludovic Trinquart. 2016. The global burden of journal peer review in the biomedical literature: Strong imbalance in the collective enterprise. *PLoS One*, 11(11):e0166387.
- Klaus Krippendorff. 2004. Measuring the reliability of qualitative text analysis data. *Quality and Quantity*, 38:787–800.
- Irene Kwok and Yuzhou Wang. 2013. Locate the hate: Detecting tweets against blacks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 27.
- John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.
- Anne Lauscher, Goran Glavaš, and Simone Paolo Ponzetto. 2018. [An argument-annotated corpus of scientific publications](#). In *Proceedings of the 5th Workshop on Argument Mining*, pages 40–46, Brussels, Belgium. Association for Computational Linguistics.
- Benoit Lavoie and Owen Rainbow. 1997. [A fast and portable realizer for text generation systems](#). In *Fifth Conference on Applied Natural Language Processing*, pages 265–268, Washington, DC, USA. Association for Computational Linguistics.

- Carolin Lawrence, Bhushan Kotnis, and Mathias Niepert. 2019. [Attending to future tokens for bidirectional sequence generation](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1–10, Hong Kong, China. Association for Computational Linguistics.
- John Lawrence and Chris Reed. 2019. [Argument mining: A survey](#). *Computational Linguistics*, 45(4):765–818.
- Dieu Thu Le, Cam-Tu Nguyen, and Kim Anh Nguyen. 2018. [Dave the debater: a retrieval-based and generative argumentative dialogue agent](#). In *Proceedings of the 5th Workshop on Argument Mining*, pages 121–130, Brussels, Belgium. Association for Computational Linguistics.
- Rémi Lebret, David Grangier, and Michael Auli. 2016. [Neural text generation from structured data with application to the biography domain](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1203–1213, Austin, Texas. Association for Computational Linguistics.
- Jason Lee, Elman Mansimov, and Kyunghyun Cho. 2018. [Deterministic non-autoregressive neural sequence modeling by iterative refinement](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1173–1182, Brussels, Belgium. Association for Computational Linguistics.
- Ran Levy, Yonatan Bilu, Daniel Hershcovich, Ehud Aharoni, and Noam Slonim. 2014. [Context dependent claim detection](#). In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1489–1500, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

- Ran Levy, Ben Bogin, Shai Gretz, Ranit Aharonov, and Noam Slonim. 2018. [Towards an argumentative content search engine using weak supervision](#). In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2066–2081, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020a. [BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020b. [Retrieval-augmented generation for knowledge-intensive nlp tasks](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.
- Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. [A diversity-promoting objective function for neural conversation models](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California. Association for Computational Linguistics.
- Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. 2017. [Adversarial learning for neural dialogue generation](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages

2157–2169, Copenhagen, Denmark. Association for Computational Linguistics.

Juncen Li, Robin Jia, He He, and Percy Liang. 2018. [Delete, retrieve, generate: a simple approach to sentiment and style transfer](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1865–1874, New Orleans, Louisiana. Association for Computational Linguistics.

Rongzhong Lian, Min Xie, Fan Wang, Jinhua Peng, and Hua Wu. 2019. Learning to select knowledge for response generation in dialog systems. *arXiv preprint arXiv:1902.04911*.

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

Chin-Yew Lin and Eduard Hovy. 2000a. [The automated acquisition of topic signatures for text summarization](#). In *COLING 2000 Volume 1: The 18th International Conference on Computational Linguistics*.

Chin-Yew Lin and Eduard Hovy. 2000b. The automated acquisition of topic signatures for text summarization. In *Proceedings of the 18th conference on Computational linguistics-Volume 1*, pages 495–501. Association for Computational Linguistics.

Marco Lippi and Paolo Torrioni. 2016. Argumentation mining: State of the art and emerging trends. *ACM Transactions on Internet Technology (TOIT)*, 16(2):10.

- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Teng Long, Emmanuel Bengio, Ryan Lowe, Jackie Chi Kit Cheung, and Doina Precup. 2017. [World knowledge for reading comprehension: Rare entity prediction with hierarchical LSTMs using external descriptions](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 825–834, Copenhagen, Denmark. Association for Computational Linguistics.
- Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. [Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3219–3232, Brussels, Belgium. Association for Computational Linguistics.
- Fuli Luo, Damai Dai, Pengcheng Yang, Tianyu Liu, Baobao Chang, Zhifang Sui, and Xu Sun. 2019. [Learning to control the fine-grained sentiment for story ending generation](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6020–6026, Florence, Italy. Association for Computational Linguistics.
- Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2015a. [Multi-task sequence to sequence learning](#). In *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Thang Luong, Hieu Pham, and Christopher D. Manning. 2015b. [Effective approaches to attention-based neural machine translation](#). In *Proceedings of the*

- 2015 *Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.
- Shervin Malmasi and Marcos Zampieri. 2018. Challenges in discriminating profanity from hate speech. *Journal of Experimental & Theoretical Artificial Intelligence*, 30(2):187–202.
- Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. [The Stanford CoreNLP natural language processing toolkit](#). In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.
- Elman Mansimov, Alex Wang, Sean Welleck, and Kyunghyun Cho. 2019. A generalized framework of sequence generation with application to undirected sequence models. *arXiv preprint arXiv:1905.12790*.
- Tobias Mayer, Elena Cabrio, and Serena Villata. 2020. Transformer-based argument mining for healthcare applications. In *ECAI 2020 - 24th European Conference on Artificial Intelligence*, volume 325 of *Frontiers in Artificial Intelligence and Applications*, pages 2108–2115. IOS Press.
- Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. [On faithfulness and factuality in abstractive summarization](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.
- Kris McGuffie and Alex Newhouse. 2020. The radicalization risks of gpt-3 and advanced neural language models. *arXiv preprint arXiv:2009.06807*.

- Kathleen R. McKeown. 1985. *Text Generation: Using Discourse Strategies and Focus Constraints to Generate Natural Language Text*. Cambridge University Press, New York, NY, USA.
- Hongyuan Mei, Mohit Bansal, and Matthew R. Walter. 2016. [What to talk about and how? selective generation using LSTMs with coarse-to-fine alignment](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 720–730, San Diego, California. Association for Computational Linguistics.
- Todor Mihaylov and Anette Frank. 2018. [Knowledgeable reader: Enhancing cloze-style reading comprehension with external commonsense knowledge](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 821–832, Melbourne, Australia. Association for Computational Linguistics.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.
- George A. Miller. 1994. [WordNet: A lexical database for English](#). In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.
- Raquel Mochales and Marie-Francine Moens. 2011. Argumentation mining. *Artificial Intelligence and Law*, 19(1):1–22.
- Marie-Francine Moens, Erik Boiy, Raquel Mochales Palau, and Chris Reed. 2007. Automatic detection of arguments in legal texts. In *Proceedings of the 11th international conference on Artificial intelligence and law*, pages 225–230. ACM.

- Amit Moryossef, Yoav Goldberg, and Ido Dagan. 2019. [Step-by-step: Separating planning from realization in neural data-to-text generation](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2267–2277, Minneapolis, Minnesota. Association for Computational Linguistics.
- Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gülçehre, and Bing Xiang. 2016. [Abstractive text summarization using sequence-to-sequence RNNs and beyond](#). In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany. Association for Computational Linguistics.
- Huy Nguyen and Diane Litman. 2016. [Context-aware argumentative relation mining](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1127–1137, Berlin, Germany. Association for Computational Linguistics.
- Huy Nguyen and Diane Litman. 2018. Argument mining for improving the automated scoring of persuasive essays. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.
- Jianmo Ni and Julian McAuley. 2018. [Personalized review generation by expanding phrases and attending on aspect-aware representations](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 706–711, Melbourne, Australia. Association for Computational Linguistics.
- Vlad Niculae, Joonsuk Park, and Claire Cardie. 2017. [Argument mining with structured SVMs and RNNs](#). In *Proceedings of the 55th Annual Meeting of the*

- Association for Computational Linguistics (Volume 1: Long Papers)*, pages 985–995, Vancouver, Canada. Association for Computational Linguistics.
- Eric W Noreen. 1989. *Computer-intensive methods for testing hypotheses*. Wiley New York.
- Roman Novak, Michael Auli, and David Grangier. 2016. Iterative refinement for machine translation. *arXiv preprint arXiv:1610.06602*.
- Naoaki Okazaki. 2007. [Crfsuite: a fast implementation of conditional random fields \(crfs\)](#).
- Ray Oshikawa, Jing Qian, and William Yang Wang. 2020. [A survey on natural language processing for fake news detection](#). In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 6086–6093, Marseille, France. European Language Resources Association.
- Raquel Mochales Palau and Marie-Francine Moens. 2009. Argumentation mining: the detection, classification and structure of arguments in text. In *Proceedings of the 12th international conference on artificial intelligence and law*, pages 98–107. ACM.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. [Bleu: a method for automatic evaluation of machine translation](#). In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.
- Joonsuk Park, Cheryl Blake, and Claire Cardie. 2015. Toward machine-assisted participation in erulemaking: An argumentation model of evaluability. In

Proceedings of the 15th International Conference on Artificial Intelligence and Law, pages 206–210. ACM.

Joonsuk Park and Claire Cardie. 2014. [Identifying appropriate support for propositions in online user comments](#). In *Proceedings of the First Workshop on Argumentation Mining*, pages 29–38, Baltimore, Maryland. Association for Computational Linguistics.

Joonsuk Park and Claire Cardie. 2018. [A corpus of eRulemaking user comments for measuring evaluability of arguments](#). In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Joonsuk Park, Sally Klingel, Claire Cardie, Mary Newhart, Cynthia Farina, and Joan-Josep Vallbé. 2012. Facilitative moderation for online participation in erulemaking. In *Proceedings of the 13th Annual International Conference on Digital Government Research*, pages 173–182. ACM.

Adam Paszke, Sam Gross, Soumith Chintala, and Gregory Chanan. 2017. Pytorch: Tensors and dynamic neural networks in python with strong gpu acceleration. *PyTorch: Tensors and dynamic neural networks in Python with strong GPU acceleration*.

Andreas Peldszus and Manfred Stede. 2013. From argument diagrams to argumentation mining in texts: A survey. *International Journal of Cognitive Informatics and Natural Intelligence (IJCINI)*, 7(1):1–31.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. [GloVe: Global vectors for word representation](#). In *Proceedings of the 2014 Conference on*

- Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.
- Verónica Pérez-Rosas, Bennett Kleinberg, Alexandra Lefevre, and Rada Mihalcea. 2018. [Automatic detection of fake news](#). In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3391–3401, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- Isaac Persing and Vincent Ng. 2016. [End-to-end argumentation mining in student essays](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1384–1394, San Diego, California. Association for Computational Linguistics.
- Isaac Persing and Vincent Ng. 2020. [Unsupervised argumentation mining in student essays](#). In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 6795–6803, Marseille, France. European Language Resources Association.
- Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. [Deep contextualized word representations](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.
- Jan Pomikálek. 2011. *Removing boilerplate and duplicate content from web corpora*. Ph.D. thesis, Masaryk university, Faculty of informatics, Brno, Czech Republic.

- Matt Post and David Vilar. 2018. [Fast lexically constrained decoding with dynamic beam allocation for neural machine translation](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1314–1324, New Orleans, Louisiana. Association for Computational Linguistics.
- Shrimai Prabhumoye, Alan W Black, and Ruslan Salakhutdinov. 2020. [Exploring controllable text generation techniques](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1–14, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. 2008. [The Penn Discourse TreeBank 2.0](#). In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).
- Simon Price and Peter A Flach. 2017. Computational support for academic peer review: A perspective from artificial intelligence. *Communications of the ACM*, 60(3):70–79.
- Ratish Puduppully, Li Dong, and Mirella Lapata. 2019. [Data-to-text generation with entity modeling](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2023–2035, Florence, Italy. Association for Computational Linguistics.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

- Lance A Ramshaw and Mitchell P Marcus. 1999. Text chunking using transformation-based learning. In *Natural language processing using very large corpora*, pages 157–176. Springer.
- Chris Reed. 1999. The role of saliency in generating natural language arguments. In *IJCAI*, pages 876–883.
- Chris Reed, Derek Long, and Maria Fox. 1996. An architecture for argumentative dialogue planning. In *International Conference on Formal and Applied Practical Reasoning*, pages 555–566. Springer.
- Lena Reed, Shereen Oraby, and Marilyn Walker. 2018. [Can neural generators for dialogue learn sentence planning and discourse structuring?](#) In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 284–295, Tilburg University, The Netherlands. Association for Computational Linguistics.
- Nils Reimers and Iryna Gurevych. 2017. [Reporting score distributions makes a difference: Performance study of LSTM-networks for sequence tagging.](#) In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 338–348, Copenhagen, Denmark. Association for Computational Linguistics.
- Paul Reisert, Naoya Inoue, Naoaki Okazaki, and Kentaro Inui. 2015. [A computational approach for generating toulmin model argumentation.](#) In *Proceedings of the 2nd Workshop on Argumentation Mining*, pages 45–55, Denver, CO. Association for Computational Linguistics.
- Ehud Reiter and Robert Dale. 2000. *Building natural language generation systems*. Cambridge university press.

- Ehud Reiter, Roma Robertson, and Liesl Osman. 2000. [Knowledge acquisition for natural language generation](#). In *INLG'2000 Proceedings of the First International Conference on Natural Language Generation*, pages 217–224, Mitzpe Ramon, Israel. Association for Computational Linguistics.
- Leonardo FR Ribeiro, Martin Schmitt, Hinrich Schütze, and Iryna Gurevych. 2020. Investigating pretrained language models for graph-to-text generation. *arXiv preprint arXiv:2007.08426*.
- Richard D Rieke, Malcolm Osgood Sillars, and Tarla Rai Peterson. 1997. *Argumentation and critical decision making*. New York: Longman.
- Ruty Rinott, Lena Dankin, Carlos Alzate Perez, Mitesh M. Khapra, Ehud Aharoni, and Noam Slonim. 2015. [Show me your evidence - an automatic method for context dependent evidence detection](#). In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 440–450, Lisbon, Portugal. Association for Computational Linguistics.
- Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. 1995. Okapi at trec-3. *Nist Special Publication Sp*, 109:109.
- Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. 2018. [Object hallucination in image captioning](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4035–4045, Brussels, Belgium. Association for Computational Linguistics.
- Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. [A neural attention model for abstractive sentence summarization](#). In *Proceedings of the 2015*

- Conference on Empirical Methods in Natural Language Processing*, pages 379–389, Lisbon, Portugal. Association for Computational Linguistics.
- Evan Sandhaus. 2008. The new york times annotated corpus. *Linguistic Data Consortium, Philadelphia*, 6(12):e26752.
- Cicero Nogueira dos Santos, Igor Melnyk, and Inkit Padhi. 2018. [Fighting offensive language on social media with unsupervised text style transfer](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 189–194, Melbourne, Australia. Association for Computational Linguistics.
- Misa Sato, Kohsuke Yanai, Toshinori Miyoshi, Toshihiko Yanase, Makoto Iwayama, Qinghua Sun, and Yoshiki Niwa. 2015. [End-to-end argument generation system in debating](#). In *Proceedings of ACL-IJCNLP 2015 System Demonstrations*, pages 109–114, Beijing, China. Association for Computational Linguistics and The Asian Federation of Natural Language Processing.
- Claudia Schulz, Steffen Eger, Johannes Daxenberger, Tobias Kahse, and Iryna Gurevych. 2018. [Multi-task learning for argumentation mining in low-resource settings](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 35–41, New Orleans, Louisiana. Association for Computational Linguistics.
- Donia Scott and Clarisse Sieckenius de Souza. 1990. Getting the message across in rst-based text generation. *Current research in natural language generation*, 4:47–73.
- Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. [Get to the point:](#)

- Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.
- Abigail See, Stephen Roller, Douwe Kiela, and Jason Weston. 2019. [What makes a good conversation? how controllable attributes affect human judgments](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1702–1723, Minneapolis, Minnesota. Association for Computational Linguistics.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. [Neural machine translation of rare words with subword units](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.
- Chengcheng Shao, Giovanni Luca Ciampaglia, Onur Varol, Alessandro Flammini, and Filippo Menczer. 2017. The spread of fake news by social bots. *arXiv preprint arXiv:1707.07592*, 96:104.
- Emily Sheng, Kai-Wei Chang, Premkumar Natarajan, and Nanyun Peng. 2019. [The woman worked as a babysitter: On biases in language generation](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3407–3412, Hong Kong, China. Association for Computational Linguistics.
- Noam Slonim, Yonatan Bilu, Carlos Alzate, Roy Bar-Haim, Ben Bogin, Francesca

- Bonin, Leshem Choshen, Edo Cohen-Karlik, Lena Dankin, Lilach Edelstein, et al. 2021. An autonomous debating system. *Nature*, 591(7850):379–384.
- Carlota S Smith. 2003. *Modes of discourse: The local structure of texts*, volume 103. Cambridge University Press.
- Ahmed Soliman, Jan Hafer, and Florian Lemmerich. 2019. A characterization of political communities on reddit. In *Proceedings of the 30th ACM conference on hypertext and Social Media*, pages 259–263.
- Zhenqiao Song, Xiaoqing Zheng, Lu Liu, Mu Xu, and Xuanjing Huang. 2019. [Generating responses with a specific emotion in dialog](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3685–3695, Florence, Italy. Association for Computational Linguistics.
- Alessandro Sordani, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. 2015. [A neural network approach to context-sensitive generation of conversational responses](#). In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 196–205, Denver, Colorado. Association for Computational Linguistics.
- Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.
- Valentin I. Spitzkovsky and Angel X. Chang. 2012. A cross-lingual dictionary for English Wikipedia concepts. In *Language Resources and Evaluation (LREC)*, Istanbul, Turkey.

- Christian Stab, Johannes Daxenberger, Chris Stahlhut, Tristan Miller, Benjamin Schiller, Christopher Tauchmann, Steffen Eger, and Iryna Gurevych. 2018. [ArgumentText: Searching for arguments in heterogeneous sources](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 21–25, New Orleans, Louisiana. Association for Computational Linguistics.
- Christian Stab and Iryna Gurevych. 2014. [Identifying argumentative discourse structures in persuasive essays](#). In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 46–56, Doha, Qatar. Association for Computational Linguistics.
- Christian Stab and Iryna Gurevych. 2017. [Parsing argumentation structures in persuasive essays](#). *Computational Linguistics*, 43(3):619–659.
- Amanda Stent, Rashmi Prasad, and Marilyn Walker. 2004. [Trainable sentence planning for complex information presentations in spoken dialog systems](#). In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 79–86, Barcelona, Spain.
- Philip J Stone, Dexter C Dunphy, and Marshall S Smith. 1966. The general inquirer: A computer approach to content analysis.
- Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.
- Bowen Tan, Zichao Yang, Maruan Al-Shedivat, Eric Xing, and Zhiting Hu. 2021. [Progressive generation of long text with pretrained language models](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association*

- for Computational Linguistics: Human Language Technologies*, pages 4313–4324, Online. Association for Computational Linguistics.
- Jian Tang, Yifan Yang, Sam Carton, Ming Zhang, and Qiaozhu Mei. 2016. Context-aware natural language generation with recurrent neural networks. *arXiv preprint arXiv:1611.09900*.
- Milagro Teruel, Cristian Cardellino, Fernando Cardellino, Laura Alonso Alemany, and Serena Villata. 2018. Increasing argument annotation reproducibility by using inter-annotator agreement to improve guidelines. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.
- Simone Teufel, Advait Siddharthan, and Colin Batchelor. 2009. [Towards domain-independent argumentative zoning: Evidence from chemistry and computational linguistics](#). In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1493–1502, Singapore. Association for Computational Linguistics.
- Ran Tian, Shashi Narayan, Thibault Sellam, and Ankur P Parikh. 2019. Sticking to the facts: Confident decoding for faithful data-to-text generation. *arXiv preprint arXiv:1910.08684*.
- Stephen Edelston Toulmin. 1958. *The use of argument*. Cambridge University Press.
- Minh Tran, Yipeng Zhang, and Mohammad Soleymani. 2020. [Towards a friendly online community: An unsupervised style transfer framework for profanity redaction](#). In *Proceedings of the 28th International Conference on Com-*

- putational Linguistics*, pages 2107–2114, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Joshua Uyheng and Kathleen M Carley. 2020. Bots and online hate during the covid-19 pandemic: case studies in the united states and the philippines. *Journal of Computational Social Science*, 3(2):445–468.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015a. [Pointer networks](#). In *Advances in Neural Information Processing Systems*, volume 28, pages 2692–2700. Curran Associates, Inc.
- Oriol Vinyals and Quoc Le. 2015. A neural conversational model. *arXiv preprint arXiv:1506.05869*.
- Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015b. [Show and tell: A neural image caption generator](#). In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3156–3164.
- Henning Wachsmuth, Nona Naderi, Ivan Habernal, Yufang Hou, Graeme Hirst, Iryna Gurevych, and Benno Stein. 2017a. [Argumentation quality assessment: Theory vs. practice](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 250–255, Vancouver, Canada. Association for Computational Linguistics.
- Henning Wachsmuth, Martin Potthast, Khalid Al-Khatib, Yamen Ajour, Jana Puschmann, Jiani Qu, Jonas Dorsch, Viorel Morari, Janek Bevendorff, and

- Benno Stein. 2017b. [Building an argument search engine for the web](#). In *Proceedings of the 4th Workshop on Argument Mining*, pages 49–59, Copenhagen, Denmark. Association for Computational Linguistics.
- Henning Wachsmuth, Manfred Stede, Roxanne El Baff, Khalid Al-Khatib, Maria Skeppstedt, and Benno Stein. 2018a. [Argumentation synthesis following rhetorical strategies](#). In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3753–3765, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- Henning Wachsmuth, Shahbaz Syed, and Benno Stein. 2018b. [Retrieval of the best counterargument without prior topic knowledge](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 241–251, Melbourne, Australia. Association for Computational Linguistics.
- Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. [Universal adversarial triggers for attacking and analyzing NLP](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2153–2162, Hong Kong, China. Association for Computational Linguistics.
- Douglas Walton, Christopher Reed, and Fabrizio Macagno. 2008. *Argumentation schemes*. Cambridge University Press.
- Di Wang, Nebojsa Jojic, Chris Brockett, and Eric Nyberg. 2017a. [Steering output style and topic in neural response generation](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2140–2150, Copenhagen, Denmark. Association for Computational Linguistics.

- Lu Wang, Nick Beauchamp, Sarah Shugars, and Kechen Qin. 2017b. [Winning on the merits: The joint effects of content and style on debate outcomes](#). *Transactions of the Association for Computational Linguistics*, 5:219–232.
- Lu Wang and Wang Ling. 2016. [Neural network-based abstract generation for opinions and arguments](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 47–57, San Diego, California. Association for Computational Linguistics.
- Wenbo Wang, Lu Chen, Krishnaprasad Thirunarayan, and Amit P Sheth. 2014. [Cursing in english on twitter](#). In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*, pages 415–425.
- Amy Beth Warriner, Victor Kuperman, and Marc Brysbaert. 2013. Norms of valence, arousal, and dominance for 13,915 english lemmas. *Behavior research methods*, 45(4):1191–1207.
- Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, Lina M. Rojas-Barahona, Pei-Hao Su, Stefan Ultes, David Vandyke, and Steve Young. 2016. [Conditional generation and snapshot learning in neural dialogue systems](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2153–2162, Austin, Texas. Association for Computational Linguistics.
- Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. 2015. [Semantically conditioned LSTM-based natural language generation for spoken dialogue systems](#). In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1711–1721, Lisbon, Portugal. Association for Computational Linguistics.

- Rongxiang Weng, Heng Yu, Xiangpeng Wei, and Weihua Luo. 2020. [Towards enhancing faithfulness for neural machine translation](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2675–2684, Online. Association for Computational Linguistics.
- Jason Weston, Emily Dinan, and Alexander Miller. 2018. [Retrieve and refine: Improved sequence generation models for dialogue](#). In *Proceedings of the 2018 EMNLP Workshop SCAI: The 2nd International Workshop on Search-Oriented Conversational AI*, pages 87–92, Brussels, Belgium. Association for Computational Linguistics.
- Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. [Recognizing contextual polarity in phrase-level sentiment analysis](#). In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 347–354, Vancouver, British Columbia, Canada. Association for Computational Linguistics.
- Sam Wiseman and Alexander M. Rush. 2016. [Sequence-to-sequence learning as beam-search optimization](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1296–1306, Austin, Texas. Association for Computational Linguistics.
- Sam Wiseman, Stuart Shieber, and Alexander Rush. 2017. [Challenges in data-to-document generation](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2253–2263, Copenhagen, Denmark. Association for Computational Linguistics.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite,

- Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.
- Linda Woodson. 1979. *A handbook of modern rhetorical terms*. National Council of Teachers.
- Stephen J Wright. 2015. Coordinate descent algorithms. *Mathematical Programming*, 151(1):3–34.
- Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google’s neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.
- Zequi Wu, Michel Galley, Chris Brockett, Yizhe Zhang, Xiang Gao, Chris Quirk, Rik Koncel-Kedziorski, Jianfeng Gao, Hannaneh Hajishirzi, Mari Ostendorf, et al. 2020. A controllable model of grounded response generation. *arXiv preprint arXiv:2005.00613*.
- Yingce Xia, Fei Tian, Lijun Wu, Jianxin Lin, Tao Qin, Nenghai Yu, and Tie-Yan Liu. 2017. Deliberation networks: Sequence generation beyond one-pass decoding. In *Advances in Neural Information Processing Systems*, pages 1784–1794.
- Toshihiko Yanase, Toshinori Miyoshi, Kohsuke Yanai, Misa Sato, Makoto Iwayama, Yoshiki Niwa, Paul Reisert, and Kentaro Inui. 2015. [Learning sentence ordering for opinion generation of debate](#). In *Proceedings of the 2nd*

- Workshop on Argumentation Mining*, pages 94–103, Denver, CO. Association for Computational Linguistics.
- Kai-Cheng Yang, Christopher Torres-Lugo, and Filippo Menczer. 2020. Prevalence of low-credibility information on twitter during the covid-19 outbreak. *arXiv preprint arXiv:2004.14484*.
- Lili Yao, Nanyun Peng, Ralph Weischedel, Kevin Knight, Dongyan Zhao, and Rui Yan. 2019a. [Plan-and-write: Towards better automatic storytelling](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):7378–7385.
- Lili Yao, Nanyun Peng, Ralph Weischedel, Kevin Knight, Dongyan Zhao, and Rui Yan. 2019b. Plan-and-write: Towards better automatic storytelling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7378–7385.
- Zhiwei Yu, Jiwei Tan, and Xiaojun Wan. 2018. [A neural approach to pun generation](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1650–1660, Melbourne, Australia. Association for Computational Linguistics.
- Ming Yuan and Yi Lin. 2006. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67.
- Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. [Defending against neural fake news](#). In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 9054–9065. Curran Associates, Inc.

Yizhe Zhang, Siqu Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2019. Dialogpt: Large-scale generative pre-training for conversational response generation. *arXiv preprint arXiv:1911.00536*.

Hao Zhou, Tom Young, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. 2018. [Commonsense knowledge aware conversation generation with graph attention](#). In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 4623–4629. International Joint Conferences on Artificial Intelligence Organization.

Kun Zhou, Wayne Xin Zhao, Shuqing Bian, Yuanhang Zhou, Ji-Rong Wen, and Jingsong Yu. 2020. Improving conversational recommender systems via knowledge graph based semantic fusion. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1006–1014.

Ingrid Zukerman, Richard McConachy, and Sarah George. 2000. [Using argumentation strategies in automated argument generation](#). In *INLG'2000 Proceedings of the First International Conference on Natural Language Generation*, pages 55–62, Mitzpe Ramon, Israel. Association for Computational Linguistics.