Improving Controllability for Neural Text Generation

Xinyu Hua

March, 2020

Khoury College of Computer Sciences
Northeastern University
Boston, MA 02115

Thesis Committee:
Lu Wang (Chair), Northeastern University
David Smith, Northeastern University
Byron Wallace, Northeastern University
Vincent Ng, University of Texas at Dallas

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

Abstract

Text generation is a core task in artificial intelligence. Over the past decade, neural text generation models have made substantial progress in a wide range of applications, such as machine translation, summarization, and dialogue generation. The dominant approach is the sequence-to-sequence (seq2seq) paradigm with attention mechanism. A typical seq2seq model is end-to-end trained with log likelihood maximization over the target sequence on token-level.

In practice, this framework is simple and effective. It can produce fluent and grammatical output since it directly learns a language model. However, several issues are widely observed across tasks and domains. Due to limited control over the output, the model is prone to hallucination and generic response, especially for open-ended tasks where the output introduces new information that cannot be found in the input. Additionally, for longer and more structured text generation, the model does not produce very cohesive results. As it is not optimized during training.

In this thesis, we aim to improve the controllability of neural text generation models through explicit text planning and conditioning over desired discourse functions. Inspired by traditional text generation systems, we employ a separate text planning module that performs content selection (*what to say*) and ordering (*when to say what*). The generated text plans serve as constraints over the final surface realization stage, which is trained to faithfully reflect the selected content and user-specified discourse functions.

We focus on a novel argument generation task as a test bed. The goal of this task is to generate a counter-argument paragraph given an input statement of certain stance over a controversial topic. It differs from conventional seq2seq tasks in two main aspects: 1) successful arguments exhibit rich discourse structures, for which the text planning becomes extremely important. 2) Unlike machine translation and summarization, the output often contains information not mentioned in the input, necessitating access to external knowledge and ability to use it in a natural way.

In the first part of this thesis, we discuss an argument generation system that is capable of retrieving supporting evidence from Wikipedia articles. The retrieved content is encoded together with the input. During decoding, we adopt a multi-task learning based approach to encourage the inclusion of external evidence in the final output. In the second part, we focus on improving the text planning module. We formulate text planning as a sentence-level keyphrase selection procedure. Based on the input text and retrieval results, we first build a set of candidate keyphrases. The text planner determines the inclusion of each keyphrase for the output sentences, along which it also determines the sentence discourse functions. Then the surface realization module produces the output that conforms to the text plans. In the last section, we propose a template-based generation method to better reflect the effect of fine-grained discourse functions over the lexical level realization outcome. We also plan to explore iterative refinement methods to promote the realization faithfulness while maintaining the overall fluency.

Contents

Chapter 1

Introduction

1.1 Motivation

The automatic construction of natural language text is a fundamental problem in artificial intelligence, as it enables more efficient interaction between machine and human. Traditionally, text generation systems aim to convert certain input modality into the desired output text, such as translating French text into English, describing images with captions, or summarizing long documents into headlines. Before the emergence of neural models, most text generation systems are template-based and equipped with hand-crafted rules to ensure the correctness of the output [6, 18, 23, 42, 44, 54]. Consequently, building such systems is labor intensive and the same procedure likely has to be repeated for different applications.

The breakthrough of sequence-to-sequence (seq2seq) learning has made tremendous progress in this area [3, 46, 68]. A typical seq2seq model comprises of an encoder-decoder structure, where the encoder consumes the input sequence symbols and converts them into dense hidden representations, and the decoder learns to produce the output symbols one at a time. The learning is driven by maximizing log likelihood of the ground-truth symbols over each time step, effectively specifying a conditional language model $P(y_i|y_{<i}, \boldsymbol{x})$, where $\boldsymbol{x}$ is the input sequence and y_i is the i-th token in the output. The model can be end-to-end trained in a completely data-driven manner, without the need for rule crafting or expert knowledge. More recently, large-scale pre-training on deep Transformer models has pushed the boundary of this paradigm even further [14, 53, 75]. The openAI GPT-2 model, which is trained over 40GB of Internet text, has shown to be capable to generate impressively fluent document for virtually any given topic.

However, seq2seq models have several known limitations. The auto-regressive generation procedure results in low *controllability*. A decoding error in one step can steer away the entire generation to a different semantic space [7, 55]. Consequently, the model can produce fabricated events (*hallucination*) [58, 73] or it might always resort to a generic and boring response [35, 66]. Additionally, although with sufficient training the model can approach human level fluency, its outputs usually lack cohesiveness and proper discourse structures. Because learning such global properties is difficult due to the lack of accurate supervision signals.

In this thesis, we aim to design neural text generation models with improved controllability over the target output. We draw inspirations from traditional text generation systems, where an

intermediate text planning step is performed prior to the surface form realization. We adopt a more generalized definition of text planning, which includes the selection of relevant content (*what to say*) and the sentence-level content organization (*when to say what*). Concretely, our system manipulates keyphrases as content units. Given a set of candidate keyphrases that are either retrieved or user-specified, the text planner predicts the inclusion of them for each sentence. The surface realization module then produces the final output, given lexical constraints imposed by the text plans. We conduct experiments across different domains to verify the effectiveness of this proposal. In particular, we focus on a novel argument generation task, as it necessitates more fine-grained text planning and control to achieve desired communication goals. More details are discussed in the following section.

1.2 Argument Generation

Argumentation is a crucial procedure in various aspects of our lives, including informed decision making [20, 28, 29, 45], critical thinking training [2, 61], and online civic discussions [10, 48]. However, manual construction of arguments is time consuming and often requires domain expertise to guarantee the soundness and persuasiveness of the argument. Existing work on automated argument construction mostly leverages a retrieval based method [8, 17, 62] and a stance classifier [4] to extract indexed human written arguments. While these prototypes demonstrate great potentials, they are often limited by the human arguments in domains and diversity. They can hardly adapt to unseen topics or utilize emerging evidence or facts.

To address the above issues, we utilize the data-driven neural generation approach. In order to leverage the powerful seq2seq framework, we formulate the argument generation task as: given an input statement of certain stance for a controversial topic, generating a counter-argument paragraph that offers an alternative stance. For experiments, we collect a large scale dataset from Reddit `r/ChangeMyView` - an online forum that encourages open discussions. A typical post in this forum is posted by an OP (original poster) user, who starts the threads with a post of her stance over some topic (e.g., *Government should be allowed to view my e-mails*). Other users engage in the discussion through replying to the thread, among which the ones deemed persuasive are marked with a special `delta` symbol. We leverage such signal and user endorsement (`karma`) to retain high quality persuasive replies for study.

Although neural text generation models have achieved outstanding results in tasks such as machine translation and summarization, distinct challenges exist for this open-ended domain:

- Human usually leverages external information as supporting evidence or logic reasoning. As a result, the input alone is insufficient to infer the output, and it is prohibitively difficult to learn such external world knowledge from a single dataset.
- Persuasive arguments often exhibit proper discourse structures, which may vary depending on the type of evidence or reasoning. Labeling such types are time consuming and often suffers from low agreements.

In the following section, we discuss our proposed models to tackle the above challenges.

1.3 Method Overview

In Chapter 3 we first introduce a neural argument generation framework equipped with a retrieval component. This model is able to retrieve and rank relevant evidence sentences from Wikipedia, which are then encoded together with the input statement as a concatenated sequence. For decoding, we employ a multi-task learning based approach to first generates the keyphrases as talking points, followed by the target argument in a separate decoder. Experimental results demonstrate that our proposed model achieves better BLEU [49] and METEOR [13] scores than a baseline model without such information. While human judges still prefers the pure retrieval based system over our model, highlighting the challenge for the generation paradigm in this domain.

In order to achieve better controllability, we present a two-step neural generation model as detailed in Chapter 4. The key contribution is a separate text planning module that can be end-to-end trained with the main model, which also produces content selection and sentence style specification. We utilize keyphrases as selection units, and supply a set of keyphrase candidates as additional input to facilitate text planning. The text planner first determines which keyphrases will be used for each sentence, through an attention mechanism. The selection results will be converted into vectors to participate in the calculation of surface realization decoder states. During training, an additional loss term is added to account for the selection accuracy, based on match with human examples. In order to verify the generalizability of this approach, we consider two additional domains: the Wikipedia introduction generation and paper abstract generation (AGENDA [31]) tasks. Experimental results demonstrate improved automatic metrics over comparisons without the planning stage. Moreover, we observe a strong correlation between the keyphrase selection and the final output quality in BLEU and ROUGE [38], further implying the benefits of the text planner.

Our preliminary work so far has identified text planning as a crucial step in ensuring the generation quality for argument generation. The controllability it offers are mainly from the learned representation of the selected keyphrases and the coarse-grained style specification. However, there is still no guarantee that the supplied keyphrases will be used by the surface realization decoder. Even when they are used, it might also result in unnatural output when the model fails to generate necessary functional words. To mitigate these problems, our major plan for the rest of this thesis is to design a refinement-based generation method that can better handle the modeling of content, functional, and discourse words given the appropriate discourse functions. We hypothesize that the interplay between content and sentence function can be leveraged to predict the proper discourse templates.

Chapter 2

Related Work

2.1 Natural Language Generation (NLG)

Natural language generation concerns the general task of automatically producing natural language output. Depending on the input modality, NLG can be broadly categorized as text-to-text (e.g., machine translation, summarization, and dialogue generation) and data-to-text (e.g., sports and weather report generation, review generation given attributes). Over the past decade, data-driven approaches have been widely applied to text-to-text tasks and achieved impressive results [3, 59, 68]. Meanwhile, due to the difficulties in constructing large-scale paired corpora, data-to-text tasks still pose challenges, especially in producing both fluent and faithful output [52, 73]. Recent endeavors are limited to specific domains and text genres, and mostly deal with relatively short outputs [11, 16]. Despite promising results, it is unclear how well these models can transfer to new domains with more structured output. Part of the goal of this thesis is to design NLG models that can handle such scenario.

Many traditional NLG systems break down the task into three steps. **Content determination** selects a subset of the input information that are relevant to the context and audience. For example, Barzilay and Lee model the dynamic of topic shifts to determine the order of information to present [5]. Duboue and McKeown cluster the semantic input and study their associations with language models, which in turn inform the inclusion of different data in the generation [15]. Second, during the **text planning** stage, the model organizes the selected data in a logical and natural order. Because of the difficulties in acquiring labeled data, existing methods mostly rely on hand-crafted rules and domain knowledge such as Rhetorical Structure Theory [23, 42, 44]. In the final **surface realization** stage, the system converts the intermediate text plans into natural language text. The major challenge lies in the lexical variability and how to generate functional words unseen from the input. To ensure the grammticality, early methods rely on templates that are manually constructed for certain domains [6, 18, 43, 57, 74].

2.2 Neural Encoder-Decoder Models

One recent breakthrough in this area is the neural encoder-decoder framework. It is first proposed for machine translation [68], where a recurrent neural network (RNN) based encoder first

consumes the source language as a sequence of tokens and learns dense representations for each of them. The decoder, also a RNN model, yields the target language token-by-token. To facilitate the learning of alignment between source and target, attention mechanism is proposed to weight the encoder states as context for each decoder step [3]. Unlike many rule-based approaches, this framework can be end-to-end trained with minimum human intervention. It is also shown to perform remarkably well on other text-to-text tasks such as summarization [9, 59, 63] and dialogue generation [36, 66].

However, the sequential nature of RNN still makes long output generation difficult due to vanishing gradient [22]. The Transformer is proposed as an alternative [69], which solely relies on the attention mechanism to learn the dependencies between input and output. It not only achieves state-of-the-art performance for various sequence learning tasks [60, 69], but also shows great power in a wide range of NLP tasks with large-scale pre-training [14, 53, 75].

To date there exist a multitude of pre-trained Transformer frameworks, among which the BERT model [14] and OpenAI GPT-2 [53] are the most representative. The BERT model is an encoder-only Transformer. It adopts the masked-language model loss and is entirely non-autoregressive. Although it is primarily designed for natural language understanding tasks, researchers have shown its capabilities for NLG tasks [34, 72]. The OpenAI GPT-2 model is pre-trained in an autoregressive manner, which naturally learns a sequential language model. It thus can be easily fine-tuned for various NLG tasks without much architectural modifications [12, 21, 64].

2.3 Controllability in Text Generation

Large-scale pre-training and neural encoder-decoder framework are now the dominant ingredients for NLG models. Neural NLG models can be entirely data-driven and end-to-end trained without feature engineering or template crafting. Nonetheless, contrary to the traditional approaches, they blend the planning and realization tasks. As a result, it is hard to ensure the output will faithfully reflect the input data. For example, Wiseman, Shieber, and Rush observe that the neural data-to-text generation model tends to hallucinate factual information that cannot be found in the input [73]. Furthermore, in long output such as argumentative essays, human experts have rated neural model's generation as "not well-structured" and "not get to the point"[1]. This is likely because language model pre-training is insufficient to capture the discourse structures, which largely limits the usefulness of these models when the outputs serve to deliver certain communicative goals, such as to educate, to persuade, or to entertain.

To solve the above challenges, we believe it is crucial to understand and improve the **controllability** of neural NLG models. Prior work generally frames controllable text generation as producing output that satisfies certain given attributes, such as sentiment [19, 24], topic [12], text genres [30], and simplicity [67]. In this thesis, controllability is measured in two dimensions: 1) whether the generation correctly reflects the key semantic information in the input; 2) when desired discourse roles are given for each output sentence, how well the model can utilize them.

[1] https://www.economist.com/open-future/2019/10/01/how-to-respond-to-climate-change-if-you-are-an-algorithm

Chapter 3

Argument Generation with External Information (Completed Work)

3.1 Motivation and Task Formulation

Argumentation is one of the most important rhetorical modes in discourse theory [47]. It is adopted throughout our daily communication in a wide-range of settings, such as persuasive essay composition, corporation decision making, and political debate. Successful arguments usually contain a combination of credible evidence and convincing reasoning, which makes it a complex task even for humans. Thus we believe the automatic construction of arguments will not only expedite the decision-making process, but also serve as a crucial building block for general NLG frameworks.

Prior work on argument generation has highlighted two major challenges: 1) *the encoding of topic-related world knowledge*, and 2) *the design of the downstream NLG module given the topic and stance constraints*. A popular approach is to build an inventory of human-written arguments. Then, information retrieval tools can be utilized to search the most relevant existing arguments [62, 70, 71]. In order to construct natural language output, the dominant method is to rely on sentence ordering and stance classification algorithms based on argumentation theories and other human heuristics [56, 62]. Although the generated arguments are topic-relevant and locally fluent, they are inherently limited in expressiveness and domains.

In this work, we consider a similar retrieval framework for inclusion of external knowledge, but devise a neural NLG component for better content realization. We focus on the counter-argument generation task, where the model produces a paragraph-level counter-argument opposing an input argument. An example is shown below:

Input argument (OP): CMV: Bernie Sanders is too old to be president. Bernie Sanders is clearly an intelligent man with a lot of interesting views (. . .) But right now I feel I cannot truly support him for actually being elected president because of his age.
Counter-argument: This is one of the benefits of the grueling campaign trail. If the man is capable of making it through the year and a half primary and general election season, he is certainly more than fit to be president currently. (. . .) For some historical perspective, there have been many Presidents far younger who have had horrible health problems (JFK and LBJ, to name a few), while others have gone on to remain active in politics for years after.

Notice that the counter-argument starts with a reasoning on the relation between campaign and presidency, then draws historical precedents to further support its thesis. None of these information is directly mentioned in the input, demonstrating the necessities for external information, and the complexities in content organization.

3.2 Data Collection

We construct a new dataset for argument generation from Reddit ChangeMyView[1] (henceforth CMV) forum. It is an active online community featuring open discussion and persuasion. Each thread is started with an original post (OP), which describes opinions over certain topic. Other users offer counter-argument replies to change the view of the OP user. Whenever the OP user acknowledges a change of view, a `delta` will be awarded to that post. Additionally, like other forums, the system also records upvotes/downvotes endorsed by users. The difference of upvotes and downvotes is denoted as `karma`.

For quality control, we only retain the *root replies* that are: 1) longer than 5 words, 2) without offensive languages, 3) awarded with a `delta` or has positive `karma`. Further, certain topics are non-standard in argumentation study and largely personal preference (such as discussions on fictional characters). We thus focus on the politics domain, which tends to receive good popularity and also covers diverse topics. In order to isolate these threads, we build a logistic regression classifier with unigram features, learned from heuristically labeled Wikipedia abstracts. In total, we retain $12,549$ threads, each with 9.4 high-quality counter-arguments on average.

3.3 Framework

Our proposed pipeline consists of two components. The evidence retriever first queries Wikipedia for relevant articles, which are then reranked based on similarity to the input. A keyphrase extractor further condenses the evidence into a list of keyphrases as "talking points". Next, the neural argument generation component encodes the input and evidence, and generates the keyphrases in the first pass, followed by the final counter-argument in a separate decoder. We depict the simplified pipeline in Figure 3.1. More details can be found in our paper [25].

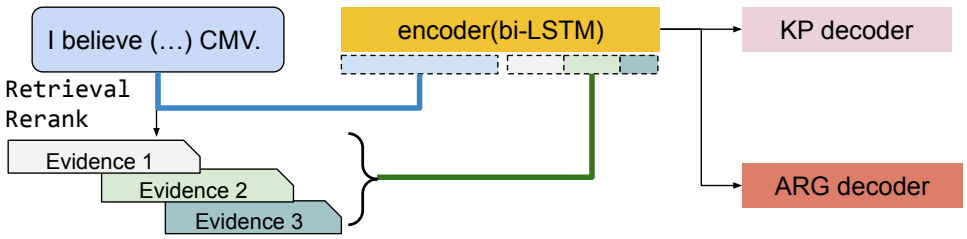

Figure 3.1: Our system pipeline. The input OP (blue) is first used to retrieve Wikipedia articles. The concatenation of the input and evidence is fed in the biLSTM encoder. The generation component first yields keyphrases as talking points, and then produces the counter-argument by an argument decoder.

[1] https://www.reddit.com/r/changemyview/

3.3.1 Evidence Retrieval and Reranking

We consider Wikipedia as the knowledge base because of its broad topic coverage and objective nature. We construct queries from either the counter-argument (training time) or input (inference time). We first identify the post-representative topic signature words [39]. We create query for each input/counter-argument sentence, by concatenating noun phrases and verbs that contain topic signatures. For each query, we keep the top five returned Wikipedia articles, which are then segmented into sentences and aggregated for TF-IDF based reranking. Top 10 sentences are kept as evidence for subsequent steps.

3.3.2 Keyphrase Extraction

Prior work has explored different kinds of information bearing items for content planning. Such as database records [32] and AMR trees [33, 65]. Our system operates over keyphrases, because they can be automatically identified and easily encoded sequentially. We first extract candidate noun phrases and verb phrases using Stanford CoreNLP [41], then filter base on length (2-10 tokens) and whether they contain content words.

3.3.3 Argument Generation

We formalize the task as given an input statement $\boldsymbol{x}^O$ and retrieved evidence $\boldsymbol{x}^E$, to generate a sequence of keyphrases $\boldsymbol{y}^p$ and argument $\boldsymbol{y}^a$. During training, we jointly optimize the likelihood of the two output sequences given the input $\boldsymbol{x} = \{\boldsymbol{x}^O; \boldsymbol{x}^E\}$. As illustrated in Figure 3.1, a bidirectional LSTM (biLSTM) encoder first consumes the sequence of concatenated inputs $[\boldsymbol{x}^O; \boldsymbol{x}^E]$ and learns a hidden states $\boldsymbol{h}_i$ for step i. We consider two separate LSTM decoders, both are initialized by the last hidden states of the encoder. We first generate keyphrases and then argument. Both decoders are equipped with the attention mechanism [3] to access the input. The argument decoder additionally attends the keyphrase decoder, enabling better utilization of content plans.

Decoding Strategy. At inference time, we run a **hybrid beam search** for argument decoding. Concretely, we modify the standard beam search: (1) Given beam size k, for each step, we take the top n words ($n < k$) deterministically and randomly draw the next $k - n$ words from the rest of vocabulary. (2) Every p steps, we rerank the beams base on coverage of input content words.

3.4 Experiments and Results

3.4.1 Experimental setups

To diversify the evidence samples and to expedite training, for each input statement sentence, we sample up to three evidence sentences to form one training example. We repeat three times for the same original instance. We split the final CMV dataset into $224,553$ for training, $13,911$ and $30,417$ for validation and test. We consider pre-training a standard seq2seq model with input OP and target argument as language model initialization for part of the full model. Experiments confirm that this step boosts METEOR [13] scores by more than 2 points.

3.4.2 Baselines and Comparisons

We report results over the following baseline and comparisons: (1) RETRIEVAL returns the concatenated evidence sentences; (2) SEQ2SEQ directly learns to generate argument from input OP; (3) SEQ2SEQ + *encode evd*, that encodes the concatenation of OP and evidence sentences. We further conduct ablation study by: a) decoding sharing (DEC-SHARED); b) disabling keyphrase attention (w/o *attend KP*).

3.4.3 Results and Discussions.

We report BLEU [49] (up to bigrams) and METEOR [13] for automatic evaluation. Table 3.1 shows that our proposed models generally achieve significantly better BLEU than all comparisons. The RETRIEVAL method scores higher METEOR, which is likely because its output are substantially longer and therefore have higher recall.

	BLEU	MTR	Len.
RETRIEVAL	15.32	**12.19**	151.2
SEQ2SEQ	10.21	5.74	34.9
+ *encode evd*	18.03	7.32	67.0
DEC-SHARED	**24.71**	10.05	74.8
w/o *attend KP*	21.22	8.91	69.1
DEC-SEPARATE	24.52	11.27	88.3
w/o *attend KP*	24.24	10.63	88.6

Table 3.1: Results on argument generation by BLEU-2 and METEOR (MTR). The best performing model is highlighted in **bold**.

We carry out human evaluation on **grammaticality**, **informativeness** (whether the argument contains useful information), and **relevance** (whether the argument is on-topic and of different stance). Three proficient English speakers are asked to read 30 randomly chosen sets of samples. For each output, human judges rate the three aspects on a scale of 1 (worst) to 5 (best), as shown in Table 3.2.

System	Gram.	Info.	Rel.
RETRIEVAL	**4.5** $\pm$ 0.6	**3.7** $\pm$ 0.9	**3.3** $\pm$ 1.1
SEQ2SEQ	3.3 $\pm$ 1.1	1.2 $\pm$ 0.5	1.4 $\pm$ 0.7
PROPOSED	2.5 $\pm$ 0.8	1.6 $\pm$ 0.8	1.8 $\pm$ 0.8

Table 3.2: Human evaluation results. Our model is perceived to be more informative and relevant than SEQ2SEQ baseline. While RETRIEVAL's output is significantly preferred by human in all aspects.

External evidence improves generation quality. Our proposed model that leverages external evidence outperforms the SEQ2SEQ model on both automatic and human evaluation. Besides, encoding the retrieved evidence helps SEQ2SEQ with both BLEU and METEOR, further confirming the necessities of the external information.

Decoding keyphrases helps argument generation. By first generating keyphrases, our model yields better BLEU and METEOR scores than the SEQ2SEQ comparisons. Allowing the argument decoder to attend generated keyphrase helps gain further improvements, highlighting the crucial role of content planning in this task.

Generation-based models compare unfavorably to retrieval model. Table 3.2 indicates that human prefer RETRIEVAL in all aspects. Closer inspections reveal that our model tends to capture stylistic language that are generic to the domain. Meanwhile the retrieved evidence sentences are human-edited, thus perceived to be more fluent and informative.

Chapter 4

Controllable Generation with Text Planning (Completed Work)

4.1 Motivation and Task Formulation

Neural sequential NLG models have achieved major breakthroughs in many tasks [3, 46, 66, 68, 69]. They differ from traditional systems in blending the planning and realization stages with the end-to-end training. While their outputs are usually impressively fluent, they are more inclined to be incoherent and unfaithful to the input [37, 73]. We aim to solve this problem by designing a two-step generation framework, that learns to produce sentence level content plans and reflect them in a surface realizer.

We consider a generic NLG setup: given an input statement $\boldsymbol{x}$, and a keyphrase bank $\mathcal{M}$, we aim to generate the output $\boldsymbol{y}$ with a content plan over $\mathcal{M}$. The content plan is implemented as sentence level keyphrase selection, conditioned on the input and its own selection history. Optionally, we assign linguistic style for each sentence, which further captures the domain-specific discourse goals. To show the generalizability of our framework, we experiment with three distinct domains, as detailed in the following sections.

4.2 Datasets

4.2.1 Argument Generation

Following our prior work [25, 27], we study the argument generation task on Reddit ChangeMyView. We consider the OP statement as input $\boldsymbol{x}$. For keyphrase bank $\mathcal{M}$, we first retrieve relevant passages from `commoncrawl.org`, which are then filtered based on whether their stances oppose the OP. We chunk the retained passages into noun phrases and verb phrases, from which the ones with topic signature word or is a Wikipedia title are extracted to form $\mathcal{M}$.

We additionally annotate each sentence with one of three argumentative discourse functions [40, 51]: CLAIM are propositions with one or two talking points; PREMISE are supporting arguments with reasoning and evidence; FUNCTIONAL are generic statements such as "*I under-*

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

Table 4.1: Statistics of the three datasets. On AGENDA, entities are extracted from abstract as keyphrases, hence all candidates are "selected".

stand what you said.". The annotation is done automatically with heuristics [1], which identifies 29.1% sentences as CLAIM, 62.2% and 8.7% as PREMISE and FUNCTIONAL, respectively.

4.2.2 Wikipedia Paragraph Generation

We study a second task of generating Wikipedia introduction paragraph. We create an aligned dataset from normal Wikipedia and its simplified version. The alignment is based on title matching, after removing the articles without a proper first paragraph of at least 10 words. The input comprises the article title and a binary simplicity style. For each title, we merge the keyphrases from both simple and normal version to form $\mathcal{M}$. Since the language style in Wikipedia is less diverse, therefore we consider sentence complexity as style, proxied by the length.

4.2.3 Paper Abstract Generation

The third task is generating abstracts for scientific papers given paper title and entities [1, 31]. Table 4.1 shows that this dataset is the smallest and has the longest output, the learning of sentence style thus becomes very difficult. Therefore we disable the style specification component.

4.3 Model

Input Encoding. The input $\boldsymbol{x}$ is first encoded with a biLSTM network. For each keyphrase in $\mathcal{M}$, we first calculate the keyphrase embedding $\boldsymbol{e}_k$ by summing up its words' GloVe embeddings [50]. Then, the embeddings of all keyphrases are fed into a biLSTM keyphrase reader, whose hidden states $\boldsymbol{h}^e_k$ is used as the final keyphrase representation. To facilitate learning to start and end, we always add special tokens `<START>` and `<END>` to $\mathcal{M}$.

Sentence-Level Content Planning. Given the keyphrase bank $\mathcal{M}$, the content planning model selects a subset of $\mathcal{M}$ for each sentence. We devise a LSTM f as content planning decoder, which consumes the sum of selected keyphrase representations for sentence $j-1$, and predicts

[1] We refer readers to our paper [26] for the complete set of rules.

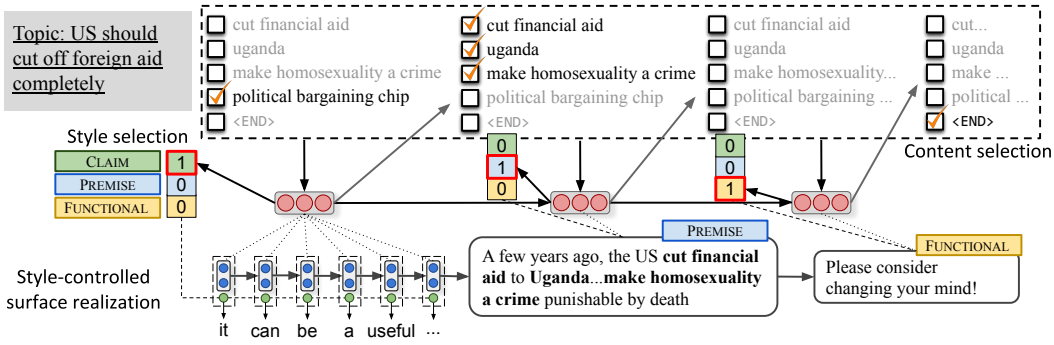

Figure 4.1: Overview of our proposed text planning based system.

selection for sentence j using an attention mechanism:

$$\boldsymbol{s}_j = f(\boldsymbol{s}_{j-1}, \sum_{k=1}^{|\mathcal{M}|} \boldsymbol{v}_{j,k} \boldsymbol{h}^e_k) \tag{4.1}$$

$$P(\boldsymbol{v}_{j+1,k}|\boldsymbol{v}_{1:j}) = \sigma(\boldsymbol{w}_v^\intercal \boldsymbol{s}_j + \boldsymbol{q}_j \boldsymbol{W}^c \boldsymbol{h}^e_k) \tag{4.2}$$

$$\boldsymbol{q}_j = (\sum_{r=0}^{j} \boldsymbol{v}_r)^\intercal \times [\boldsymbol{h}^e_1, \dots \boldsymbol{h}^e_{|\mathcal{M}|}] \tag{4.3}$$

$$\hat{\boldsymbol{t}}_j = \text{softmax}(\boldsymbol{w}_s^\intercal \tanh(\boldsymbol{W}^s[\boldsymbol{m}_j; \boldsymbol{s}_j])) \tag{4.4}$$

where $\boldsymbol{s}_j$ is the LSTM states, $\boldsymbol{v}_{j,k} \in \{0, 1\}$ indicates whether the k-th keyphrase is selected for sentence j. We maintain a selection history $\boldsymbol{q}_j$ that acts as query in predicting $\boldsymbol{v}_{j+1}$ (Eq (4.2)). A categorical variable $\boldsymbol{t}_j$ is predicted to capture the stylistic variations, conditioned on both the planner's hidden states and the summation of selected keyphrase representations (Eq (4.4)). During training, both the content selection and style specification incur cross-entropy terms that are jointly optimized with the realization model.

Surface Realization. Given the content plans and predicted style labels, the surface realizer specifies a conditional language model $P(y_t|y_{1:t-1}, \boldsymbol{s}_j, \boldsymbol{t}_k)$, implemented as a LSTM g:

$$\boldsymbol{z}_t = g(\boldsymbol{z}_{t-1}, \tanh(\boldsymbol{W}^{ws}\boldsymbol{s}_{J(t)} + \boldsymbol{W}^{ww}\boldsymbol{y}_{t-1})) \tag{4.5}$$

$$P(y_t|y_{1:t-1}, \boldsymbol{s}_j, \boldsymbol{t}_k) = \text{softmax}(\tanh(\boldsymbol{W}^o[\boldsymbol{z}_t; \boldsymbol{c}^w_t; \boldsymbol{c}^e_t; \boldsymbol{t}_{J(t)}])) \tag{4.6}$$

where $\boldsymbol{z}_t$ is the hidden states, $\boldsymbol{c}^w_t$ and $\boldsymbol{c}^e_t$ are context vectors from a bilinear attention from $\boldsymbol{z}_t$ to the input encoder states $\boldsymbol{h}_i$ and keyphrase memory bank $\boldsymbol{h}^e_k$. $\boldsymbol{W}^{**}$ are trainable parameters.

4.4 Experiments

4.4.1 Baselines and Comparisons.

We consider SEQ2SEQ with attention [3] as a baseline for all tasks, which encodes the input text concatenated with keyphrase bank as a sequence of tokens. For argument generation, our prior

	B-2	R-L	MTR	Len.	B-2	R-L	MTR	Len.
	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 4.4: Results on Wikipedia generation. Best results without oracle planning are in **bold**.

work [25], together with a RETRIEVAL baseline that returns external evidence sentences, are used for comparison. For Wikipedia, the RETRIEVAL baseline returns the most similar training set paragraph with the input title and keyphrases. The logistic regression model (LOGREGSEL) predicts keyphrase selection by global types. For abstract generation, the GRAPHWRITER [31] with rich knowledge graph encoding is also listed. To isolate the effect of content planning, we also experiment with the Oracle Plan setup, where the surface realizer has access to the gold-standard keyphrase selection and style labels.

4.4.2 Results and Discussions.

Argument Generation. Table 4.2 shows that our model outperforms all baselines on BLEU and ROUGE. When oralce plan is supplied, it further gains 3 BLEU points and achieves better METEOR than all comparisons, indicating the importance of content selection and ordering. Our model without style specification scores lower BLEU and METEOR.

	B-2	R-L	MTR	Len.
RETRIEVAL	7.81	15.68	**10.59**	150.0
SEQ2SEQ	3.64	19.00	9.85	51.7
H&W [25]	5.73	14.44	3.82	36.5
OURS (Oracle)	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

Table 4.2: Results on argument generation with BLEU-2, ROUGE-L, and METEOR. Best systems without oracle planning are in **bold**.

Wikipedia Generation. Results on Wikipedia generation are shown in Table 4.4, where our model again outperforms all comparisons in almost all metrics. We also observe a significant drop in ablated model, which implies the effectiveness of style usage in this domain. The oracle plan further improves our model with 3-5 BLEU and ROUGE points.

Abstract Generation. In Table 4.3 we compare with the state-of-the-art GRAPHWRITER model on AGENDA dataset. Our model with oracle plans achieve competitive results in all metrics, even without using any relation information. Our model also significantly outperforms the SEQ2SEQ baseline, which has the same input as ours.

	B-2	R-L	MTR	Len.
GRAPHWRITER	**29.95**	**28.56**	**19.90**	130.1
SEQ2SEQ	18.13	21.03	13.95	134.8
OURS (Oracle)	25.03	26.18	19.21	125.8
OURS	20.32	23.30	15.95	128.3

Table 4.3: Results on paper abstract generation.

Chapter 5

Improving Controllability for Long Text Generation (Proposed Work)

5.1 Motivation

One major issue with our systems so far is the lack of fine-grained control. This originates from both the potential insufficiency in keyphrase-only content modeling, and the sequential surface realization stage. For instance, given the same keyphrase set {"*multi-faith society*", "*people*", "*religion*", "*anything illegal*", "*obey the law*"}, the human written sentence and one of our systems' output are:

- *In a multi-faith society, people have the right to practise their religion as long as they obey the law and don't do anything illegal.* (Human)
- *In a multi-faith society where people are forced to follow their religion, they are forced to obey the law, and if they do anything illegal they will be killed.* (Model)

The keyphrase set of this example is reasonably detailed and covers most content. But our model's output deviates from the desired discourse function (to express the belief of the author) of the human written sentence. It is clear to us that, even with correctly predicted content plans and a content faithful realization, the generation fails to deliver the communicative goals without understanding the discourse role for each sentence.

Furthermore, as noted in early chapters, another major challenge for neural NLG models is global coherence and cohesiveness over long output. Even the widely circulated article generated by GPT-2 model still exhibits inconsistencies across paragraphs. We believe the controllability on discourse level will be the key for both the above two questions.

5.2 Fine-Grained Discourse Function

Many traditional text generation systems explicitly model the discourse relation between messages. For example, a soccer report usually starts with a general description and outcome of the game, then expands into major events in chronological order. Defining the set of common discourse roles is crucial to produce more structured output, but is highly domain-dependent and

can require substantial expert knowledge. This naturally hinders the use of data-driven models. In our past work [26] sentence style specification serves a similar purpose. However we find the conditional generation setup with only limited interaction between realization and style variable to be suboptimal. In our proposal, we aim to leverage unsupervised discourse role labeling method, and investigate its effect over the word level realization.

5.3 Transfer from Large Pre-trained Model with Refinement

Large Transformer models pre-trained over heterogeneous corpora are shown to be extremely effective in language modeling [14, 53]. They can also be adapted for NLG tasks, the quality of which often depends on the decoding strategies. The straightforward sampling based autoregressive decoding are efficient solution for domains without complicated discourse structures and planning, such as machine translation and dialogue generation. While for tasks such as argument generation and news article generation, it is extremely difficult to find a satisfactory sequence in one shot. A critical next step is thus to refine the generation with desired constraints from either content or discourse level.

5.4 Timeline and Milestones

Feb 2020	(1) Implement and evaluate one-pass based generation model with latent variable controlling the discourse function and template generation. (2) Prepare data in different domains for generation, such as news, Wikipedia.
Mar 2020	(1) Thesis comprehensive exam. (2) Thesis proposal. (3) Implement and evaluate two-pass and non-autoregressive based models to control the discourse function and template. Compare with the one-pass based method. (4) Experiment the refinement-based generation on different domains.
Apr - May 2020	(1) Experiments and paper writing on refinement-based generation for EMNLP 2020. (2) Finish PhD course requirement and submit for candidacy reviews.
Jun - Aug 2020	(1) Propose argument generation models on persuasion, and explore other text genres in argumentation, such as debate transcripts. (2) Toolkit building for our argument generation model that can be deployed online.
Sep - Dec 2020	(1) Submit work to ACL 2021. (2) Start thesis writing.
Jan - Mar 2021	(1) Schedule dissertation defense. (2) Thesis writing.
Mar - Jun 2021	(1) Finish and submit thesis.

Bibliography

[1] 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. 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, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-3011. URL https://www.aclweb.org/anthology/N18-3011. 4.2.3

[2] Richard Andrews. Critical thinking and/or argumentation in higher education. In *The Palgrave handbook of critical thinking in higher education*, pages 49–62. Springer, 2015. 1.2

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014. 1.1, 2.1, 2.2, 3.3.3, 4.1, 4.4.1

[4] Roy Bar-Haim, Indrajit Bhattacharya, Francesco Dinuzzo, Amrita Saha, and Noam Slonim. 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, April 2017. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/E17-1024. 1.2

[5] Regina Barzilay and Lillian Lee. 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, May 2 - May 7 2004. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/N04-1015. 2.1

[6] John A Bateman. Enabling technology for multilingual natural language generation: the kpml development environment. *Natural Language Engineering*, 3(1):15–55, 1997. 1.1, 2.1

[7] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, edi-

tors, *Advances in Neural Information Processing Systems 28*, pages 1171–1179. Curran Associates, Inc., 2015. URL http://papers.nips.cc/paper/5956-scheduled-sampling-for-sequence-prediction-with-recurrent-neural-n pdf. 1.1

[8] Filip Boltužić and Jan Šnajder. Identifying prominent arguments in online debates using semantic textual similarity. In *Proceedings of the 2nd Workshop on Argumentation Mining*, pages 110–115, Denver, CO, June 2015. Association for Computational Linguistics. doi: 10.3115/v1/W15-0514. URL https://www.aclweb.org/anthology/W15-0514. 1.2

[9] Jianpeng Cheng and Mirella Lapata. Neural summarization by extracting sentences and words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 484–494, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1046. URL https://www.aclweb.org/anthology/P16-1046. 2.2

[10] Cary Coglianese. E-rulemaking: Information technology and the regulatory process. *Admin. L. Rev.*, 56:353, 2004. 1.2

[11] Emilie Colin, Claire Gardent, Yassine M'rabet, Shashi Narayan, and Laura Perez-Beltrachini. The WebNLG challenge: Generating text from DBPedia data. In *Proceedings of the 9th International Natural Language Generation conference*, pages 163–167, Edinburgh, UK, September 5-8 2016. Association for Computational Linguistics. doi: 10.18653/v1/W16-6626. URL https://www.aclweb.org/anthology/W16-6626. 2.1

[12] Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. Plug and play language models: a simple approach to controlled text generation. *arXiv preprint arXiv:1912.02164*, 2019. 2.2, 2.3

[13] Michael Denkowski and Alon Lavie. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 376–380, Baltimore, Maryland, USA, June 2014. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W14-3348. 1.3, 3.4.1, 3.4.3

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://www.aclweb.org/anthology/N19-1423. 1.1, 2.2, 5.3

[15] Pablo Ariel Duboue and Kathleen R. McKeown. 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, 2003. URL https://www.aclweb.org/anthology/W03-1016. 2.1

[16] Ondřej Dušek, Jekaterina Novikova, and Verena Rieser. Findings of the E2E NLG Challenge. In *Proceedings of the 11th International Conference on Natural Language Generation*, Tilburg, The Netherlands, 2018. URL https://arxiv.org/abs/1810.01170. arXiv:1810.01170. 2.1

[17] Roxanne El Baff, Henning Wachsmuth, Khalid Al Khatib, Manfred Stede, and Benno Stein. Computational argumentation synthesis as a language modeling task. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 54–64, Tokyo, Japan, October–November 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-8607. URL https://www.aclweb.org/anthology/W19-8607. 1.2

[18] Michael Elhadad and Jacques Robin. An overview of surge: A reusable comprehensive syntactic realization component. 1996. 1.1, 2.1

[19] Jessica Ficler and Yoav Goldberg. Controlling linguistic style aspects in neural language generation. In *Proceedings of the Workshop on Stylistic Variation*, pages 94–104, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-4912. URL https://www.aclweb.org/anthology/W17-4912. 2.3

[20] Julie Gainsburg, John Fox, and Lawrence M Solan. Argumentation and decision making in professional practice. *Theory Into Practice*, 55(4):332–341, 2016. 1.2

[21] Sergey Golovanov, Rauf Kurbanov, Sergey Nikolenko, Kyryl Truskovskyi, Alexander Tselousov, and Thomas Wolf. Large-scale transfer learning for natural language generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6053–6058, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1608. URL https://www.aclweb.org/anthology/P19-1608. 2.2

[22] Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, Jürgen Schmidhuber, et al. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, 2001. 2.2

[23] Eduard Hovy. Generating natural language under pragmatic constraints. *Journal of Pragmatics*, 11(6):689–719, 1987. 1.1, 2.1

[24] Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1587–1596. JMLR. org, 2017. 2.3

[25] Xinyu Hua and Lu Wang. 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, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1021. URL https://www.aclweb.org/anthology/P18-1021. 3.3, 4.2.1, 4.4.1, 4.4.2

[26] Xinyu Hua and Lu Wang. 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, Novem-

ber 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1055. URL https://www.aclweb.org/anthology/D19-1055. 1, 5.2

[27] Xinyu Hua, Zhe Hu, and Lu Wang. 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, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1255. URL https://www.aclweb.org/anthology/P19-1255. 4.2.1

[28] Maria-Pilar Jime'nez-Aleixandre. Knowledge producers or knowledge consumers? argumentation and decision making about environmental management. *International Journal of Science Education*, 24(11):1171–1190, 2002. 1.2

[29] Antonis Kakas and Pavlos Moraitis. Argumentation based decision making for autonomous agents. In *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pages 883–890, 2003. 1.2

[30] Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*, 2019. 2.3

[31] Rik Koncel-Kedziorski, Dhanush Bekal, Yi Luan, Mirella Lapata, and Hannaneh Hajishirzi. Text Generation from Knowledge Graphs with Graph Transformers. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2284–2293, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1238. URL https://www.aclweb.org/anthology/N19-1238. 1.3, 4.2.3, 4.4.1

[32] Ioannis Konstas and Mirella Lapata. Inducing document plans for concept-to-text generation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1503–1514, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D13-1157. 3.3.2

[33] Ioannis Konstas, Srinivasan Iyer, Mark Yatskar, Yejin Choi, and Luke Zettlemoyer. Neural AMR: Sequence-to-sequence models for parsing and generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 146–157, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1014. URL https://www.aclweb.org/anthology/P17-1014. 3.3.2

[34] Carolin Lawrence, Bhushan Kotnis, and Mathias Niepert. 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, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1001. URL https://www.aclweb.org/anthology/D19-1001. 2.2

[35] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 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, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1014. URL https://www.aclweb.org/anthology/N16-1014. 1.1

[36] Jiwei Li, Michel Galley, Chris Brockett, Georgios Spithourakis, Jianfeng Gao, and Bill Dolan. A persona-based neural conversation model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 994–1003, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1094. URL https://www.aclweb.org/anthology/P16-1094. 2.2

[37] Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. Adversarial learning for neural dialogue generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2157–2169, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1230. URL https://www.aclweb.org/anthology/D17-1230. 4.1

[38] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, 2004. URL http://aclweb.org/anthology/W04-1013. 1.3

[39] Chin-Yew Lin and Eduard Hovy. The automated acquisition of topic signatures for text summarization. In *COLING 2000 Volume 1: The 18th International Conference on Computational Linguistics*, 2000. URL https://www.aclweb.org/anthology/C00-1072. 3.3.1

[40] Marco Lippi and Paolo Torroni. Argumentation mining: State of the art and emerging trends. *ACM Transactions on Internet Technology (TOIT)*, 16(2):1–25, 2016. 4.2.1

[41] Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 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, June 2014. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P14-5010. 3.3.2

[42] Kathleen McKeown. *Text generation*. Cambridge University Press, 1992. 1.1, 2.1

[43] Susan W McRoy, Songsak Channarukul, and Syed S Ali. Yag: A template-based generator for real-time systems. In *Proceedings of the first international conference on Natural language generation-Volume 14*, pages 264–267. Association for Computational Linguistics, 2000. 2.1

[44] Johanna D. Moore and Cecile L. Paris. Planning text for advisory dialogues: Capturing intentional and rhetorical information. *Computational Linguistics*, 19(4):651–694, 1993. URL https://www.aclweb.org/anthology/J93-4004. 1.1, 2.1

[45] Jann Müller and Anthony Hunter. An argumentation-based approach for decision making. In *2012 IEEE 24th International Conference on Tools with Artificial Intelligence*, volume 1, pages 564–571. IEEE, 2012. 1.2

[46] Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*, 2016. 1.1, 4.1

[47] Samuel Phillips Newman. *A practical system of rhetoric*. Newman and Ivison, 1851. 3.1

[48] Vlad Niculae, Joonsuk Park, and Claire Cardie. 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, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1091. URL https://www.aclweb.org/anthology/P17-1091. 1.2

[49] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL https://www.aclweb.org/anthology/P02-1040. 1.3, 3.4.3

[50] Jeffrey Pennington, Richard Socher, and Christopher Manning. 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, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1162. URL https://www.aclweb.org/anthology/D14-1162. 4.3

[51] Isaac Persing and Vincent Ng. 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, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1164. URL https://www.aclweb.org/anthology/N16-1164. 4.2.1

[52] Ratish Puduppully, Li Dong, and Mirella Lapata. Data-to-text generation with content selection and planning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6908–6915, 2019. 2.1

[53] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019. 1.1, 2.2, 5.3

[54] Owen Rambow and Tanya Korelsky. Applied text generation. In *Proceedings of the Third Conference on Applied Natural Language Processing*, pages 40–47, Trento, Italy, March 1992. Association for Computational Linguistics. doi: 10.3115/974499.974508. URL https://www.aclweb.org/anthology/A92-1006. 1.1

[55] Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL http://arxiv.org/abs/1511.06732. 1.1

[56] Paul Reisert, Naoya Inoue, Naoaki Okazaki, and Kentaro Inui. A computational approach for generating toulmin model argumentation. In *Proceedings of the 2nd Workshop on*

Argumentation Mining, pages 45–55, Denver, CO, June 2015. Association for Computational Linguistics. doi: 10.3115/v1/W15-0507. URL https://www.aclweb.org/anthology/W15-0507. 3.1

[57] Ehud Reiter, Chris Mellish, and John Levine. Automatic generation of technical documentation. *Applied Artificial Intelligence an International Journal*, 9(3):259–287, 1995. 2.1

[58] Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. Object hallucination in image captioning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4035–4045, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1437. URL https://www.aclweb.org/anthology/D18-1437. 1.1

[59] Alexander M. Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1044. URL https://www.aclweb.org/anthology/D15-1044. 2.1, 2.2

[60] Mansour Saffar Mehrjardi, Amine Trabelsi, and Osmar R. Zaiane. Self-attentional models application in task-oriented dialogue generation systems. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 1031–1040, Varna, Bulgaria, September 2019. INCOMA Ltd. doi: 10.26615/978-954-452-056-4_119. URL https://www.aclweb.org/anthology/R19-1119. 2.2

[61] Judith A Sanders, Richard L Wiseman, and Robert H Gass. Does teaching argumentation facilitate critical thinking? *Communication Reports*, 7(1):27–35, 1994. 1.2

[62] Misa Sato, Kohsuke Yanai, Toshinori Miyoshi, Toshihiko Yanase, Makoto Iwayama, Qinghua Sun, and Yoshiki Niwa. End-to-end argument generation system in debating. In *Proceedings of ACL-IJCNLP 2015 System Demonstrations*, pages 109–114, Beijing, China, July 2015. Association for Computational Linguistics and The Asian Federation of Natural Language Processing. doi: 10.3115/v1/P15-4019. URL https://www.aclweb.org/anthology/P15-4019. 1.2, 3.1

[63] Abigail See, Peter J. Liu, and Christopher D. Manning. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1099. URL https://www.aclweb.org/anthology/P17-1099. 2.2

[64] Abigail See, Aneesh Pappu, Rohun Saxena, Akhila Yerukola, and Christopher D. Manning. Do massively pretrained language models make better storytellers? In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 843–861, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/K19-1079. URL https://www.aclweb.org/anthology/K19-1079. 2.2

[65] Linfeng Song, Yue Zhang, Zhiguo Wang, and Daniel Gildea. A graph-to-sequence model for AMR-to-text generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1616–1626, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1150. URL https://www.aclweb.org/anthology/P18-1150. 3.3.2

[66] Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. 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, May–June 2015. Association for Computational Linguistics. doi: 10.3115/v1/N15-1020. URL https://www.aclweb.org/anthology/N15-1020. 1.1, 2.2, 4.1

[67] Elior Sulem, Omri Abend, and Ari Rappoport. Simple and effective text simplification using semantic and neural methods. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 162–173, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1016. URL https://www.aclweb.org/anthology/P18-1016. 2.3

[68] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014. 1.1, 2.1, 2.2, 4.1

[69] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 2.2, 4.1

[70] Henning Wachsmuth, Martin Potthast, Khalid Al-Khatib, Yamen Ajjour, Jana Puschmann, Jiani Qu, Jonas Dorsch, Viorel Morari, Janek Bevendorff, and Benno Stein. Building an argument search engine for the web. In *Proceedings of the 4th Workshop on Argument Mining*, pages 49–59, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-5106. URL https://www.aclweb.org/anthology/W17-5106. 3.1

[71] Henning Wachsmuth, Shahbaz Syed, and Benno Stein. 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, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1023. URL https://www.aclweb.org/anthology/P18-1023. 3.1

[72] Alex Wang and Kyunghyun Cho. BERT has a mouth, and it must speak: BERT as a Markov random field language model. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pages 30–36, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-2304. URL https://www.aclweb.org/anthology/W19-2304. 2.2

[73] Sam Wiseman, Stuart Shieber, and Alexander Rush. Challenges in data-to-document generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural*

Language Processing, pages 2253–2263, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1239. URL https://www.aclweb.org/anthology/D17-1239. 1.1, 2.1, 2.3, 4.1

[74] Sam Wiseman, Stuart M Shieber, and Alexander M Rush. Learning neural templates for text generation. *arXiv preprint arXiv:1808.10122*, 2018. 2.1

[75] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*, 2019. 1.1, 2.2