Controllable Summarization with Constrained Markov Decision Process

Hou Pong Chan[1], Lu Wang[2], and Irwin King[3]

[1]University of Macau, Macau SAR, China
[2]University of Michigan, Ann Arbor, MI, USA
[3]The Chinese University of Hong Kong, Hong Kong SAR, China
[1]hpchan@um.edu.mo
[2]wangluxy@umich.edu
[3]king@cse.cuhk.edu.hk

Abstract

We study *controllable text summarization*, which allows users to gain control on a particular attribute (e.g., length limit) of the generated summaries. In this work, we propose a novel training framework based on *Constrained Markov Decision Process* (CMDP), which conveniently includes a *reward function* along with a set of *constraints*, to facilitate better summarization control. The reward function encourages the generation to resemble the human-written reference, while the constraints are used to explicitly prevent the generated summaries from violating user-imposed requirements. Our framework can be applied to control important attributes of summarization, including *length*, *covered entities*, and *abstractiveness*, as we devise specific constraints for each of these aspects. Extensive experiments on popular benchmarks show that our CMDP framework helps generate informative summaries while complying with a given attribute's requirement.[1]

1 Introduction

Text summarization aims to condense the information of an input document into a concise summary. Although recently neural abstractive summarization models have achieved promising performance (See et al., 2017; Paulus et al., 2018), they do not allow users to indicate their preference to control different aspects of the generated summaries. Controllable summarization has many use cases. For instance, it can summarize product descriptions to fit within a word limit in online advertising. In another example, teachers can demonstrate the technique of paraphrasing important information by showing a system-generated summary with high abstractiveness. Controllable summarization can also complement information retrieval systems, for example, to only generate summaries covering the entities that users are interested in. Figure 1 illustrates one such usage, where our proposed model produces distinct abstractive summaries of the same source document, focusing on different input entities.

To allow users to control a particular attribute of the generated summaries, Fan et al. (2018) proposed a token-based controllable summarization model (ControlSum). Although ControlSum incorporates control tokens that let users specify a requirement on a summary attribute, the maximum likelihood training objective of the model does not provide explicit supervision signals that prevent the model from violating the specified attribute requirement. Consequently, a substantial portion of the generated summaries still fail to meet the specified attribute requirement as shown in our experiments.

One possible solution to enforce the attribute requirement is to apply reinforcement learning (RL) with Markov Decision Process (MDP) (Bellman, 1957) to optimize a weighted sum of reward functions, including a penalty function to penalize the violation of the attribute requirement, and a summarization metric to encourage the generated summaries to be consistent with the references. However, selecting appropriate weights for different reward functions is a delicate task, and requires intensive hyperparameter tuning.

In this work, we argue that applying constraints on the training objective is a more convenient way to control an attribute of a summary, since it avoids tuning reward function weights. We formulate the problem of training controllable text summarization models as a constrained Markov Decision

---

[1]Our source code is available at `https://github.com/kenchan0226/control-sum-cmdp`.

> **Document:**
> Marseille, France (CNN) The French prosecutor leading an investigation into the crash of Germanwings Flight 9525 insisted Wednesday that he was not aware of any video footage from on board the plane. Marseille prosecutor Brice Robin, in charge of the criminal inquiry into the crash, told CNN that "so far no videos were used in the crash investigation." ... Robin's comments follow claims by two publications, German daily Bild and French Paris Match, of a cell phone video showing the harrowing final seconds from on board the flight as it crashed into the French Alps on March 24 ... Paris Match and Bild reported that the video was recovered from a phone at the wreckage site. The two publications described ...
>
> **Summaries:**
> **[Germanwings Flight 9525]** Germanwings Flight 9525 crashed into the French Alps.
> **[Brice Robin]** Prosecutor Brice Robin says no video footage was used in the crash investigation.
> **[French Paris Match]** French Paris Match says video was found.

Figure 1: A sample document and three summaries generated by our entity-controlled model based on DistilGPT2 (Sanh et al., 2019) and fine-tuned by our proposed method. Each summary corresponds to the requested entity inside the pair of brackets.

Process (CMDP) (Altman, 1999), a RL framework trained with both rewards and constraints. In this setup, we maximize a summarization metric to encourage the similarity between the output summaries and the references, as well as impose constraints to disallow the summaries from violating a specified attribute requirement.

Moreover, we apply our approach to improve token-based controllable summarization models and control important summary attributes including length, covered entities, and abstractiveness by creating specific constraints for each attribute. For length control, we divide summary length into disjoint length bins and restrict the summary length according to the desired length bin. For entity control, we design constraints that guide the generated summary to cover the salient information of user-specified entities. To control abstractiveness, which measures the degree of textual novelty between a summary and its input document, we define bins corresponding to three abstractiveness levels, and design constraints that allow users to control the summary's abstractiveness.

Extensive experiments are conducted on popular benchmarks, to evaluate the effectiveness of our CMDP training framework with different types of attribute requirements. Concretely, we use our CMDP framework to fine-tune controllable summarization models based on pointer-generator network (See et al., 2017), a Recurrent Neural Network (RNN) (Hochreiter and Schmidhuber,

1997) model, and DistilGPT2 (Sanh et al., 2019), a large-scale pre-trained Transformer (Vaswani et al., 2017) model.[2] Experiment results demonstrate that our approach consistently improves both controllable summarization models' capabilities of following the specified attribute requirement. In addition, our framework increases the ROUGE scores of the generated summaries when provided with the reference control tokens (e.g., the tokens that represent the entities in the reference summary). Human evaluations further confirm that our framework produces informative summaries that conform to the attribute requirement.

The key contributions of this paper include: (1) A novel training framework that provides explicit guidance signals to supervise a controllable summarization model to conform to the specified attribute requirement; (2) Constraints that allow users to control the length, covered entities, and the abstractiveness of the generated summaries, respectively; (3) Consistent performance improvement of controllable summarization models based on two different architectures.

2 Related Work

Summarization Systems with Specified Attributes. Several methods extend abstractive summarization models to allow users to control a specific attribute of summaries. Fan et al. (2018) propose a method that allows users to control an attribute such as length, entity, and style of summaries by prepending special tokens to the input document. Liu et al. (2018) focus on controlling the exact length of summaries. They multiply the input word embeddings in the decoder by the specified summary length. Song et al. (2020) propose a masked language model to control the portion of copied words in the output summary for the sentence summarization task. This model controls the abstractiveness of a summary at the word level. In contrast, our work controls the extractive fragment density (Grusky et al., 2018) of the output summary, which restricts the abstractiveness at the fragment level. Makino et al. (2019) and Laban et al. (2020) incorporate a penalty term on the training objective to penalize a model for violating the length requirement for word limit control. However, it requires hyperparameter tuning for the weight of penalty if one wants to apply their method to another dataset. Our approach imposes

---

[2]We choose DistilGPT2 since it is smaller than GPT2.

constraints on the training objective and does not need to search suitable weights for penalties based on human inspection.

Query-focused summarization aims to predict a summary that answers specific questions, for example, ''*How often did Lebron James visit his hometown?*''. Most of the query-focused summarization methods are extractive and they are based on centrality ranking (Wan, 2008; Wan and Zhang, 2014), manifold-ranking (Wan et al., 2007; Wan and Xiao, 2009; Wan, 2009), or sentence-compression framework (Wang et al., 2013). Recently, Nema et al. (2017) propose two query attention-based models for abstractive query-focused summarization. On the other hand, entity-controlled summarization aims to produce a summary that captures the salient information of the desired entities, for example, ''Lebron James''.

Abstractive Summarization. Most of the existing abstractive summarization models (Gehrmann et al., 2018; Zhang et al., 2020a; Chan et al., 2020) are built on the encoder-decoder model (Bahdanau et al., 2015) to generate summaries. See et al. (2017) propose the pointer-generator network, which allows copying words from the source to the output summary. The structure-infused copy mechanism (Song et al., 2018) incorporates the syntactic structure of the source text into the pointer-generator network to facilitate copying important words to the output summary. Lebanoff et al. (2019) propose a summarization framework that first extracts either a single sentence or a pair of sentences from the source document, then it condenses or fuses the selected sentence(s) to generate a summary. The above models do not allow users to constrain the degree of copying nor sentence fusion from the source document.

Recent methods apply RL with MDP to optimize an abstractive summarization model towards a single or a weighted sum of reward functions. Several methods (Paulus et al., 2018; Çelikyilmaz et al., 2018) adopt the ROUGE-L score (Lin, 2004) as the reward function. The SENECA model (Sharma et al., 2019) optimizes a weighted sum of ROUGE-2, ROUGE-L, and a coherence score from a coherence model. To improve the factual correctness of the generated summaries, several methods (Huang et al., 2020; Zhang et al., 2020c) use RL to maximize a weighted sum of ROUGE scores and a factual correctness score computed by

a model. Kryscinski et al. (2018) use the weighted sum of ROUGE-L and 3-gram novelty as the reward to increase the abstractiveness of summaries, but this method does not allow users to control the abstractiveness level of summaries. Pasunuru and Bansal (2018) extend the ROUGE-L reward by up-weighting the salient words detected by a classifier. One can modify this word-level weighting scheme to encourage the summary to contain certain keywords, but this method does not explicitly encourage the model to generate relevant information about the keywords. In contrast, we design a constraint to enforce a summary to retain relevant information of the requested entities. Ziegler et al. (2020) apply RL to fine-tune a GPT2 model (Radford et al., 2019). The reward is provided by a model trained from human preferences on different summaries. Though one can use a weighted sum of rewards to control an attribute of generated summaries, such a method needs to tune the weights for rewards. Our CMDP approach avoids the tuning of such weights.

Controllable Text Generation. Controllable text generation has received increasing attention from researchers. In machine translation, several methods (Sennrich et al., 2016; Kobus et al., 2017; Takeno et al., 2017) apply special tokens to control the politeness, domain, or length of the translation output. Ficler and Goldberg (2017) concatenate a style embedding with the decoder input to control the style of the generated review. Kikuchi et al. (2016), Miao et al. (2019), and Schumann et al. (2020) introduce different techniques to control sentence length for the headline generation task, such as feeding a length embedding to the decoder. The label-fine-tuning (LFT) model (Niu and Bansal, 2018) uses special tokens to control the politeness of responses for dialogue response generation. Several insertion-based decoding methods (Sun et al., 2017; Zhu et al., 2019; Gu et al., 2019) are proposed to complete a fill-in-the-blank sentence, for example, ''*keywords 1 __ keywords 2 __*''. These decoding methods can be used to enforce the output to contain certain keywords, but users need to specify the relative order among the keywords. In contrast, entity-controlled summarization lets the model determine the relative order among the requested entities. Recently, Keskar et al. (2019) train a large language model conditioned on control codes that specify particular attributes such as domain or

language style. Compared with the above methods, our approach incorporates the attribute requirement into the training objective, which gives more explicit supervision signals to the summarizer.

3 Controllable Summarization with Constrained Markov Decision Process

3.1 Problem Definition

Given a text document $\mathbf{x}$ and a requirement on an attribute a (e.g., length limit of 20 words), the goal of *controllable text summarization* is to generate a summary $\mathbf{y}$ that satisfies the requirement. Both the input document and output summary are sequences of words, namely, $\mathbf{x} = [x_1, \ldots, x_{l_\mathbf{x}}]$ and $\mathbf{y} = [y_1, \ldots, y_{l_\mathbf{y}}]$, where $l_\mathbf{x}$ and $l_\mathbf{y}$ are the numbers of words in $\mathbf{x}$ and $\mathbf{y}$, respectively. In this work, we focus on single-document summarization.

3.2 Constrained Markov Decision Process Formulation

We propose a constrained Markov Decision Process (CMDP) approach to guide a controllable summarization model to follow the attribute requirement. Assume an agent interacts with an environment to generate a summary in discrete time steps. At each step t, the agent performs an action by sampling a word y_t from its policy $\pi_{\boldsymbol{\theta}}$, which is a controllable summarization model. Then the agent updates its internal state representation (hidden state of the decoder) and proceeds to the next step. Once the agent produces the end-of-sequence (EOS) token, we denote the current time step as T, the environment gives a reward $r(y_1, \ldots, y_T, \mathbf{y}^*, \mathbf{x})$, and a set of costs $c_i(y_1, \ldots, y_T, \mathbf{y}^*, \mathbf{x})$ to the agent. The process then terminates. The reward function r measures the similarity between the output summary $[y_1, \ldots, y_T]$ and the reference summary $\mathbf{y}^*$, while a cost function c_i measures how well a summary satisfies an attribute requirement, for example, we can define a *length cost function* to measure the difference between the output summary length l_y and the specified length limit l: $l_y - l$. The goal of the agent is to maximize the expected reward while ensuring the costs are under constraints as follows:

$$\max_{\pi_{\boldsymbol{\theta}}} \quad \mathbb{E}_{\mathbf{y}_{1:T} \sim \pi_{\boldsymbol{\theta}}}[r(\mathbf{y}_{1:T}, \mathbf{y}^*, \mathbf{x})],$$
$$\text{s.t.} \quad \mathbb{E}_{\mathbf{y}_{1:T} \sim \pi_{\boldsymbol{\theta}}}[c_i(\mathbf{y}_{1:T}, \mathbf{y}^*, \mathbf{x})] \leq \alpha_i, \quad (1)$$
$$i = 1, \ldots, m,$$

where $\mathbf{y}_{1:T}$ denotes $y_1, \ldots, y_T$, α_i is a pre-defined threshold associated with cost function c_i, m is the size of the set of constraints. A constraint restricts an attribute of the generated summary. For example, to limit the summary length, we can define a constraint to enforce the length cost function to be no larger than 0, $l_y - l \leq 0$.

Lagrange Relaxation. Following Tessler et al. (2019), we apply the Lagrange relaxation technique (Bertsekas, 1997) to approximate the constrained optimization problem in Eq. (1). We use $J(\pi_{\boldsymbol{\theta}})$ as a shorthand to denote $\mathbb{E}_{\mathbf{y}_{1:T} \sim \pi_{\boldsymbol{\theta}}}[r(\mathbf{y}_{1:T}, \mathbf{y}^*, \mathbf{x})]$ and use $J_{c_i}(\pi_{\boldsymbol{\theta}})$ to denote $\mathbb{E}_{\mathbf{y}_{1:T} \sim \pi_{\boldsymbol{\theta}}}[c_i(\mathbf{y}_{1:T}, \mathbf{y}^*, \mathbf{x})]$. We then define a Lagrangian function $L(\boldsymbol{\lambda}, \boldsymbol{\theta}) = J(\pi_{\boldsymbol{\theta}}) - \sum_{i=1}^{m} \lambda_i(J_{c_i}(\pi_{\boldsymbol{\theta}}) - \alpha_i)$, where λ_i is a Lagrangian multiplier and $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_m] \in \mathbb{R}^m$. When $\lambda_i \geq 0$, $\forall i$, the optimal value of $\max_{\boldsymbol{\theta}} L(\boldsymbol{\lambda}, \boldsymbol{\theta})$ is an upper bound to the optimal value of Eq. (1). If we minimize the optimal value of $\max_{\boldsymbol{\theta}} L(\boldsymbol{\lambda}, \boldsymbol{\theta})$, we will obtain a tighter upper bound on the optimal value of Eq. (1). Thus, we approximate Eq. (1) by the following relaxed problem:

$$\min_{\boldsymbol{\lambda} \succeq 0} \max_{\boldsymbol{\theta}} J(\pi_{\boldsymbol{\theta}}) - \sum_{i=1}^{m} \lambda_i(J_{c_i}(\pi_{\boldsymbol{\theta}}) - \alpha_i), \quad (2)$$

where $\boldsymbol{\lambda} \succeq 0$ denotes that every entry in $\boldsymbol{\lambda}$ is non-negative. Intuitively, this relaxed problem penalizes the behavior of violating the constraints, and all the Lagrange multipliers λ_i are learnable. In contrast, the MDP formulation requires the manual tuning of weights for penalty terms.

Policy Training. We optimize $\boldsymbol{\theta}$ and $\boldsymbol{\lambda}$ alternatively using gradient ascent and descent: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \eta_1 \nabla_{\boldsymbol{\theta}} L(\boldsymbol{\lambda}, \boldsymbol{\theta})$, $\boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda} - \eta_2 \nabla_{\boldsymbol{\lambda}} L(\boldsymbol{\lambda}, \boldsymbol{\theta})$, where η_1 and η_2 are learning rates for $\boldsymbol{\theta}$ and $\boldsymbol{\lambda}$ respectively. The gradients are expressed as follows. $\nabla_{\boldsymbol{\theta}} L = \mathbb{E}_{\pi_{\boldsymbol{\theta}}}[\sum_{t=0}^{T} \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(y_t | \mathbf{y}_{1:t-1})(r - \boldsymbol{\lambda}^T \mathbf{c})]$, $\nabla_{\boldsymbol{\lambda}} L = -(\mathbb{E}_{\pi_{\boldsymbol{\theta}}}[\mathbf{c}] - \boldsymbol{\alpha})$, where $\mathbf{c} = [c_1, \ldots, c_m] \in \mathbb{R}^m$, $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_m] \in \mathbb{R}^m$, $\mathbb{E}_{\pi_{\boldsymbol{\theta}}}$ is a shorthand for $\mathbb{E}_{\mathbf{y}_{1:T} \sim \pi_{\boldsymbol{\theta}}}$. Since it is intractable to enumerate all possible $\mathbf{y}_{1:T}$, we approximate the expectation $\mathbb{E}_{\mathbf{y}_{1:T} \sim \pi_{\boldsymbol{\theta}}}$ using a sample of output sequence $\mathbf{y}_{1:T} \sim \pi_{\boldsymbol{\theta}}$. Moreover, we also subtract the reward by a baseline b, which is a standard technique to reduce the variance of the gradient estimator (Sutton and Barto, 1998). The gradients are then estimated by:

$$\nabla_{\boldsymbol{\theta}} L \approx \sum_{t=0}^{T} \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(y_t | \mathbf{y}_{1:T})(r - \boldsymbol{\lambda}^T \mathbf{c} - b), \quad (3)$$

$$\nabla_{\boldsymbol{\lambda}} L \approx -(\mathbf{c} - \boldsymbol{\alpha}). \quad (4)$$

We can interpret $\nabla_\theta L$ as the standard policy gradient with a regularization term $-\lambda^T \mathbf{c}$, where λ is trained by a gradient descent algorithm.

In this work, we apply the self-critical baseline (Rennie et al., 2017). Specifically, we use greedy search to generate an output sequence $\bar{\mathbf{y}}$ from the policy. Then, we treat the reward of this sequence $r(\bar{\mathbf{y}}, \mathbf{y}^*, \mathbf{x})$ as the baseline b.

Reward Function. We apply **BERTScore** (Zhang et al., 2020b) as the reward function to measure the similarity between an output summary and the reference summary based on their BERT (Devlin et al., 2019) contextual embeddings. We do not use ROUGE scores (Lin, 2004) as the reward since they cannot match paraphrases in an output.

3-gram Repetition Constraint. Similar to prior work (Paulus et al., 2018; Liu and Lapata, 2019; Laban et al., 2020), we address the problem of repetition of text fragments by adding a 3-gram repetition constraint into our framework. We define a cost function that measures the ratio of 3-gram repetition in a summary: $\text{RepeatRatio}_3(\mathbf{y}) = \#repeat\ 3\text{-}gram/\#\ 3\text{-}gram$. Then we set its threshold to zero and apply the following 3-gram repetition constraint: $\text{RepeatRatio}_3(\mathbf{y}) \leq 0$.

3.3 Implementation with RNN and Pre-trained Transformer

We apply our CMDP framework to train two types of controllable summarization models: pointer-generator network (See et al., 2017) and Distil-GPT2 (Sanh et al., 2019). The pointer-generator network is a popular abstractive summarization model based on RNN encoder-decoder model (Bahdanau et al., 2015). We also incorporate the intra decoder attention (Paulus et al., 2018) mechanism since it has been shown to improve the performance of the pointer-generator. GPT2 (Radford et al., 2019) is a large-scale pre-trained language model based on Transformer (Vaswani et al., 2017). DistilGPT2 is a compressed version of GPT2 model using the knowledge distillation technique (Sanh et al., 2019). We append the text ''TL;DR'' to the input document to trigger the summarization operation by DistilGPT2. We append control tokens to these two models.

3.4 Length-controlled Summarization

Length-controlled summarization aims to control the length of generated summaries. We adopt

the setting proposed by Fan et al. (2018), which allows users to constrain the summary length to a pre-defined range (e.g., 33 to 37 words). We first divide summary length into 10 disjoint length bins $\mathcal{LB} = (lb_1, \ldots, lb_{10})$. Each length bin corresponds to a range of length, and each bin contains a roughly equal number of training samples in the corpus. Let lb_{i^*} denote the specified length bin. The goal of this task is to generate a summary $\mathbf{y}$ that satisfies the specified length bin lb_{i^*}.

Base Model. We expand the vocabulary of the model with ten special tokens (e.g., <bin_2>) to denote the corresponding bins. In training, we feed the token that indicates the length bin of the reference summary. During testing, we control the length of the output summary by inputting the token of our specified length bin. For pointer-generator, we prepend the token at the beginning of the document. For DistilGPT2, we insert the special token into the ''TL;DR:'' prefix (e.g., ''TL;DR<bin_2>:'').

Length Bin Constraint. To encourage the summary length to match the specified length bin, we define a cost function that computes the normalized distance between the length bin of the generated summary $\hat{i}$ and the specified length bin i^*: $|\hat{i} - i^*|/10$, then we set the threshold $\alpha = 0$, which leads to the following length bin constraint: $|\hat{i} - i^*| \leq 0$. We adopt a normalized cost function to prevent the values of costs from being too large and dominating the gradient $\nabla_\theta L$ in Eq. (3).

3.5 Entity-controlled Summarization

Our second task is to generate a summary that focuses on entities requested by a user. Fan et al. (2018) anonymize each entity in the document by a special token. In contrast, we do not anonymize the entities, which is a more realistic setup.

Base Model. During training, we prepend the reference entities to the document. These requested entities are separated by segmenters, for example, ''Lebron James <ent> LA Lakers''. In test time, we control the focus of the summary by feeding in our specified entities. To make the reference summaries focus on the reference entities, we remove the reference summary sentences that contain neither reference entities nor coreferent

1217

mentions of reference entities on training, validation, and test splits.[3]

QA Constraint. We apply a question-answering (QA) constraint to guide the generated summary to capture the important information of the requested entities. The main idea is to use the QA-based metric from Eyal et al. (2019) and Scialom et al. (2019) to evaluate the capability of a summary to answer a set of questions regarding the reference entities. The QA constraint ensures that the score of the QA-based metric is above a threshold.

Specifically, we first construct a set of cloze question-answer pairs by individually masking each of the named entities from the reference summary to create the question, with the masked entity as its gold-standard answer. The summary predicted by a system is considered as the *context* for a QA model. We feed each of the cloze questions and the context to the QA model, then the QA model extracts an answer from the context for each cloze question. We use the F_1 score of the answers extracted by the QA model as the evaluation metric, denoted as **QA-F_1 score**. If a summary presents the key information of the reference entities, then the QA-model can predict the correct answers from the summary most of the time. We use the negative of QA-F_1 as our cost function and set the threshold to -0.9. Our QA constraint is then defined as: $-\text{QA-}F_1(\mathbf{y}) \leq -0.9$.

The QA model is a BERT model (Devlin et al., 2019) with a span classification head on top of the last-layer hidden states. The span classification head is a fully connected layer that predicts the beginning and ending positions of the answer span on the context. We obtain a BERT-based QA model that is fine-tuned on SQuAD 2.0 (Rajpurkar et al., 2018) from Huggingface Transformers (Wolf et al., 2019). Then we further fine-tune the QA model on the CNN/Dailymail (Hermann et al., 2015; Nallapati et al., 2016) corpus using our constructed question-context-answer triplets. We construct 349,653/17,442 cloze question-context-answer triplets for training and development. The details of the construction method are described in §A.2.

Entity Repetition Constraint. We find that the QA constraint will cause the model to repeatedly

generate the same requested entity in a sentence, because the model wants to increase the chance that the QA model will select the requested entities as the answer. Since a named entity usually contains one or two words, the entity repetition behavior cannot be fixed by the 3-gram repetition constraint. To address this problem, we first define a function $\text{ER}(\mathbf{y})$ to measure the fraction of sentences in $\mathbf{y}$ that contain repetition of requested entities. We then use $\text{ER}(\mathbf{y})$ as the cost function and apply the following constraint: $\text{ER}(\mathbf{y}) \leq 0$.

3.6 Abstractiveness-controlled Summarization

Our third task is abstractiveness-controlled summarization, which allows a user to specify the degree of text novelty between a generated summary and the corresponding document.[4] In this work, we adopt **extractive fragment density** (Grusky et al., 2018) to measure the abstractiveness of a summary. Given a document $\mathbf{x}$ and a summary $\mathbf{y}$, the set of extractive fragments $\mathcal{F}(\mathbf{x}, \mathbf{y})$ is the set of common sequences of words in $\mathbf{x}$ and $\mathbf{y}$. Extractive fragment density is defined as the mean square of the extractive fragment lengths: $\frac{1}{l_{\mathbf{y}}} \sum_{f \in \mathcal{F}(\mathbf{x}, \mathbf{y})} |f|^2$. Intuitively, a summary that copies many longer text fragments from the document has a higher extractive fragment density and a lower abstractiveness. We divide the values of extractive fragment density into three abstractiveness bins: $ab_1 = (3.3, +\infty]$, $ab_2 = (1.3, 3.3]$, $ab_3 = [0, 1.3]$, which indicates low, medium, and high abstractiveness respectively. The goal of abstractiveness control is to generate a summary $\mathbf{y}$ that follows the specified abstractiveness bin ab_{i^*}.

Base Model. Similar to length control, we use special tokens to denote the abstractiveness bins and input a special token to control the abstractiveness level of the output summary.

Abstractiveness Bin Constraint. To avoid the output summary from violating the specified abstractivenss bin, we apply a cost function to evaluate the normalized distance between the abstractiveness bin of the output summary $\hat{i}$ and the desired abstractiveness bin i^*: $|\hat{i} - i^*|/3$. We set the threshold to 0 and obtain the following abstractiveness bin constraint: $|\hat{i} - i^*| \leq 0$.

---

[3]Fewer than 2% of the removed sentences contain named entities that have coreferent mentions.

[4]Abstraction refers to the process of semantic generalization of concepts in the source document. The degree of text novelty is a proxy for measuring abstractiveness.

Bin	CNN/DM	Newsroom	Newsroom-b
1	37.88%	45.92%	33.94%
2	57.56%	25.96%	37.54%
3	4.56%	28.12%	28.52%

Table 1: Distribution of abstractiveness bins of reference summaries on CNN/DM, Newsroom, and Newsroom-b training sets. Bin 3 is the most abstractive bin. Newsroom-b is a subset of Newsroom which has a more *balanced* distribution of abstractiveness bins.

Conjunction Constraint. We find that after applying the abstractiveness constraint, the model often inserts the conjunction ''but'' into a copied fragment to decrease the extractive fragment density, even if there is no contrast relationship. Since it is difficult to detect the improper use of conjunction, we devise a constraint to avoid the model from generating ''but'' when the reference summary does not contain ''but''. Concretely, we first define a binary function $IC(\mathbf{y})$ as follows. $IC(\mathbf{y}) = 1$ if the predicted summary $\mathbf{y}$ contains ''but'' and the reference summary does not contain ''but''; otherwise, $IC(\mathbf{y}) = 0$. We then apply the following conjunction constraint: $IC(\mathbf{y}) \leq 0$. This method can be generalized to other discourse markers depending on specific model behavior.

4 Experimental Setup

Datasets. We use three popular summarization datasets in our experiments. The first one is the **CNN/DailyMail** (Hermann et al., 2015; Nallapati et al., 2016) corpus. We use the standard splits, which have 287,113/13,368/11,490 samples for training, validation, and test sets. Each summary in the training set has 66 words on average. We follow the preprocessing steps of See et al. (2017). Table 1 shows the distribution of abstractiveness bins. We can observe that most of the reference summaries belong to abstractiveness bin 1 and 2, indicating that this dataset is not abstractive.

Moreover, we use a subset of the **Newsroom** (Grusky et al., 2018) corpus. Newsroom contains 1.3 million news articles with summaries from 38 different news publishers. We construct a subset of the Newsroom corpus called **Newsroom-b** that has a more balanced distribution of abstractiveness bins. We extract all the samples from three of the news publishers (Washington Post, The Guardian, and New York Times) and obtain the splits of

297,327/31,815/32,047 for training, validation, and test sets. The distribution of abstractiveness bins is shown in Table 1.

Furthermore, we conduct experiments of length control on the **DUC-2002** dataset (Ellis, 2002) using a test-only setup (Chen et al., 2018; Chen and Bansal, 2018; Chan and King, 2021). DUC-2002 consists of 567 documents and each document has two reference summaries. We remove the documents that are shorter than their corresponding reference summaries, resulting in 554 documents. This dataset has long reference summaries with an average length of 113 words.

Baselines and Comparison. We use maximum likelihood (ML) loss to train the pointer-generator and DistilGPT2 based controllable summarization models described in §3.5, denoted as **PG** and **D.GPT2** respectively. We then use a suffix ''+CMDP'' to indicate that a model is fine-tuned by our CMDP framework. The following baselines do not use pre-trained models. We consider the **ControlSum** (Fan et al., 2018) model as a baseline for all of our control settings. For entity control, we incorporate query-focused summarization baselines including **GRSUM** (Wan, 2008), an extractive model that incorporates query-relevance into a random walk algorithm, **QueryAtt** (Nema et al., 2017), an abstractive model that applies a query attention to focus on different parts of the input query, and **SD2** (Nema et al., 2017), which integrates an orthogonality constraint into the QueryAtt model to encourage the successive query attention context vectors to be orthogonal to each other. Both the QueryAtt and SD2 models have a strong inductive bias that the generated summary should focus on the query. We modify the ROUGESal (Pasunuru and Bansal, 2018) method by doubling the weights to the words of the requested entities and treat it as a baseline, denoted as **ROUGEEnt**.

Evaluation Metrics. For length control and entity control, we evaluate the quality of summaries using **ROUGE-1**, **ROUGE-2**, and **ROUGE-L** F_1 scores with full-length and stemming (Lin, 2004). For abstractiveness control, we use embedding-based metrics, **BERTScore** (Zhang et al., 2020b) and **MoverScore** (Zhao et al., 2019), to measure the semantic similarity between an output summary and a reference summary. To evaluate how well the generated summaries satisfy the attribute

1219

Method	Bin 1			Bin 4			Bin 7			Bin 10		
	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L
ControlSum	32.40	14.30	28.28	36.30	15.34	31.95	38.55	16.18	34.50	40.30	17.08	36.59
PG	27.93	12.06	24.40	31.41	12.51	27.23	31.81	12.27	27.54	31.94	11.79	28.09
PG+CMDP	**35.30**	**17.00**	**31.98**	37.88	**17.59**	34.27	**39.85**	**18.46**	**36.17**	40.73	17.11	37.30
D.GPT2	31.21	13.36	27.12	36.27	15.97	31.91	38.18	16.43	33.64	40.87	17.45	36.62
D.GPT2+CMDP	33.09	13.48	29.74	**38.41**	16.55	**34.59**	39.65	16.77	35.79	**42.05**	**17.77**	**38.35**

Table 2: Results of length control on different specified length bins using the DUC-2002 data. Our CMDP framework consistently improves the ROUGE scores of PG and D.GPT2 ($p < 0.04$, approximate randomization test, for ROUGE-1 and ROUGE-L).

requirement, we define a metric called **bin %** to measure the percentage of generated summaries that follow the specified bin (length or abstractiveness bin). We use the **QA-F_1** score defined in §3.5 to evaluate whether a summary retains the essential information of the reference entities. We define **reference entities** as all the named entities (typed as location, person, and organization) that appear in both the reference summary and the first 400 words of the input document. We also define **appear %** to measure the percentage of requested entities that appear in the summary. For the non-reference control settings, the entire test set is evaluated under different control constraints and reference summaries do not exist in these cases.

Implementation Details. We use Spacy (Honnibal et al., 2020) for coreference resolution. For RNN-based models, we use the Adam algorithm (Kingma and Ba, 2015) for training. We first use ML loss to train a RNN-based model until the validation loss stops decreasing for three consecutive checkpoints. Then we start the (C)MDP training. The initial learning rates are 1e-3 and 5e-5 for ML and CMDP training, respectively. For Transformer-based models, we use the AdamW algorithm (Loshchilov and Hutter, 2017) for training. We first use ML loss to train a Transformer-based model for 12 epochs. Then we start the (C)MDP training. The initial learning rates are 5e-5 and 1.77e-5 for ML and CMDP training. During CMDP training of D.GPT2, we freeze the bottom four layers of the model. We initialize the values of λ to 0.01.

5 Automatic Evaluation Results

5.1 Results of Length Control

Reference Length Bin. We first evaluate the performance of length controlled models when

Method	R-1	R-2	R-L	Bin %
ControlSum	39.75	17.43	36.70	48.15
PG	35.07	15.05	32.11	74.09
PG+CMDP	39.77	16.65	36.66	**94.37**
D.GPT2	39.28	17.36	36.07	50.74
D.GPT2+CMDP	**41.72**	**17.99**	**39.00**	70.13
D.GPT2+MDP	41.46	17.69	38.74	69.71

Table 3: Results of length control using reference length bins as the input on the CNN/DM dataset. Our CMDP framework significantly improves the ROUGE scores and bin % of both PG and D.GPT2 ($p < 0.0001$, approx. randomization test).

supplying the length bin of the reference summary (reference length bin) at testing time. The results are shown in Table 3. We observe that *after applying our CMDP framework, both PG and D.GPT2 models obtain significantly higher ROUGE scores and a larger portion of their generated summaries follow the specified length bin.* We also report the results of the D.GPT2 model after fine-tuned by RL with MDP (D.GPT2+MDP). In this MDP approach, the reward is BERTScore minus a weighted sum of length bin distance and 3-gram repetition ratio. We tune the weights of penalties on the validation set and set the weights for length bin distance and 3-gram repetition to 0.4 and 0.6, respectively. We can see that our CMDP approach outperforms the MDP approach. The above results demonstrate the effectiveness of our framework.

Moreover, we observe that the D.GPT2 based models obtain higher ROUGE scores but lower bin % than the PG based models. One possible reason is that the large-scale pre-training in D.GPT2 makes the model more difficult to adapt to a specific bin requirement. This suggests a trade-off between the task metrics and the bin %.
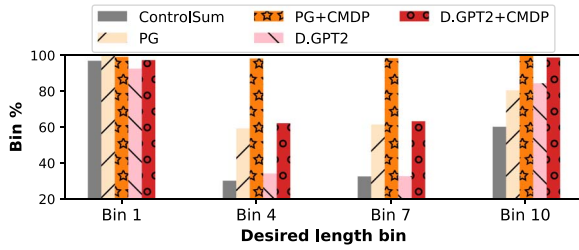
Figure 2: Bin % of different models with different specified length bins on the DUC-2002 dataset. Our framework improves the bin % of PG and D.GPT2 for bin 4, 7, and 10 by a wide margin.

Arbitrary Length Bin. We evaluate the performance of length-controlled models when supplying different length bins at testing time. We report the results of length-controlled models on four different length bins: 1, 4, 7, and 10. The DUC-2002 dataset is adopted since this dataset has long reference summaries. Hence, we can evaluate the quality of summaries with different lengths by truncating the summaries. We truncate the reference and system summaries to 33, 46, 59, and 100 for specified length bins of 1, 4, 7, and 10, respectively, when computing ROUGE scores. ROUGE evaluation with truncation is a common practice for evaluating a system summary when given a length budget (Hong et al., 2014). The intuition is that a good summary should contain the more essential information at the beginning.

We analyze the results of length-controlled models on different length bins. Figure 2 illustrates the results of bin % obtained by different models. We observe that *all the models achieve more than 90 bin % for length bin 1*. It is because length bin 1 represents the range of $(0, 33]$ in length, it is easy to satisfy the requirement by generating a very short summary. *For length bin 4, 7, and 10, our CMDP framework improves the bin % of both PG and D.GPT2 models by a wide margin*. From Table 2, we can see that *our framework consistently improves the ROUGE scores of PG and D.GPT2 models*.

Costs and Lagrangian Multipliers. Furthermore, we analyze the values of costs ($\mathbf{c}$) and Lagrangian multipliers ($\boldsymbol{\lambda}$) of our PG+CMDP model during training. From Figure 3, we can see that *the costs received by the agent decrease gradually over iterations*. It is because the relaxed training objective of our framework in Eq. (2) penalizes the behavior of violating the constraints.
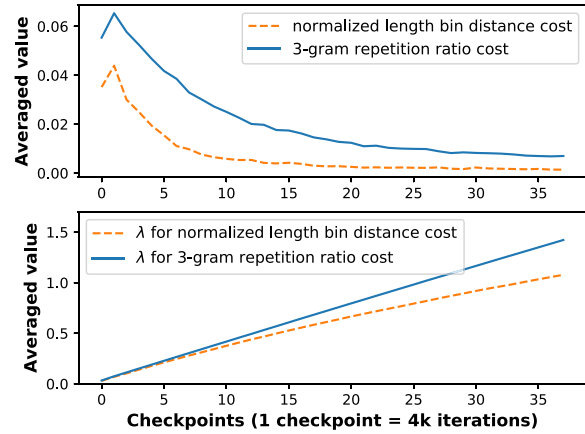

Figure 3: Values of costs ($\mathbf{c}$) and Lagrangian multipliers ($\boldsymbol{\lambda}$) of PG+CMDP for length control on every checkpoint (4k iterations) during training. Each value is averaged over 4k iterations.

We also observe that *the values of Lagrangian multipliers $\boldsymbol{\lambda}$ keeps increasing*. The reason is that according to Eq. (4), the gradient of $\boldsymbol{\lambda}$ is negative as long as there is a sample that violates the constraints during training. As mentioned in § 3.2, $\boldsymbol{\lambda}$ is learned by a gradient descent algorithm and the algorithm increases $\boldsymbol{\lambda}$ when the gradient is negative.

5.2 Results of Entity Control

Reference Entities. We first evaluate the performance of entity-controlled models in summarizing the reference entities. For each of the models, we feed in all the reference entities to generate a summary that centers on the reference entities. The results are presented in Table 4. We use the CNN/DM dataset for entity-controlled summarization because it contains named entities in 99.74% of the reference summaries in its test set, whereas the Newsroom-b dataset only has 85.24%. When computing QA-F_1 and appear %, we ignore the samples that do not have a named entity in the reference summary. We observe that *our framework consistently and significantly improves the ROUGE scores, QA-F_1 score, and appear % for both of the PG and D.GPT2 models*. These results demonstrate the effectiveness of our framework in summarizing reference entities.

We also adopt the D.GPT2+MDP model as a rival system. In this control setting, the reward is BERTScore($\mathbf{y}$) + γ_1QA$F_1(\mathbf{y})$ − γ_2RepeatRatio$_3(\mathbf{y})$ − γ_3ER($\mathbf{y}$). We set $\gamma_1, \gamma_2, \gamma_3$ to 0.15, 0.4, and 0.5 respectively after hyper-parameter tuning. It is observed that the MDP

1221

Method	R-1	R-2	R-L	QA-F_1	Appear %
GRSUM	35.89	15.86	31.96	34.92	76.22
ROUGEEnt	39.45	20.36	36.78	23.47	83.75
ControlSum	39.41	19.94	36.55	27.02	74.08
QueryAtt	38.92	20.38	36.47	25.12	75.10
SD2	39.43	20.71	36.88	27.23	75.97
PG	37.61	19.27	35.04	23.53	37.96
PG+CMDP	40.81	20.23	37.56	30.38	86.64
D.GPT2	41.68	22.32	38.85	35.32	82.31
D.GPT2+CMDP	**45.00**	**23.65**	41.85	**36.00**	93.37
D.GPT2+MDP	**45.00**	23.50	**41.90**	35.72	**94.46**

Table 4: Results of entity-controlled models using reference entities as the input on the CNN/DM dataset. Our CMDP framework significantly improves the ROUGE scores, QA-F_1, and appear % ($p < 0.0001$, approx. randomization test).

approach and our CMDP approach obtain similar performance while our approach has fewer hyperparameters to tune.

Entities at Different Positions. Next, we evaluate the capability of entity-controlled models to summarize entities at different positions of the document with the following setup. For each of these models, we use the named entities at document sentences 1 to 2, 3 to 4, 5 to 6, and 7 to 8 as the requested entities respectively. Since we do not have reference summaries for these entities, we use the document sentences to construct cloze questions to evaluate the output summaries. For each requested entity, we build cloze questions by masking each document sentence that contains the entity or its coreferent mention. We use the F_1 score of the answer predicted by the QA model as an evaluation metric, denoted as QA'-F_1.

We analyze the performance of our method for entities at various sentences of the document. The results of appear % and QA'-F_1 scores are presented in Figure 4. We observe that *our CMDP framework consistently improves the appear % and QA'-F_1 scores of both PG and D.GPT2 models for entities at different positions*. Without our CMDP training, the appear % are low for entities at latter positions of the document. The reason is that we use reference entities for model training and the reference entities are concentrated in the first few sentences of the document, which bias a neural model towards these sentences. There are 45.6% of reference entities appear in the first two document sentences in the training set of CNN/DM. Nevertheless, *the neural models fine-tuned by our*
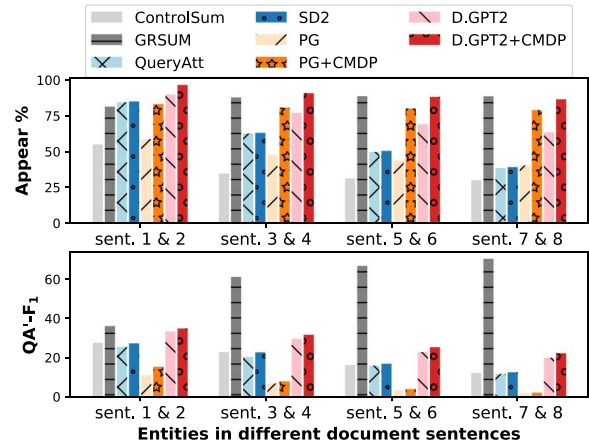

Figure 4: Results of entity-controlled models for entities in different document sentences. Our CMDP framework consistently improves the QA'-F_1 and appear % for entities at different positions.

CMDP achieve high appear % for entities at varying positions.

Moreover, we observe that the GRSUM system achieves highest QA'-F_1 scores and its appear % scores are similar to that of D.GPT2+CMDP. We analyze the reasons as follows. The GRSUM system is an extractive method while all other methods in Figure 4 are abstractive methods. It is relatively easy for an extractive method to select document sentences that mention the request entities to obtain high appear %. In the setting of non-reference entity control, we use document sentences to construct the cloze questions for the QA'-F_1 metric since we do not have a reference summary. Hence, the QA'-F_1 metric tends to give higher scores to extractive summaries. Moreover, we also observe that the GRSUM model achieves higher QA'-F_1 scores for the entities at latter sentences of the document. The entities at latter positions of a news article are usually less important entities that are only mentioned once and do not have coreferent mentions. The GRSUM system relies on term vectors to measure the relevance of a sentence. Thus, this system cannot recognize a coreferent mention that uses completely different words (e.g., pronoun). As a result, it is easier for GRSUM to extract a summary for entities at latter locations. However, an extractive method cannot paraphrase the information of the document to generate a concise entity-focused summary.

5.3 Results of Abstractiveness Control

We analyze the capability of abstractiveness-controlled models to generate summaries with

1222

Method	bin 1			bin 2			bin 3		
	BERTS.	MoverS.	Bin %	BERTS.	MoverS.	Bin %	BERTS.	MoverS.	Bin %
ControlSum	26.53	16.15	99.40	24.50	13.11	6.42	20.53	9.87	24.30
PG	26.49	15.99	99.36	22.59	11.15	8.49	17.67	7.45	26.03
PG+CMDP	29.62	18.44	**99.72**	26.95	12.96	**97.65**	22.78	7.67	**98.88**
D.GPT2	27.85	17.05	99.17	26.41	14.57	0.47	22.30	11.16	00.37
D.GPT2+CMDP	**30.12**	**18.45**	99.72	**31.21**	**17.52**	72.48	**25.75**	**13.18**	87.09
D.GPT2+MDP	29.77	18.27	**99.77**	30.40	17.06	72.37	25.25	13.06	80.95

Table 5: Results of abstractiveness-controlled models with different specified bins on Newsroom-b dataset. Bin 3 is the most abstractive bin. Our CMDP framework significantly improves the BERTScore, MoverScore, and bin % over all the bins ($p < 0.003$, approx. randomization test).

Method	bin 1			bin 2			bin 3		
	BERTS.	MoverS.	Bin %	BERTS.	MoverS.	Bin %	BERTS.	MoverS.	Bin %
ControlSum	38.55	23.56	99.94	39.47	23.23	1.09	37.51	20.58	0.03
PG	35.42	20.00	99.85	34.91	18.67	1.42	32.53	15.87	0.21
PG+CMDP	41.77	25.96	**100.00**	40.79	23.54	**75.10**	34.22	17.71	**48.62**
D.GPT2	39.02	23.82	99.90	39.58	23.30	1.93	38.15	21.23	0.01
D.GPT2+CMDP	**43.23**	**26.65**	99.56	**44.07**	**26.39**	62.60	**42.03**	**24.71**	1.94
D.GPT2+MDP	42.56	26.43	99.77	43.59	26.23	55.67	41.44	24.42	2.09

Table 6: Results of abstractiveness-controlled models with different specified bins on CNN/DM dataset. Our CMDP framework significantly improves the BERTScore and MoverScore ($p < 0.003$, approx. randomization test) over all the bins. It also significantly improves the bin % for bin 2 and 3 ($p < 0.00001$, approx. randomization test).

different abstractiveness levels. In our experiments, for each of the abstractiveness-controlled models, we feed in abstractiveness bin 1, bin 2, and bin 3 independently. The results on Newsroom-b and CNN/DM datasets are presented in Table 5 and 6. We can see that *our CMDP framework consistently improves the BERTScores and MoverScores of PG and D.GPT2 models*. We also observe that *all the models achieve more than 99 bin % for bin 1 (least abstractive)*, because it is easier for models to directly copy document sentences than to paraphrase document information. *For abstractiveness bin 2 and 3, our CMDP framework substantially improves the bin % of PG and D.GPT2 models*, which show that our framework improves the ability of summarization models to generate summaries of higher abstractiveness levels. Similar to the results of length control, there is a trade-off between the task metrics and the bin %.

We then compare the bin % results on the CNN/DM dataset with that on Newsroom-b. It is observed that for abstractiveness bin 3 (most abstractive), all the models achieve a low bin % on CNN/DM but a substantially higher bin % on

Newsroom-b. This is because in the CNN/DM, there are only 4.6% of the training samples belonging to bin 3. Hence, it is difficult for a model to learn to generate a highly abstractive summary. In contrast, the Newsroom-b dataset has a balanced distribution of abstractiveness bins so that a model can learn from more abstractive references.

Furthermore, we compare our framework with the D.GPT2+MDP model on both datasets. The reward is $\text{BERTScore}(\mathbf{y}) - \gamma_1|\hat{i} - i^*|/3 - \gamma_2\text{RepeatRatio}_3(\mathbf{y}) - \gamma_3\text{IC}(\mathbf{y})$, where $\hat{i}$ denotes the abstractiveness bin of the generated summary and i^* denotes the specified abstractiveness bin. On the CNN/DM dataset, we set $\gamma_1, \gamma_2, \gamma_3$ to 0.3, 0.5, and 0.3 respectively. On the Newsroom-b dataset, we set these weights to 0.4, 0.5, and 0.3 respectively. We observe that the MDP approach and our CMDP approach obtain similar performance while our approach has fewer hyperparameters to tune.

6 Human Evaluation

We conduct human evaluation to verify the quality of the generated summaries. We hire postgraduate

1223

Method	Fluency	Entity-rel.	Faithful.
SD2	4.83	3.63	70%
D.GPT2	4.65	3.33	68%
D.GPT2+CMDP	4.83	3.92	71%

Table 7: Human fluency, entity-relevance, and faithfulness scores of entity-controlled models with the reference entities as the input. Faithful. denotes the percentage of generated summaries that are faithful. The Krippendorf's α inter-rater agreement for all columns are 0.68, 0.77, and 0.56.

Sent.	Method	Fluen.	Ent.-rel.	Faith.
3&4	SD2	4.75	2.81	63%
	D.GPT2+CMDP	4.79	3.36	64%
5&6	SD2	4.78	2.68	62%
	D.GPT2+CMDP	4.78	3.29	62%

Table 8: Human fluency, entity-relevance, and faithfulness scores of entity-controlled models for entities at different document sentences. The Krippendorf's α inter-rater agreement for these scores are 0.60, 0.78, and 0.44.

students as annotators and each test sample is evaluated by three annotators. The names of models are blinded to the annotators.

6.1 Results of Entity Control

The human annotators evaluate entity-controlled summarization models using the following metrics: (i) **fluency**: estimating the readability and grammaticality of a summary using a rating from 1 to 5; (ii) **faithfulness**: a yes/no question indicating whether a summary is factually consistent with the document. The annotators are instructed to state ''yes'' only if the summary does not contain any factual inconsistencies; and (iii) **entity-relevance**: evaluating how well a summary retains the key information of the requested entities from 1 to 5.

Reference Entities. We ask human annotators to evaluate the quality of summaries when requesting reference entities. For each of the entity-controlled models, we feed in all the reference entities. The overall number of annotators is six. For each of the test samples, we present the input document, requested entities, reference summary, and three system summaries generated by SD2, D.GPT2, and D.GPT2+CMDP models. We present the evaluation scores on 100 random samples of the CNN/DM dataset in Table 7. For the faithfulness metric, we report the percentage of faithful summary computed by majority vote (i.e., at least two out of three annotators vote as faithful). Our D.GPT2+CMDP method significantly outperforms the D.GPT2 and SD2 models in terms of entity-relevance (power analysis with mixed effects model (Card et al., 2020), power > 0.99, approx. randomization test, $p < 0.0001$) while maintaining similar fluency and faithfulness with the SD2 model (approx. randomization test, $p > 0.97$).

Entities at Different Positions. We pick the best two models (SD2 and D.GPT2+CMDP) in the previous section to further conduct human evaluation for entities at different sentences of the document. The total number of annotators is four. As mentioned in §5.2, most of the reference entities are located in document sentences 1 to 2. To avoid too much overlapping with the reference entities setting, we do not choose the bin of sentences 1 to 2 and conduct evaluation on the subsequent two bins, sentences 3 to 4 and 5 to 6. For each model, we feed in the named entities at document sentences 3 to 4 and 5 to 6 as the requested entities respectively. Since we do not have gold-standard summaries for this setup, we cannot show the reference summaries to the annotators. The results on 100 random samples are shown in Table 8. Our D.GPT2+CMDP model consistently achieves higher entity-relevance scores than the SD2 model (power analysis with mixed effects model, power > 0.81, approx. randomization test, $p < 0.0001$) and obtains competitive fluency and faithfulness scores (approx. randomization test, $p > 0.41$).

6.2 Results of Abstractiveness Control

The annotators evaluate abstractiveness-controlled models using the following setting. There are six annotators for the results of CNN/DM dataset and three annotators for the results of Newsroom-b dataset. For each test sample, we generate two groups of system summaries (group 1 and group 2). For group 1, we use our D.GPT2+CMDP model to generate three different summaries by feeding abstractiveness bin 1, bin 2, and bin 3 respectively. For group 2, we use our PG+CMDP model to generate three different summaries using a similar method. During evaluation, we present the source document, the reference summary, and two groups of system summaries to the

1224

Method	CNN/DM		Newsroom-b	
	EM	PM	EM	PM
PG+CMDP	66%	94%	84%	96%
D.GPT2+CMDP	66%	92%	86%	98%

Table 9: Results of exact match (EM) and partial match (PM) scores of human abstractiveness rankings that are consistent with the specified bins. The Krippendorf's α inter-rater agreement for the abstractiveness rankings on CNN/DM and Newsroom-b are 0.85 and 0.72 respectively.

annotators. The summaries within each group are randomly shuffled.

Abstractiveness Among Summaries. We evaluate the abstractiveness of the generated summaries by human judgments using the following setup. For each group of system summaries, we ask the annotators to give a ranking among the three system summaries according to their abstractiveness. For instance, if an annotator thinks that summary 1 > summary 2 > summary 3 in terms of abstractiveness, then the annotator gives a ranking of $[3, 2, 1]$ to them. The abstractiveness rankings from different annotators are then aggregated by averaging. If the aggregated abstractiveness ranking is consistent with the order of our specified abstractiveness bins, then this group of summaries has an **exact match**. For example, suppose the order of our specified abstractiveness bins is $[3, 2, 1]$. If the aggregated abstractiveness ranking is $[3, 1.6, 1.3]$, then then this group of summaries has an exact match. If the aggregated abstractiveness ranking is $[3, 1.3, 1.6]$, then there is no exact match. Moreover, we investigate whether the summaries of abstractiveness bin 1 and bin 3 can be distinguished by annotators. If the aggregated abstractiveness ranking is consistent with the order of abstractiveness bin 1 and bin 3, then there is a **partial match**. Suppose the order of our specified abstractiveness bins is $[3, 2, 1]$, if the aggregated ranking is $[3, 1.3, 1.6]$, then there is a partial match. If the aggregated ranking is $[1.6, 1.3, 3]$, then there is no partial match.

We analyze the exact match and partial match scores of abstractiveness-controlled models as follows. The results on 100 random test samples of the CNN/DM and Newsroom-b datasets[5] are pre-

Bin	Method	Flu.	Rel.	Faithful.
1	PG+CMDP	4.79	3.43	98%
	D.GPT2+CMDP	4.75	3.34	96%
2	PG+CMDP	4.52	2.34	58%
	D.GPT2+CMDP	4.57	3.14	66%
3	PG+CMDP	4.47	2.00	52%
	D.GPT2+CMDP	4.60	2.99	66%

Table 10: Human fluency, relevance, and faithfulness scores of abstractiveness-controlled models on Newsroom-b. The Krippendorf's α inter-rater agreement for these metrics are 0.51, 0.37, and 0.40.

sented in Table 9. We observe that our models on both of the two datasets achieve very high partial match scores, but our models on the CNN/DM dataset obtain lower exact match scores than that on the Newsroom-b dataset (approx. randomization test, $p < 0.02$). This is because the CNN/DM dataset is extractive in nature. Hence, it is more difficult to learn three levels of abstractiveness on CNN/DM. Nonetheless, our models can still achieve more than 60% exact match scores.

Quality of Individual Summaries. Next, we ask the annotators to evaluate the qualities of the summaries of three different abstractiveness bins using the following metrics: (i) **fluency**: measuring the readability of a summary from 1 to 5; (ii) **faithfulness**: a yes/no question asking whether a summary is factually consistent with the document; and (iii) **relevance**: evaluating how well a summary retains the salient information of the document on 1-5. The results of 100 random test samples from the Newsroom-b dataset[6] are presented in Table 10. When using abstractiveness bin 1 (lowest level), all the models achieve significantly higher fluency, relevance, and faithfulness (approx. randomization test, $p < 0.005$). The scores of all these metrics drop substantially for abstractiveness bin 2 and bin 3 because paraphrasing is more challenging than copying. Figure 5 illustrates sample summaries generated by our D.GPT2+CMDP model on the Newsroom-b dataset. We observe that the generated summary of bin 3 has a factual error, which is italicized in the figure.

---

[5] We use both CNN/DM and Newsroom-b because we want to understand the impact of the training dataset on the abstractiveness of the output summaries.

[6] We choose Newsroom-b because there are more generated summaries that satisfy the abstractiveness bin requirement, which is more suitable for comparing the quality of summaries of different abstractiveness bins.

Reference Summary: Manchester United have announced a 10-year contract with German manufacturers Adidas to be the club's new kit sponsor for a record-breaking minimum £750m.
Abstractiveness bin 1 (least abstractive): Manchester United have announced a 10-year contract with the German manufacturers Adidas to be the club's new kit sponsor for a record-breaking minimum £750m.
Abstractiveness bin 2 (medium abstractive): Manchester United have announced a 10-year contract with Adidas to sponsor new kit sponsor for £750m club record signing.
Abstractiveness bin 3 (most abstractive): Manchester United are hoping to secure £750m sponsorship deal with German company after contract is signed in 2015-16 season.

Figure 5: Sample summaries generated by our D.GPT2+CMDP model with abstractiveness bin 1, 2, and 3 on the Newsroom-b testing set. Extractive fragments in summaries are in blue color. Factual errors are in red color.

7 Conclusion

We propose a novel CMDP training framework for controllable text summarization. Our framework imposes constraints on the training objective to explicitly disallow the output summaries from violating the requirement specified by users. Moreover, we apply our framework to control key summarization attributes such as length, covered entities, and abstractiveness of the summaries. We then devise specific constraints to restrict each of these attributes respectively. Empirical studies on popular benchmarks demonstrate that our framework significantly improves the capability of controllable summarization models to conform to the desired attribute requirement.

In our framework, we can set hard constraints without tuning threshold values. For instance, we set the threshold of our length bin constraint to 0 to disallow the violation of length bin requirement. Compared to the weights of penalty in the MDP framework, the threshold value in a soft constraint is also easier to set. For example, the goal of entity control is to generate a summary that presents the key information of the requested entities, which implies that the generated summaries should obtain a high QA-F1 score. The range of QA-F1 score is [0, 1]. In order to encourage the generated summaries to obtain a high QA-F1 score, the threshold for QA-F1 score should be close to 1, this gives us a clue about how to set the value of threshold. On the other hand, the MDP framework does not give us any clues to set the values of penalty weights. In summary, our CMDP

framework needs to tune one threshold value for entity control and it does not need to tune any threshold for other control settings. Whereas the numbers of penalty weights to be tuned in the MDP framework are 2, 3, and 3 for length, entity, and abstractiveness control respectively.

References

Eitan Altman. 1999. *Constrained Markov Decision Processes*, volume 7. CRC Press.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings*.

Richard Bellman. 1957. A Markovian decision process. *Journal of Mathematics and Mechanics*, pages 679–684.

Dimitri P. Bertsekas. 1997. Nonlinear programming. *Journal of the Operational Research Society*, 48(3):334–334. https://doi.org/10.1057/palgrave.jors.2600425

Dallas Card, Peter Henderson, Urvashi Khandelwal, Robin Jia, Kyle Mahowald, and Dan Jurafsky. 2020. With little power comes great responsibility. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16–20, 2020,*

pages 9263–9274. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.emnlp-main.745

Asli Çelikyilmaz, Antoine Bosselut, Xiaodong He, and Yejin Choi. 2018. Deep communicating agents for abstractive summarization. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1–6, 2018, Volume 1 (Long Papers)*, pages 1662–1675. Association for Computational Linguistics. https://doi.org/10.18653/v1/N18-1150

Hou Pong Chan, Wang Chen, and Irwin King. 2020. A unified dual-view model for review summarization and sentiment classification with inconsistency loss. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25–30, 2020*, pages 1191–1200. https://doi.org/10.1145/3397271.3401039

Hou Pong Chan and Irwin King. 2021. A condense-then-select strategy for text summarization. *Knowledge-Based Systems*, page 107235. https://doi.org/10.1016/j.knosys.2021.107235

Xiuying Chen, Shen Gao, Chongyang Tao, Yan Song, Dongyan Zhao, and Rui Yan. 2018. Iterative document representation learning towards summarization with polishing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 – November 4, 2018*, pages 4088–4097. https://doi.org/10.18653/v1/D18-1442

Yen-Chun Chen and Mohit Bansal. 2018. Fast abstractive summarization with reinforce-selected sentence rewriting. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15–20, 2018, Volume 1: Long Papers*, pages 675–686. Association for Computational Linguistics. https://doi.org/10.18653/v1/P18-1063

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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2–7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Angela Ellis. 2002. Document Understanding Conferences (DUC) 2002 Data.

Matan Eyal, Tal Baumel, and Michael Elhadad. 2019. Question answering as an automatic evaluation metric for news article summarization. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2–7, 2019, Volume 1 (Long and Short Papers)*, pages 3938–3948. Association for Computational Linguistics.

Angela Fan, David Grangier, and Michael Auli. 2018. Controllable abstractive summarization. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation, NMT@ACL 2018, Melbourne, Australia, July 20, 2018*, pages 45–54.

Jessica Ficler and Yoav Goldberg. 2017. Controlling linguistic style aspects in neural language generation. *CoRR*, abs/1707.02633.

Sebastian Gehrmann, Yuntian Deng, and Alexander M. Rush. 2018. Bottom-up abstractive summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 4098–4109. Association for Computational Linguistics.

Max Grusky, Mor Naaman, and Yoav Artzi. 2018. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1–6, 2018, Volume 1 (Long Papers)*, pages 708–719. https://doi.org/10.18653/v1/N18-1065

Jiatao Gu, Qi Liu, and Kyunghyun Cho. 2019. Insertion-based decoding with automatically inferred generation order. *Transactions of the Association for Computational Linguistics*, 7:661–676.

Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7–12, 2015, Montreal, Quebec, Canada*, pages 1693–1701.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780. `https://doi.org/10.1162/neco.1997.9.8.1735`, PubMed: 9377276

Kai Hong, John M. Conroy, Benoît Favre, Alex Kulesza, Hui Lin, and Ani Nenkova. 2014. A repository of state of the art and competitive baseline summaries for generic news summarization. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26–31, 2014*, pages 1608–1616.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

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, ACL 2020, Online, July 5–10, 2020*, pages 5094–5107. `https://doi.org/10.18653/v1/2020.acl-main.457`

Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. 2019. CTRL: A conditional transformer language model for controllable generation. *CoRR*, abs/1909.05858.

Yuta Kikuchi, Graham Neubig, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. 2016. Controlling output length in neural encoder-decoders. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1–4, 2016*, pages 1328–1338. `https://doi.org/10.18653/v1/D16-1140`

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings*.

Catherine Kobus, Josep Crego, and Jean Senellart. 2017. Domain control for neural machine translation. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP* 2017, pages 372–378, Varna, Bulgaria. INCOMA Ltd.

Wojciech Kryscinski, Romain Paulus, Caiming Xiong, and Richard Socher. 2018. Improving abstraction in text summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 1808–1817. `https://doi.org/10.18653/v1/D18-1207`

Philippe Laban, Andrew Hsi, John Canny, and Marti A. Hearst. 2020. The summary loop: Learning to write abstractive summaries without examples. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5–10, 2020*, pages 5135–5150. `https://doi.org/10.18653/v1/2020.acl-main.460`

Logan Lebanoff, Kaiqiang Song, Franck Dernoncourt, Doo Soon Kim, Seokhwan Kim, Walter Chang, and Fei Liu. 2019. Scoring sentence singletons and pairs for abstractive summarization. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28–August 2, 2019, Volume 1: Long Papers*, pages 2175–2189. Association for Computational Linguistics. `https://doi.org/10.18653/v1/P19-1209`

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. *Text Summarization Branches Out*.

1228

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3–7, 2019*, pages 3728–3738. `https://doi.org/10.18653/v1/D19-1387`

Yizhu Liu, Zhiyi Luo, and Kenny Q. Zhu. 2018. Controlling length in abstractive summarization using a convolutional neural network. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 4110–4119.

Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in Adam. *CoRR*, abs/1711.05101.

Takuya Makino, Tomoya Iwakura, Hiroya Takamura, and Manabu Okumura. 2019. Global optimization under length constraint for neural text summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1039–1048, Florence, Italy. Association for Computational Linguistics. `https://doi.org/10.18653/v1/P19-1099`

Ning Miao, Hao Zhou, Lili Mou, Rui Yan, and Lei Li. 2019. CGMH: constrained sentence generation by metropolis-hastings sampling. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6834–6842. `https://doi.org/10.1609/aaai.v33i01.33016834`

Ramesh Nallapati, Bowen Zhou, Cícero Nogueira dos Santos, Çaglar 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, CoNLL 2016, Berlin, Germany, August 11–12, 2016*, pages 280–290. `https://doi.org/10.18653/v1/K16-1028`

Preksha Nema, Mitesh M. Khapra, Anirban Laha, and Balaraman Ravindran. 2017. Diversity driven attention model for query-based abstractive summarization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 – August 4, Volume 1: Long Papers*, pages 1063–1072. `https://doi.org/10.18653/v1/P17-1098`

Tong Niu and Mohit Bansal. 2018. Polite dialogue generation without parallel data. *TACL*, 6:373–389. `https://doi.org/10.1162/tacl_a_00027`

Ramakanth Pasunuru and Mohit Bansal. 2018. Multi-reward reinforced summarization with saliency and entailment. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1–6, 2018, Volume 2 (Short Papers)*, pages 646–653. `https://doi.org/10.18653/v1/N18-2102`

Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A deep reinforced model for abstractive summarization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*.

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

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers*, pages 784–789. `https://doi.org/10.18653/v1/N18-2102`

Steven J. Rennie, Etienne Marcheret, Youssef Mroueh, Jarret Ross, and Vaibhava Goel. 2017. Self-critical sequence training for image captioning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21–26, 2017,*

pages 1179–1195. `https://doi.org/10.1109/CVPR.2017.131`

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Raphael Schumann, Lili Mou, Yao Lu, Olga Vechtomova, and Katja Markert. 2020. Discrete optimization for unsupervised sentence summarization with word-level extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5–10, 2020*, pages 5032–5042. `https://doi.org/10.18653/v1/2020.acl-main.452`

Thomas Scialom, Sylvain Lamprier, Benjamin Piwowarski, and Jacopo Staiano. 2019. Answers unite! unsupervised metrics for reinforced summarization models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3–7, 2019*, pages 3244–3254.

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, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1073–1083.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Controlling politeness in neural machine translation via side constraints. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12–17, 2016*, pages 35–40. `https://doi.org/10.18653/v1/N16-1005`

Eva Sharma, Luyang Huang, Zhe Hu, and Lu Wang. 2019. An entity-driven framework for abstractive summarization. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Nat-*

ural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3–7, 2019, pages 3278–3289. Association for Computational Linguistics. `https://doi.org/10.18653/v1/D19-1323`

Kaiqiang Song, Bingqing Wang, Zhe Feng, Ren Liu, and Fei Liu. 2020. Controlling the amount of verbatim copying in abstractive summarization. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8902–8909. AAAI Press. `https://doi.org/10.1609/aaai.v34i05.6420`

Kaiqiang Song, Lin Zhao, and Fei Liu. 2018. Structure-infused copy mechanisms for abstractive summarization. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20–26, 2018*, pages 1717–1729.

Qing Sun, Stefan Lee, and Dhruv Batra. 2017. Bidirectional beam search: Forward-backward inference in neural sequence models for fill-in-the-blank image captioning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21–26, 2017*, pages 7215–7223. `https://doi.org/10.1109/CVPR.2017.763`

Richard S. Sutton and Andrew G. Barto. 1998. *Reinforcement Learning - an Introduction.* Adaptive computation and machine learning. MIT Press.

Shunsuke Takeno, Masaaki Nagata, and Kazuhide Yamamoto. 2017. Controlling target features in neural machine translation via prefix constraints. In *Proceedings of the 4th Workshop on Asian Translation, WAT@IJCNLP 2017, Taipei, Taiwan, November 27–December 1, 2017*, pages 55–63.

Chen Tessler, Daniel J. Mankowitz, and Shie Mannor. 2019. Reward constrained policy optimization. In *International Conference on Learning Representations (ICLR)*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural*

Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4–9 December 2017, Long Beach, CA, USA, pages 5998–6008.

Xiaojun Wan. 2008. Using only cross-document relationships for both generic and topic-focused multi-document summarizations. *Information Retrieval*, 11(1):25–49. https://doi.org/10.1007/s10791-007-9037-5

Xiaojun Wan. 2009. Topic analysis for topic-focused multi-document summarization. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management, CIKM 2009, Hong Kong, China, November 2–6, 2009*, pages 1609–1612. ACM.

Xiaojun Wan and Jianguo Xiao. 2009. Graph-based multi-modality learning for topic-focused multi-document summarization. In *IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA, July 11–17, 2009*, pages 1586–1591.

Xiaojun Wan, Jianwu Yang, and Jianguo Xiao. 2007. Manifold-ranking based topic-focused multi-document summarization. In *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6–12, 2007*, pages 2903–2908.

Xiaojun Wan and Jianmin Zhang. 2014. CT-SUM: extracting more certain summaries for news articles. In *The 37th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '14, Gold Coast, QLD, Australia - July 06 - 11, 2014*, pages 787–796. ACM.

Lu Wang, Hema Raghavan, Vittorio Castelli, Radu Florian, and Claire Cardie. 2013. A sentence compression based framework to query-focused multi-document summarization. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4–9 August 2013, Sofia, Bulgaria, Volume 1: Long Papers*, pages 1384–1394.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019.

Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771. https://doi.org/10.18653/v1/2020.emnlp-demos.6

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020a. PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13–18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 11328–11339. PMLR.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020b. BERTScore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26–30, 2020*. OpenReview.net.

Yuhao Zhang, Derek Merck, Emily Bao Tsai, Christopher D. Manning, and Curtis Langlotz. 2020c. Optimizing the factual correctness of a summary: A study of summarizing radiology reports. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5–10, 2020*, pages 5108–5120. https://doi.org/10.18653/v1/2020.acl-main.458

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger. 2019. MoverScore: Text generation evaluating with contextualized embeddings and earth mover distance. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3–7, 2019*, pages 563–578. Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1053

Wanrong Zhu, Zhiting Hu, and Eric P. Xing. 2019. Text infilling. *CoRR*, abs/1901.00158.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2020. Fine-tuning language models from human preferences. *CoRR*, abs/1909.08593.

A Appendix

A.1 Output Samples for Length Control

Figure 6 presents sample summaries generated by our D.GPT2+CMDP model using different length bins on the DUC-2002 testing set. We observe that our model discards secondary information when given a shorter length budget.

A.2 Training Data for the QA Model

We construct question-context-answer triplets to train a QA model. We individually mask each named entity in a reference summary to create a cloze question and the masked entity is its answer. The reference summary is used as the context. For example, suppose the reference summary y^* is ``Arsenal beat Chelsea 3-1 yesterday.'', then we construct two cloze questions, $q_1 =$``[MASK] beat Chelsea 3-1 yesterday.'' and $q_2 =$``Arsenal beat [MASK] 3-1 yesterday.'', and two answers, $a_1 =$``Arsenal'' and $a_2 =$``Chelsea''. After that, we obtain two question-context-answer triplets, (q_1, y^*, a_1) and (q_2, y^*, a_2).

Since the constructed cloze questions are too similar to the corresponding reference summaries, if we only use reference summaries as the context in our training data, it will encourage the QA model to only rely on surface clues to extract answers. To alleviate this problem, we use the method by Chen and Bansal (2018) to extract a pseudo reference summary $\tilde{y}$ from the source document. Then we use $\tilde{y}$ as the context to construct another set of question-context-answer triplets $\{(q_i, \tilde{y}, a_i)\}$. The pseudo reference summary includes the document sentences that achieve highest ROUGE-L recall with the reference summary. We discard a triplet if $\tilde{y}$ does not contain all the named entities in the reference. To have a balanced training data, we only keep the training triplets (q_i, y^*, a_i) that has a corresponding pseudo reference summary $(q_i, \tilde{y}, a_i)$.

Length bin 0:
Hurricane Gilbert slams into Kingston on Monday. 115 mph winds cause flash floods and mud slides. Jamaica expected to receive 10 inches of rain. Hurricane warnings are canceled.
Length bin 3:
NEW: Hurricane Gilbert slams into Kingston on Monday with 115 mph winds. No serious injuries were reported in the city of 750,000. The hurricane lashed Kingston's airport and aircraft. Jamaica will receive 10 inches of rain. Hurricane warnings are canceled.
Length bin 6:
NEW: Hurricane Gilbert slams into Kingston on Monday with 115 mph winds. No serious injuries were reported in the city of 750,000 people. The hurricane hit Kingston's airport and aircraft parked on its fields. The National Weather Service reports heavy damage to Kingston's airport. Jamaica will receive 10 inches of rain.
Length bin 9:
NEW: "The eye is going to move lengthwise across that island," a man says. Hurricane Gilbert slams into Kingston on Monday with 115 mph winds. The National Weather Service reports heavy damage to Kingston's airport and aircraft parked on its fields. "People were running around in the main lobby of our hotel," a man says. Jamaica is expected to receive 10 inches of rain. Hurricane warnings are canceled in Cuba.

Figure 6: Sample summaries generated by our D.GPT2+CMDP model using different length bins on the DUC-2002 testing set.

To allow the QA model to give a prediction of ``unanswerable'' to low-quality summaries, we construct two types of unanswerable training samples: irrelevant training samples and repeated-entity training samples. For irrelevant training samples, we select document sentences that do not contain the reference entities and have a low textual overlap with the reference summary (ROUGE-L recall ≤ 0.2). For repeated-entity training samples, we find out the sentences in the reference summary that contains two named entities and repeat one of its named entities. We treat such samples as unanswerable since they contain factual inconsistencies. Overall, our training data consists of 109,815 unanswerable samples and 239,838 answerable samples. We will release our training data for the QA model.