A New Approach to Overgenerating and Scoring Abstractive Summaries

Kaiqiang Song,[1] **Bingqing Wang,**[2] **Zhe Feng,**[2] **Fei Liu**[1]

[1]University of Central Florida, Orlando, FL
[2]Robert Bosch LLC, Sunnyvale, CA

kqsong@knights.ucf.edu, {bingqing.wang,zhe.feng2}@us.bosch.com
feiliu@cs.ucf.edu

Abstract

We propose a new approach to generate multiple variants of the target summary with diverse content and varying lengths, then score and select *admissible* ones according to users' needs. Abstractive summarizers trained on single reference summaries may struggle to produce outputs that achieve multiple desirable properties, i.e., capturing the most important information, being faithful to the original, grammatical and fluent. In this paper, we propose a two-staged strategy to generate a diverse set of candidate summaries from the source text in stage one, then score and select admissible ones in stage two. Importantly, our generator gives a precise control over the length of the summary, which is especially well-suited when space is limited. Our selectors are designed to predict the optimal summary length and put special emphasis on faithfulness to the original text. Both stages can be effectively trained, optimized and evaluated. Our experiments on benchmark summarization datasets suggest that this paradigm can achieve state-of-the-art performance.

1 Introduction

The learning objective of a modern abstractive summarizer is to produce system outputs that resemble reference summaries on a word-to-word basis. It does *not* promote outputs that possess multiple desirable properties, i.e., capturing the most important information, being faithful to the original text, grammatical and fluent, though some of these properties are exhibited by system abstracts as a natural outcome of a learned summarizer (See et al., 2017; Takase et al., 2016; Tan et al., 2017; Chen and Bansal, 2018; Celikyilmaz et al., 2018; Gehrmann et al., 2018; Liu and Lapata, 2019; Lebanoff et al., 2019b; Fabbri et al., 2019; Bražinskas et al., 2020). Without direct optimization of desired properties, system abstracts often change the meaning of the original document or fail to convey the main concepts (Kryscinski et al., 2020).

Source Text
• Police arrested five anti-nuclear protesters Thursday after they sought to disrupt loading of a French Antarctic research and supply vessel, a spokesman for the protesters said.
Summary
✔ Police arrest anti-nuclear protesters
✔ Protesters target French research ship
✘ French police arrest five anti-nuclear protesters
✘ Police arrest five anti-nuclear protesters in Antarctica
✘ Police arrest five anti-nuclear protesters at French Antarctic

Table 1: Example of alternative summaries generated from the source text. *Admissible* summaries are marked by ✔. System summaries that fail to preserve the meaning of the source input are marked by ✘.

In this paper, we propose a new approach to overgenerate and select *admissible* summaries, which allows a summarizer to juggle multiple objectives and strike a good balance between them (Belz and Reiter, 2006). Our approach consists of two stages. Given a source text, a *generator* explores the space of all possible lengths to produce multiple variants of the target summary that contain diverse content. We then devise *selectors* to validate the quality of alternative summaries to predict whether they are admissible. Our selection mechanism can be customized to suit particular needs without changing the generation space. Both stages can be effectively trained, optimized and evaluated.

Crucially, we take a confidence-driven approach to summary generation rather than using a left-to-right order. Beginning writers and language learners do not write in a strict sequential manner. In a similar vein, our generator produces a summary by "filling-in-the-blanks" with appropriate words. The most confident words are generated first, less vital ones later. With confidence-driven generation, our summarizer learns to dynamically add or remove content, and even paraphrase to produce a summary of a given length. In Table 2, we show an example illustrating the difference between our method and left-to-right generation. Our method dramatically enhances the capability of the generator, making it possible to explore summaries of varying lengths.

Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 1392–1404
June 6–11, 2021.

Source Text: A court here Thursday sentenced a 24-year-old man to 10 years in jail after he admitted pummelling his baby son to death to silence him while watching television.	
Left to Right Generation (1 Summary)	**Confidence Driven Generation (4 Summaries)**
Man who	Man gets 10 years
Man who killed [. . .]	Man who kill the baby gets 10 years
Man who killed baby to hear television better gets 10	Man who kill the baby to hear television gets 10 years
Man who killed baby to hear television better gets 10 years	Man who kill the baby to hear television better gets 10 years

Table 2: An example of the difference between left-to-right and confidence-driven summary generation. (LEFT) A single summary is produced in a left-to-right order. (RIGHT) Four summaries are generated in a confidence-driven mode. The most confident words are generated first, less vital ones later. Our generator learns to dynamically add or remove content given a target length to produce summaries of varying lengths—short, medium and long. The output is a diverse set of alternative summaries.

Identifying admissible summaries with desired properties is critical for a summarizer. Summaries of very short lengths may fail to capture the main concepts, and this kind of incomplete or partial information can lead to false assumptions about the original content. Moreover, summaries of moderate lengths may still contain hallucinated content that is nonexistent in the source text (Maynez et al., 2020). We present two summary selectors to combat these issues. Our first selector aims to predict what summary length is most suitable for a source text, whereas a second selector puts special emphasis on the overall quality of the system summary, in particular its faithfulness to the original text (Falke et al., 2019; Durmus et al., 2020).

A novel dataset has been introduced in this work where we associate a source text with multiple summaries, and *admissible* ones are manually labelled by human annotators. Not only can the dataset be used to judge the effectiveness of summary selectors, but it provides a new testbed for future summarizers to compare their outputs against multiple reference summaries, which is key to improve the reliability of evaluation results (Louis and Nenkova, 2013). We have focused on generating abstractive summaries from single source sentences, but the insights gained from this study could inform the design of summarizers of all forms. Our method also has a great potential to incorporate human-in-the-loop to teach the model to select the best summary. The main contributions of this paper are:

- We propose a new approach to generate multiple variants of the target summary that have varying lengths, then score and select the best summaries according to our needs.
- Our generator controls over the length of the summary, which is especially well-suited when space is limited. Our selectors are designed to predict the optimal summary length and put special emphasis on faithfulness to the original text.
- Our experiments on benchmark summarization datasets suggest that this paradigm can surpass results of previous studies or rival state-of-the-art. We conclude with a discussion of our key findings, which has implications for the development of robust abstractive summarizers.[1]

2 Related Work

It is important for neural abstractive summarizers to produce summaries that are faithful to the original texts (Cao et al., 2017; Kryscinski et al., 2019; Lebanoff et al., 2019a; Wang et al., 2020; Dong et al., 2020; Zhang et al., 2020b). However, it remains questionable as to whether a summarizer must acquire that ability by learning from human reference summaries, or possibly through external resources such as textual entailment predictions (Falke et al., 2019). In this paper, we present a two-stage strategy to over-generate, then score system summaries externally for faithfulness and overall quality.

Previous work has sought to control various aspects of the generated summary, including the style, length and amount of reused text (Kikuchi et al., 2016; Hu et al., 2017; Fan et al., 2018; Keskar et al., 2019; Makino et al., 2019; Song et al., 2020). In contrast, our generator focuses on producing *multiple* variants of the target summary that have diverse content and varying lengths. It offers precise control over the length of the summary, which has an important implication for fair comparison between different summarization systems (Napoles et al., 2011; Shapira et al., 2018).

Our methodology allows for greater flexibility in designing summary selectors. The selectors may allow multiple admissible summaries to be identi-

[1] Our code and annotated data are made available on Github at https://github.com/ucfnlp/varying-length-summ

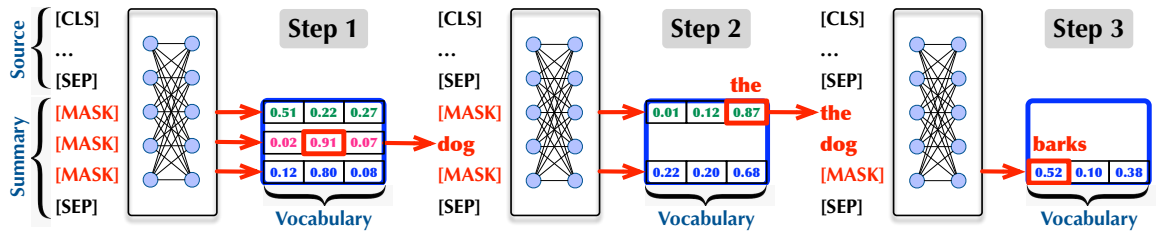

Figure 1: An illustration of the generation process. A sequence of placeholders ("[MASK]") are placed following the source text. Our model simultaneously predicts the most probable tokens for *all positions*, rather than predicting only the most probable *next* token in an autoregressive setting. We obtain the token that has the highest probability, and use it to replace the [MASK] token of that position. Next, the model makes new predictions for all remaining positions, conditioned on the source text and *all summary tokens seen thus far*. Our generator produces a summary having the *exact given length* and with a proper endpoint.

fied for any source input according to users' needs. On the contrary, post-editing of system summaries through a set of basic operations such as insertion and deletion (Gu et al., 2019; Malmi et al., 2019; Dong et al., 2019b; Correia and Martins, 2019) may have intrinsic limitations by learning from single reference summaries to produce single outputs. In this paper, we provide a new dataset where each source text is associated with multiple admissible summaries to encourage diverse outputs.

Our generator is inspired by unsupervised pre-training of deep neural models (Peters et al., 2018; Radford et al., 2019; Devlin et al., 2019; Yan et al., 2020; Zhang et al., 2020a; Lewis et al., 2020) and non-autoregressive machine translation (Gu et al., 2018; Ghazvininejad et al., 2019). Distinct from these is our confidence-driven generation that goes beyond left-to-right order. It uses a denoising objective during training and is conveniently transformed into a semi-autoregressive generator at test time. We introduce a customized beam search algorithm to promote the generation of diverse outputs. In the following section, we describe in detail our two-step strategy.

3 A Confidence-Driven Generator

We seek to produce a highly diverse set of alternative summaries from any source input, but standard neural language generators with beam search only produce high-likelihood sequences rather than diverse ones (Ippolito et al., 2019). To address this limitation, we devise a new generator that is capable of producing summaries of varying lengths. A long summary can cover more important information of the source text, whereas a short summary is easy-to-read. Moreover, it produces a summary having the *exact given length* and with a proper endpoint. This is achieved by shifting away from left-to-right generation but building a summary using a confidence-driven approach.

Our generator is illustrated in Figure 1. To generate a summary of L tokens, we place a number of [MASK] tokens following the source text, which serve as "placeholders" for summary tokens. Importantly, our generator simultaneously predicts the most probable tokens for *all positions*, as opposed to predicting only the most probable *next* token in an autoregressive setting. We obtain the token that has the highest probability across all positions, and use it to replace the [MASK] token of that position. Next, the model continues to make predictions for all remaining positions, conditioned on the source text and the summary tokens seen thus far of varying positions.

Let $\mathbf{x} = \{x_i\}_{i=1}^{N}$ be the source and $\mathbf{y} = \{y_j\}_{j=1}^{M}$ the summary sequence. Our confidence-driven generation process defines a new order of summary tokens, $\mathbf{o} = \{o_j\}_{j=1}^{M}$, $o_j \in [M]$, according to which $P_\theta(\mathbf{y}|\mathbf{x})$ is factorized into a product of conditional probabilities $P_\theta(y_{o_j}|y_{\mathbf{o}_{<j}}, \mathbf{x})$ (Eq. (1)), where θ are model parameters to be optimized during training. Our learning objective is to minimize the negative data log-likelihood (Eq. (2)) to predict missing tokens $y^*_{o_j}$ conditioned on the source text $\mathbf{x}$ and the summary tokens seen thus far $y_{\mathbf{o}_{<j}}$.

$$P_\theta(\mathbf{y}|\mathbf{x};\mathbf{o}) = \prod_{j=1}^{M} P_\theta(y_{o_j}|y_{\mathbf{o}_{<j}}, \mathbf{x}) \quad (1)$$

$$\mathcal{L}(\theta) = -\sum_{j=1}^{M} \log P_\theta(y^*_{o_j}|y_{\mathbf{o}_{<j}}, \mathbf{x}) \quad (2)$$

Our generator is trained with a denoising objective. It consists of a decoder-only architecture with 12 Transformer blocks (Dong et al., 2019a). Given

Input	**The Bank of Japan appealed to financial markets to remain calm Friday following the US decision to order Daiwa Bank Ltd. to close its US operations.**
L=6	BoJ calls for calm. (2,5 4 6 3,1)
L=7	BoJ calls for market calm. (3,7 4 5 6 2,1)
L=8	BoJ urges markets to remain calm. (5,7 6 4 3 1 8,2)
L=9	BoJ urges financial markets to remain calm. (6,2 7 4 5 9 1 8,3)
L=10	BoJ calls for calm after Daiwa closure. (1,2 6 7 5 8 10,4 9,3)
L=11	BoJ calls for calm after Daiwa Bank closure. (1,2 6 7 5 8 11,4 3 10,9)
L=12	BoJ calls for calm after Daiwa Bank closure order. (2,3 5 6 1 11 8,7 9 12 10,4)
L=13	BoJ urges markets to remain calm after Daiwa Bank closure. (6,13 8 7 9 11 4 10 5,2 1 12,3)
L=14	BoJ calls for calm after Daiwa Bank 's US closure. (3,4 7 8 2 14 13,6 5 10,9 11 12,1)
L=15	BoJ calls for calm after US order for Daiwa Bank to close. (10,3 4 5 2 15 8 13 14 9,6 7 11 12,1)
L=16	BoJ calls for calm after US order on Daiwa 's US operations. (3,5 4 7 2 16 13 12 14 8,6 11,10 9 15,1)

Table 3: The target summary length L is adjusted to produce alternative summaries that have diverse content. Our generator can dynamically add or remove content, and paraphrase to produce a summary of a given length. The numbers indicate the order in which the summary tokens are generated. "BoJ" stands for "Bank of Japan". It maps to two tokens according to Byte Pair Encoding (BPE). Each summary has an ending period, so the last word also maps to two tokens.

a source text and a summary, we replace a portion of their tokens by the [MASK] token, and the model is trained to reconstruct the original data from the corrupted text. It differs from autoregressive models in that the context of each position can consist of tokens from both left and right—a source word can attend to other source words and a summary word can attend to source words and summary words seen thus far *of varying positions*—hence capturing a bidirectional context. The training procedure is thus analogous to that of permutation-based language modeling (Yang et al., 2019).

Our training schedule begins with masking out 10% of source tokens and linearly decreases it to 0% throughout training steps. Masking out a portion of source tokens helps the model learn contextualized representations given bidirectional context. On the target side, the schedule begins with masking out 90% of summary tokens and linearly decreases it to 60%. It allows the model to learn to predict missing summary tokens and copy source tokens to the summary. When a token is chosen, it is replaced with the [MASK] token 80% of the time, a random token of the vocabulary 10% of the time, and remains unchanged otherwise.

Algorithm 1 Position-Aware Beam Search

procedure PosAwareBeam(SourceText, L, K)
▶ L is the summary length and K is the beam size.
  $\mathcal{S}_0 \leftarrow \{[\text{MASK}] \times L\}$ ▶ Initial summary.
  $\mathcal{M}_0 \leftarrow [\mathbf{1}]_{L\times|\mathcal{V}|}$ ▶ A binary mask of L positions.
  $\mathcal{H} \leftarrow \{(0, \mathcal{S}_0, \mathcal{M}_0)\}$ ▶ A priority queue.
  for $j = 1, \ldots, L$ **do**
    Candidates $\leftarrow \{\}$
    for *hyp* $\in \mathcal{H}$ **do**
      $score', \mathcal{S}', \mathcal{M}' \leftarrow hyp$
      ▶ Estimate token probabilities.
      $\mathcal{P}_{L\times|\mathcal{V}|} \leftarrow \text{Gen}(\text{SourceText}, \mathcal{S}')$
      $\mathcal{P}' \leftarrow \mathcal{P} \odot \mathcal{M}'$
      ▶ Record K-best tokens and positions.
      for $s_k, w_k, p_k \in$ *Top-K-Scores*$(\mathcal{P}')$ **do**
        $score'' \leftarrow score' + s_k$
        $\mathcal{S}'' \leftarrow replace(\mathcal{S}', p_k, w_k)$
        $\mathcal{M}'' \leftarrow replace(\mathcal{M}', p_k, [\mathbf{0}]_{\mathbf{1}\times|\mathcal{V}|})$
        Candidates.add$((score'', \mathcal{S}'', \mathcal{M}''))$
    $\mathcal{H} \leftarrow$ *Top-K-Scores*(*Candidates*)
return $\mathcal{H}_0$ ▶ The best summary of length L.

In Table 3, we present example summaries produced by our new confidence-driven generator for a source input. The summaries have varying lengths and levels of details. Our generator learns to add or remove content, and even paraphrase to produce a summary of a given length. We adjust the target summary length (L) to produce diverse summaries. Moreover, there exists more than one admissible summaries that capture the important information of the source text, while being grammatical and faithful to the original. It is important to note that, to decode the best summary of length L, our generator requires a *position-aware* beam search algorithm to explore the space of candidate summaries, which is described next.

3.1 Position-Aware Beam Search

A position-aware beam of size K not only contains the K-best candidate summaries having the highest log-likelihood at any time step, but it also records the positions of summary tokens seen thus far for each candidate summary. The tokens of candidate summaries can be decoded in any order and occur in different positions, marking an important distinction between position-aware and traditional beam search (Meister et al., 2020). The method is realized by associating each candidate summary with a binary matrix $\mathcal{M} \in \{0, 1\}_{L\times|\mathcal{V}|}$, which records what positions have been filled by which summary tokens and what positions remain available.

Concretely, we use $\mathcal{S}'$ to denote a candidate sum-

Entity Replacement

- German art experts have authenticated a painting believed to be the last portrait ever made of the composer Wolfgang Amadeus Mozart, the body which runs Berlin's museums said on Thursday.

✔ German experts identify last known portrait of **Mozart**

✘ German experts identify last known portrait of **Mount Mayon's**

Negation

- US Secretary of State Condoleezza Rice suggested Tuesday that International Atomic Energy Agency chief Mohamed ElBaradei should not interfere in diplomatic issues after he warned against the hasty use of force in the Iranian nuclear dispute.

✔ Rice suggests IAEA chief **should** stay clear of diplomacy

✘ Rice suggests IAEA chief **shouldn't** stay clear of diplomacy

Incomplete Summary

- Total Hong Kong dollar deposits grew 2.2 percent in March, compared to 2.1 percent in February, according to the Hong Kong Monetary Authority.

✔ HK Bank Deposits **Increase in March**

✘ **Increase in March**

Search and Replace

- Israel is on course to complete the main tranche of its controversial West Bank security barrier in 2004 and wrap up the project in the following year, the defence ministry said Wednesday

✔ **Israel surges ahead with West Bank barrier construction**

✘ **Soul-searching in Israel over shooting of West Bank barrier protestor**

Swap Segments

- The Security Council on Thursday voted unanimously to extend the mandate of the UN mission in Georgia for four months ahead of next week's international talks on the fallout of the recent Caucasus conflict.

✔ **Security Council extends mandate of UN mission in Georgia**

✘ **UN mission in Georgia Security Council extends mandate of**

Table 4: Corruption types. A positive instance for the selector consists of a ground-truth summary (marked by ✔) and its source text. A negative instance consists of a corrupted summary (✘) and its source text. *Entity Replacement*: replacing a named entity of the ground-truth summary with a random entity. *Negation*: negating a ground-truth summary sentence. *Incomplete Summary*: replacing the ground-truth summary with one of its sentence constituents to produce a corrupted summary that contains 5 words or less. *Search and Replace*: swapping the ground-truth summary with a similar summary in the training set that have 4 or more common bigrams. *Swap Segments*: splitting the ground-truth into two parts of similar length, the parts are swapped to produce an ungrammatical summary.

mary, $score'$ is its data log-likelihood and $\mathcal{M}'$ is a binary mask (Line 9). Our generator predicts the token probabilities $\mathcal{P}_{L\times|\mathcal{V}|}$ for all positions, conditioned on the source text and the summary tokens seen thus far. The binary mask $\mathcal{M}'$ indicates positions that remain available (Line 11–12). We obtain the top-K tokens that have the highest probability scores across all positions, record their summary hypotheses and likelihood scores. These positions are then marked as taken (Line 14–18).

The decoding process continues until all of the L positions are filled by summary tokens. This makes our method different from traditional beam search, the latter terminates when an end-of-sequence symbol [SEP] is generated for the summary. Particularly, our method is advantageous as it exerts *precise control* over the summary length. The model learns to decide what content to be included in the summary given the limited space available, yielding summaries with varying levels of details.

4 The Selectors

We present two selectors to respectively assess the overall quality of the summary and predict the optimal summary length. Our selectors assume the role of a responsible agent that, when provided with a source text and multiple alternative summaries, can effectively recognize the *admissible* ones. It has the potential to incorporate human-in-the-loop in future to teach the model to select best summaries.

4.1 Best Overall Quality

Our goal is to build a selector to discern the difference between high and low-quality summaries. In an ideal scenario, we have human annotators to vet each source text/summary pair, the annotated data are used to train the selector. The process, however, is both expensive and time-consuming. Inspired by Kryściński et al. (2020), we automatically construct a large number of minimally different pairs, where a positive instance comprises of the source text and its ground-truth summary, and a negative instance includes the source text and a corrupted summary. We experiment with various means to generate corrupted summaries from a ground-truth summary. The corruptions should resemble common mistakes made by neural abstractive summarizers, including generating factually incorrect details, failing to convey the main points of the source text, and being ungrammatical. The corruption types experimented in this paper are illustrated in Table 4.

Distinguishing our work from that of Kryściński et al. (2020) are (i) *Search and Replace*, we swap the ground-truth summary with a similar summary in the training set that have $\geq$4 common bigrams to form a negative instance. (ii) *Swap Segments* splits a ground-truth summary into two parts of similar lengths, then swaps them to produce an ungrammatical summary. (iii) *Incomplete Summary* replaces a ground-truth summary by one of its sentence constituents, yielding a corrupted summary

that fails to convey the main ideas. These corruptions are designed to emulate system summaries that are too short to capture the main concepts, or contain hallucinated content that is not found in the source text.

We next build a binary classifier to predict if a summary is admissible given the source text. To distill information from the source text and the summary, we encode them into hidden vectors using RoBERTa (Liu et al., 2019). These are denoted by $\mathbf{h_x}$ and $\mathbf{h_y}$, respectively. We create a vector for the pair, $\mathbf{h} = \mathbf{h_x} \oplus \mathbf{h_y} \oplus |\mathbf{h_x} - \mathbf{h_y}| \oplus (\mathbf{h_x} * \mathbf{h_y})$, consisting of a concatenation of the two hidden vectors, their absolute difference $|\mathbf{h_x} - \mathbf{h_y}|$ and their element-wise product $(\mathbf{h_x} * \mathbf{h_y})$. $\oplus$ is a concatenation of vectors. The output vector $\mathbf{h}$ is expected to capture the gist of the source text and the summary, and a similar approach is being used for natural language inference (Chen et al., 2018). The vector $\mathbf{h}$ is fed to a feed-forward layer to predict whether the summary is admissible given the source text. We have chosen to design the selector as a classifier rather than a ranking model because there can exist multiple, equally valid summaries for any source input. The classifier allows us to identify admissible summaries that are not only true-to-original but has the best overall quality.

4.2 Best Summary Length

Finding a suitable length for the summary is one of the most important open problems in automatic summarization (Shapira et al., 2018; Sun et al., 2019). A summary should be shorter than the original, but long enough to include the most important information. Length normalization seeks to rescale the log-likelihood score of a summary, denoted by $\mathcal{S}(\mathbf{x}, \mathbf{y}) = \log p_\theta(\mathbf{y}|\mathbf{x})$, by its length $|\mathbf{y}|$, with an exponent p (Eq. (3)). It is used by some neural abstractive summarizers (See et al., 2017; Lewis et al., 2020). However, the method does not consider the density of information in the source text and it may still generate ultra-short summaries.

$$\mathcal{S}_{ln}(\mathbf{x}, \mathbf{y}) = \mathcal{S}(\mathbf{x}, \mathbf{y})/|\mathbf{y}|^p \quad (3)$$

Instead, we attempt to estimate the appropriate length of the summary given a source text, denoted by $\mathcal{L}_{\text{pred}}$, and reward a system summary if it stays close to the estimated length (Huang et al., 2017). Concretely, we assign a per-word reward to the summary, represented by $r \min(|\mathbf{y}|, \mathcal{L}_{\text{pred}})$ (Eq. (4)). A system summary continues to be rewarded until it reaches the predicted length ($|\mathbf{y}| \leq \mathcal{L}_{\text{pred}}$). Beyond that, increasing the length of the summary does not lead to additional rewards. We obtain the predicted length $\mathcal{L}_{\text{pred}}$ using a baseline abstractive summarizer, which takes the source text as input and greedily decodes a summary in a left-to-right manner until an end-of-sequence symbol is predicted; $\mathcal{L}_{\text{pred}}$ is the length of the decoding sequence. r is a coefficient to scale the reward and it is tuned on the validation data. Finally, the reward-augmented log-likelihood $\mathcal{S}_{rwd}(\mathbf{x}, \mathbf{y})$ is used as a scoring function to rank all summary hypotheses of varying lengths.

$$\mathcal{S}_{rwd}(\mathbf{x}, \mathbf{y}) = \mathcal{S}(\mathbf{x}, \mathbf{y}) + r \min(|\mathbf{y}|, \mathcal{L}_{\text{pred}}) \quad (4)$$

System	R-1	R-2	R-L
lvt2k-1sent (Nallapati et al., 2016)	32.67	15.59	30.64
SEASS (Zhou et al., 2017)	36.15	17.54	33.63
DRGD (Li et al., 2017)	36.27	17.57	33.62
Pointer-Gen (See et al., 2017)	34.19	16.92	31.81
R3Sum (Cao et al., 2018)	37.04	19.03	34.46
EntailGen (Guo et al., 2018)	35.98	17.76	33.63
BiSET (Wang et al., 2019)	38.45	19.53	36.04
MASS (Song et al., 2019)	38.73	19.71	35.96
UniLM (Dong et al., 2019a)	38.90	20.05	36.00
PEGASUS (Zhang et al., 2020a)	39.12	19.86	36.24
Ours (AVERAGE)	35.51	16.33	32.75
Ours (BEST QUALITY)	36.71	17.27	33.63
Ours (BEST SUMMARY LENGTH)	**39.27**	**20.40**	**36.76**

Table 5: Results on the Gigaword test set evaluated by ROUGE (Lin, 2004).[2]

5 Experiments

Datasets We perform extensive experiments on Gigaword (Parker, 2011) and Newsroom (Grusky et al., 2018) datasets. The goal is to generate an abstractive summary from a lengthy source sentence. For each article, we pair its first sentence with the title to form a summarization instance. Both datasets contain large collections of news articles. Gigaword (1995–2010) contains 3,810,674 / 10,000 / 1,951 instances, respectively, in the train, validation and test splits. Newsroom (1998–2017) contains 199,341 / 21,530 / 21,377 instances, respectively. We conduct experiments on both datasets to demonstrate the generality of our two-staged strategy. Our method generates a diverse set of summaries from a source sentence in stage one, then score and select admissible summaries in stage two.

The system summaries are evaluated using both automatic metrics (ROUGE; Lin, 2004) and human evaluation of information coverage, grammaticality

[2]Our experiments are performed on the original Gigaword dataset (Parker, 2011) without anonymization. The data provided by Rush et al. (2015) replaced all digit characters with # and replaced word types seen less than 5 times with UNK.

Gigaword	Summary Length (L) 7	8	9	10	11	12	13	14	15	16	Avg.	Best Quality	Best Length
R-1 F_1 (%)	32.01	35.42	37.05	37.95	**38.05**	37.79	37.27	36.66	35.75	35.13	36.31	36.71	**39.27**
R-2 F_1 (%)	13.47	15.68	17.39	18.31	**18.24**	18.22	17.85	17.19	16.63	16.00	16.90	17.27	**20.40**
R-L F_1 (%)	29.76	32.85	34.46	**35.31**	35.10	34.87	34.21	33.53	32.71	32.02	33.48	33.63	**36.76**

Newsroom	Summary Length (L) 7	8	9	10	11	12	13	14	15	16	Avg.	Best Quality	Best Length
R-1 F_1 (%)	40.99	43.38	44.94	46.06	46.57	**46.77**	46.53	46.25	45.76	45.21	45.25	45.77	46.60
R-2 F_1 (%)	19.15	20.99	22.11	23.02	23.47	**23.59**	23.38	23.15	22.79	22.33	22.40	22.58	**23.85**
R-L F_1 (%)	38.24	40.34	41.56	42.36	**42.69**	42.68	42.31	41.88	41.29	40.63	41.40	41.48	**43.07**

Table 6: Results on Gigaword and Newsroom datasets where the generator produces summaries of varying lengths.

and faithfulness to the original text. We introduce a new dataset where a source sentence is associated with multiple summaries, and admissible ones are labelled by human annotators (§5.1). The dataset will serve as a useful testbed for future summarization research, where multiple reference summaries is key to improve the reliability of evaluation results (Louis and Nenkova, 2013). This paper focuses on generating abstractive summaries from single source sentences. However, we expect the insights gained from this study to inform the design of future summarizers of different kinds.

Experimental Setup Our generator is initialized with RoBERTa-BASE (Liu et al., 2019) due to its high performance on generation-related tasks. We use Byte Pair Encoding (Sennrich et al., 2016) with a vocabulary of 50,265 tokens. The model contains 12 Transformer blocks (Vaswani et al., 2017), with a hidden size of 768 and 12 attention heads, for a total of 110M parameters. We fine-tune the model on the train split of Gigaword and Newsroom, respectively, before applying it to the test sets. The model is fine-tuned for 20 epochs. Each epoch contains 24k / 1.5k batches and our batch size is 128. The model uses 10k / 1k warm-up steps, respectively, for Gigaword and Newsroom. We use the AdamW (Loshchilov and Hutter, 2017) optimizer with an initial learning rate of 1e-4. The momentum parameters are set to 0.9 and 0.999. On a deep learning workstation equipped with 2x Titan RTX GPUs, our model takes 64 and 5.5 hours to fine-tune on Gigaword and Newsroom. At test time, our beam size is K=20. The model produces summaries ranging from L = 7 to 16 tokens for a given source sentence.

Our selector for best overall quality is trained using 1.8M instances automatically constructed from the train split of Gigaword. The set is balanced with an equal number of positive and negative instances. 226k instances are created with the type of Search and Replace, and 400k instances are created using each of the four remaining corruption types. The reward coefficient r is set to 2.0 across all experiments.

5.1 Experimental Results

Automatic Evaluation In Table 6, we present results on Gigaword and Newsroom test sets evaluated by ROUGE (Lin, 2004). We report R-1, R-2 and R-L F_1-scores that respectively measure the overlap of unigrams, bigrams, and longest common subsequences between system and reference summaries. For each summarization instance, our generator produces multiple alternative summaries, ranging from L=7 to 16 tokens. E.g., "Daiwa Bank." corresponds to four tokens, 'Dai', 'wa', 'Bank' plus an ending period. Our BEST-QUALITY and BEST-LENGTH selectors each identifies a single best summary from the set of alternative summaries for each summarization instance.

We observe that the BEST-LENGTH selector has achieved the highest scores. It performs better than using any single target length for all summaries. Among summaries of different lengths, the highest R-2 F_1-scores are obtained when the target summary length is set to 11 and 12 tokens, respectively, for Gigaword and Newsroom. This is close to the median length of reference summaries, which are 12 and 13 tokens for these datasets. Our findings show that, the target summary length can make a non-negligible impact on automatic evaluation results. It is best for system summaries to be long enough to include the most important information to achieve satisfying results.

In Table 5, we report results on the Gigaword test split that contains 1,951 instances. Our approach is compared against strong neural abstractive systems, including PEGASUS (Zhang et al., 2020a), UniLM (Dong et al., 2019a) and MASS (Song et al., 2019). These systems draw on large-scale unsupervised pretraining to improve the quality of summaries, yielding some of the best reported results. In comparison, our BEST-LENGTH selector either

Candidate Summary	Contains the main idea?	Is true-to-original?	Is grammatical?
(1) Izetbegovic blasts Karadzic	☐ Yes ☒ No	☒ Yes ☐ No	☒ Yes ☐ No
(2) Karadzic accused of swaying US Congress	☒ Yes ☐ No	☒ Yes ☐ No	☒ Yes ☐ No
(3) Karadzic seeks to sway US Congress	☒ Yes ☐ No	☒ Yes ☐ No	☒ Yes ☐ No
(4) Karadzic seeks to sway Congress	☒ Yes ☐ No	☒ Yes ☐ No	☒ Yes ☐ No
(5) Karadzic misleading US Congress	☐ Yes ☒ No	☐ Yes ☒ No	☐ Yes ☒ No
(6) Monday's international soccer scores	☐ Yes ☒ No	☐ Yes ☒ No	☐ Yes ☒ No
Source Text: Bosnian President Alija Izetbegovic on Monday accused Bosnian Serb leader Radovan Karadzic of seeking to sway the US Congress against approving US troops to help enforce peace in the former Yugoslavia.			

Table 7: Example annotation interface. A human annotator is instructed to read over the summaries before seeing the source text to effectively recognize any hallucinated content that is not found in the source text. A native English speaker creates annotations for multiple instances, which are shared with all annotators to provide guidance.

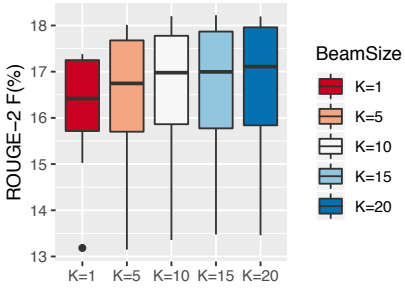

Figure 2: Effectiveness of position-aware beam search (§3.1). A larger beam tends to give better results.

	Content	Truthful	Grammatical	Overall
Average	80.7	82.6	96.5	74.2
Best Length	82.8	86.0	**97.4**	77.8
Best Quality	**93.0**	**90.8**	97.0	**88.2**

Table 8: Results of human assessment. BEST-QUALITY summaries have a higher likelihood of being admissible according to the criteria, suggesting the effectiveness of the method.

surpasses or performs comparably to these systems. The summaries selected by it achieve the highest R-2 F_1-score of 20.4%. We further choose the summary that yields the highest score for each instance, creating an oracle set of summaries, which yield a R-2 F_1-score of 33.4%. The results indicate that, with better summary selectors, there is a great potential that we can further boost summarization performance.

In Figure 2, we investigate the effectiveness of our position-aware beam search (§3.1). The beam size K is set to $\{1, 5, 10, 15, 20\}$. We report the average R-2 F_1-score across summaries of all lengths. Results show that our position-aware beam search is effective at decoding summaries and works robustly across a range of beam sizes. A larger beam (K=20) tends to give better results.

Human Evaluation We are interested in a holistic evaluation of the multiple alternative summaries produced by the generator. To accomplish this, we develop a new dataset containing 500 summarization instances randomly sampled from the Gigaword test set. Our generator produces 7 alternative summaries for each instance, which have varying lengths that range from L= 7 to 13 tokens. We recruit human evaluators to judge the quality of each summary given its source text.[3]

[3]Our annotated dataset is available on Github at https://github.com/ucfnlp/varying-length-summ

Our annotation interface is presented in Table 7. A human annotator is instructed to read over all summaries before seeing the source text. It allows him/her to effectively recognize any hallucinated content that is not found in the source text. The annotator is asked to answer three yes-no questions. They include (a) has the summary successfully convey the main points of the source text? (b) is the summary truthful to the meaning of the original? (c) is the summary grammatical? A native speaker creates gold-standard annotations for multiple instances, they are shared with all annotators to provide guidance. Our annotators are recruited using Appen (appen.com). It is a crowdsourcing platform similar to Amazon Mechanical Turk (mturk.com), but provides great quality control mechanisms to ensure high-quality work.

We recruit 5 annotators to judge the quality of each summary. A summary is deemed *admissible* under a criterion if the majority answer is yes. We observe that, 74.2% of summaries produced by our generator are admissible under all three criteria. The results suggest that our generator is able to produce multiple, equally valid summaries for a given source text. We additionally examine the percentage of admissible summaries under each criterion, results are shown in Table 8. Grammaticality has the best performance (96.5%), followed by truthfulness (82.6%) and content coverage (80.7%). There appears to be room for improvement for the latter two aspects. Moreover, the summaries chosen by our BEST-QUALITY selector demonstrate a high ad-

missible rate—93%, 90.8% and 97%—respectively for the three criteria, suggesting the effectiveness of the selector. Further, we observe a discrepancy between ROUGE and human judgments (Fabbri et al., 2020) as summaries yielding highest ROUGE scores are not always deemed admissible by human evaluators. We hope this dataset provides a testbed for future summarizers to be judged on their ability to produce multiple summaries per instance rather than a single summary.

In Table 3, we show example system summaries and the order in which summary tokens are produced. E.g., {2,5} indicate the two tokens "Bo-J" (Bank of Japan) are generated the 2nd and 5th place in the summary. We find that our generator can effectively decide what content should be included in the summary given the limited space available, yielding summaries with varying levels of details. Important spans such as "calls for calm" tend to be generated first, less vital ones later. Our findings corroborate the hypothesis that a masked language model may enable generation in a flexible word order (Liao et al., 2020). Further, we observe that the order in which tokens are generated is related to their dependencies ("call→for"), which supports the findings of Clark et al. (2019).

6 Conclusion

We investigate a new approach to neural abstractive summarization that focuses on producing multiple summary hypotheses with varying lengths and levels of details. Our selectors are designed to identify summaries that have the optimal length and the best overall quality. The approach obtains state-of-the-art results on summarization benchmarks and opens up a potential new avenue for customizing summary selectors to suit users' needs.

Future work includes extending this research to long documents. Our confidence-driven generator and the selectors could potentially be extended to operate on spans of text (Joshi et al., 2020) rather than individual tokens, thus allowing for efficient generation of multiple summary hypotheses and identification of admissible summaries and/or summary segments.

Acknowledgements

We are grateful to the reviewers for their insightful comments, which have helped us improve the paper. This research was supported in part by the National Science Foundation grant IIS-1909603.

References

Anja Belz and Ehud Reiter. 2006. Comparing automatic and human evaluation of NLG systems. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, Trento, Italy. Association for Computational Linguistics.

Arthur Bražinskas, Mirella Lapata, and Ivan Titov. 2020. Few-shot learning for opinion summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4119–4135, Online. Association for Computational Linguistics.

Ziqiang Cao, Wenjie Li, Sujian Li, and Furu Wei. 2018. Retrieve, rerank and rewrite: Soft template based neural summarization. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 152–161, Melbourne, Australia. Association for Computational Linguistics.

Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. 2017. Faithful to the original: Fact aware neural abstractive summarization. *CoRR*, abs/1711.04434.

Asli Celikyilmaz, 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, Volume 1 (Long Papers)*, pages 1662–1675, New Orleans, Louisiana. Association for Computational Linguistics.

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 (Volume 1: Long Papers)*, pages 675–686, Melbourne, Australia. Association for Computational Linguistics.

Yufei Chen, Sheng Huang, Fang Wang, Junjie Cao, Weiwei Sun, and Xiaojun Wan. 2018. Neural maximum subgraph parsing for cross-domain semantic dependency analysis. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 562–572, Brussels, Belgium. Association for Computational Linguistics.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

Gonçalo M. Correia and André F. T. Martins. 2019. A simple and effective approach to automatic post-editing with transfer learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3050–3056, Florence, Italy. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019a. Unified language model pre-training for natural language understanding and generation. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 13063–13075. Curran Associates, Inc.

Yue Dong, Zichao Li, Mehdi Rezagholizadeh, and Jackie Chi Kit Cheung. 2019b. EditNTS: An neural programmer-interpreter model for sentence simplification through explicit editing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3393–3402, Florence, Italy. Association for Computational Linguistics.

Yue Dong, Shuohang Wang, Zhe Gan, Yu Cheng, Jackie Chi Kit Cheung, and Jingjing Liu. 2020. Multi-fact correction in abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9320–9331, Online. Association for Computational Linguistics.

Esin Durmus, He He, and Mona Diab. 2020. FEQA: A question answering evaluation framework for faithfulness assessment in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5055–5070, Online. Association for Computational Linguistics.

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

Alexander R. Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2020. Summeval: Re-evaluating summarization evaluation. *arXiv preprint arXiv:2007.12626*.

Tobias Falke, Leonardo F. R. Ribeiro, Prasetya Ajie Utama, Ido Dagan, and Iryna Gurevych. 2019. Ranking generated summaries by correctness: An interesting but challenging application for natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2214–2220, Florence, Italy. 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*, pages 45–54, Melbourne, Australia. Association for Computational Linguistics.

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

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. 2019. Mask-predict: Parallel decoding of conditional masked language models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6112–6121, Hong Kong, China. Association for Computational Linguistics.

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, Volume 1 (Long Papers)*, pages 708–719, New Orleans, Louisiana. Association for Computational Linguistics.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor O.K. Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. In *International Conference on Learning Representations*.

Jiatao Gu, Changhan Wang, and Junbo Zhao. 2019. Levenshtein transformer. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 11181–11191. Curran Associates, Inc.

Han Guo, Ramakanth Pasunuru, and Mohit Bansal. 2018. Soft layer-specific multi-task summarization with entailment and question generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 687–697, Melbourne, Australia. Association for Computational Linguistics.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. 2017. Controllable text generation. *CoRR*, abs/1703.00955.

Liang Huang, Kai Zhao, and Mingbo Ma. 2017. When to finish? optimal beam search for neural text generation (modulo beam size). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2134–2139, Copenhagen, Denmark. Association for Computational Linguistics.

Daphne Ippolito, Reno Kriz, João Sedoc, Maria Kustikova, and Chris Callison-Burch. 2019. Comparison of diverse decoding methods from conditional language models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3752–3762, Florence, Italy. Association for Computational Linguistics.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Nitish Shirish Keskar, Bryan McCann, Lav Varshney, Caiming Xiong, and Richard Socher. 2019. CTRL - A Conditional Transformer Language Model for Controllable Generation. *arXiv preprint arXiv: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*, pages 1328–1338, Austin, Texas. Association for Computational Linguistics.

Wojciech Kryscinski, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Neural text summarization: A critical evaluation. 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 540–551, Hong Kong, China. Association for Computational Linguistics.

Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346, Online. Association for Computational Linguistics.

Logan Lebanoff, John Muchovej, Franck Dernoncourt, Doo Soon Kim, Seokhwan Kim, Walter Chang, and Fei Liu. 2019a. Analyzing sentence fusion in abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 104–110, Hong Kong, China. Association for Computational Linguistics.

Logan Lebanoff, Kaiqiang Song, Franck Dernoncourt, Doo Soon Kim, Seokhwan Kim, Walter Chang, and Fei Liu. 2019b. Scoring sentence singletons and pairs for abstractive summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2175–2189, Florence, Italy. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Piji Li, Wai Lam, Lidong Bing, and Zihao Wang. 2017. Deep recurrent generative decoder for abstractive text summarization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2091–2100, Copenhagen, Denmark. Association for Computational Linguistics.

Yi Liao, Xin Jiang, and Qun Liu. 2020. Probabilistically masked language model capable of autoregressive generation in arbitrary word order. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 263–274, Online. Association for Computational Linguistics.

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

Yang Liu and Mirella Lapata. 2019. Hierarchical transformers for multi-document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5070–5081, Florence, Italy. Association for Computational Linguistics.

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

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Annie Louis and Ani Nenkova. 2013. Automatically assessing machine summary content without a gold standard. *Computational Linguistics*, 39(2):267–300.

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.

Eric Malmi, Sebastian Krause, Sascha Rothe, Daniil Mirylenka, and Aliaksei Severyn. 2019. Encode, tag, realize: High-precision text editing. 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 5054–5065, Hong

Kong, China. Association for Computational Linguistics.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.

Clara Meister, Ryan Cotterell, and Tim Vieira. 2020. If beam search is the answer, what was the question? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2173–2185, Online. Association for Computational Linguistics.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany. Association for Computational Linguistics.

Courtney Napoles, Benjamin Van Durme, and Chris Callison-Burch. 2011. Evaluating sentence compression: Pitfalls and suggested remedies. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, pages 91–97, Portland, Oregon. Association for Computational Linguistics.

Robert Parker. 2011. English Gigaword fifth edition LDC2011T07. *Philadelphia: Linguistic Data Consortium*.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

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

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Ori Shapira, David Gabay, Hadar Ronen, Judit Bar-Ilan, Yael Amsterdamer, Ani Nenkova, and Ido Dagan. 2018. Evaluating multiple system summary lengths: A case study. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 774–778, Brussels, Belgium. Association for Computational Linguistics.

Kaiqiang Song, Bingqing Wang, Zhe Feng, Liu Ren, and Fei Liu. 2020. Controlling the amount of verbatim copying in abstractive summarization. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI)*.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. MASS: Masked sequence to sequence pre-training for language generation. In *Proceedings of International Conference on Machine Learning (ICML)*.

Simeng Sun, Ori Shapira, Ido Dagan, and Ani Nenkova. 2019. How to compare summarizers without target length? pitfalls, solutions and re-examination of the neural summarization literature. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pages 21–29, Minneapolis, Minnesota. Association for Computational Linguistics.

Sho Takase, Jun Suzuki, Naoaki Okazaki, Tsutomu Hirao, and Masaaki Nagata. 2016. Neural headline generation on Abstract Meaning Representation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1054–1059, Austin, Texas. Association for Computational Linguistics.

Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. 2017. Abstractive document summarization with a graph-based attentional neural model. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1171–1181, Vancouver, Canada. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. Asking and answering questions to evaluate the factual consistency of summaries. In *Proceedings of*

the 58th Annual Meeting of the Association for Computational Linguistics, pages 5008–5020, Online. Association for Computational Linguistics.

Kai Wang, Xiaojun Quan, and Rui Wang. 2019. BiSET: Bi-directional selective encoding with template for abstractive summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2153–2162, Florence, Italy. Association for Computational Linguistics.

Yu Yan, Weizhen Qi, Yeyun Gong, Dayiheng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang, and Ming Zhou. 2020. Prophetnet: Predicting future n-gram for sequence-to-sequence pre-training.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 5753–5763. Curran Associates, Inc.

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)*.

Yuhao Zhang, Derek Merck, Emily Tsai, Christopher D. Manning, and Curtis Langlotz. 2020b. 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*, pages 5108–5120, Online. Association for Computational Linguistics.

Qingyu Zhou, Nan Yang, Furu Wei, and Ming Zhou. 2017. Selective encoding for abstractive sentence summarization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1095–1104, Vancouver, Canada. Association for Computational Linguistics.