Bottom-Up Abstractive Summarization

Sebastian Gehrmann **Yuntian Deng** **Alexander M. Rush**
School of Engineering and Applied Sciences
Harvard University
{gehrmann, dengyuntian, srush}@seas.harvard.edu

Abstract

Neural network-based methods for abstractive summarization produce outputs that are more fluent than other techniques, but perform poorly at content selection. This work proposes a simple technique for addressing this issue: use a data-efficient content selector to over-determine phrases in a source document that should be part of the summary. We use this selector as a *bottom-up* attention step to constrain the model to likely phrases. We show that this approach improves the ability to compress text, while still generating fluent summaries. This two-step process is both simpler and higher performing than other end-to-end content selection models, leading to significant improvements on ROUGE for both the CNN-DM and NYT corpus. Furthermore, the content selector can be trained with as little as 1,000 sentences, making it easy to transfer a trained summarizer to a new domain.

1 Introduction

Text summarization systems aim to generate natural language summaries that compress the information in a longer text. Approaches using neural networks have shown promising results on this task with end-to-end models that encode a source document and then decode it into an abstractive summary. Current state-of-the-art neural abstractive summarization models combine extractive and abstractive techniques by using pointer-generator style models which can copy words from the source document (Gu et al., 2016; See et al., 2017). These end-to-end models produce fluent abstractive summaries but have had mixed success in content selection, i.e. deciding what to summarize, compared to fully extractive models.

There is an appeal to end-to-end models from a modeling perspective; however, there is evidence that when summarizing people follow a two-step approach of first selecting important phrases and then paraphrasing them (Anderson and Hidi, 1988; Jing and McKeown, 1999). A similar argument has been made for image captioning. Anderson et al. (2017) develop a state-of-the-art model with a two-step approach that first pre-computes bounding boxes of segmented objects and then applies attention to these regions. This so-called *bottom-up* attention is inspired by neuroscience research describing attention based on properties in-

Source Document
german chancellor angela merkel [did] not [look] too pleased about the weather during her [annual] easter holiday [in italy.] as britain [basks] in [sunshine] and temperatures of up to 21c, mrs merkel and her husband[, chemistry professor joachim sauer,] had to settle for a measly 12 degrees. the chancellor and her [spouse] have been spending easter on the small island of ischia, near naples in the mediterranean for over a [decade.]
[not so sunny:] angela merkel [and] her husband[, chemistry professor joachim sauer,] are spotted on their [annual] easter trip to the island of ischia[,] near naples[. the] couple [traditionally] spend their holiday at the five-star miramare spa hotel on the south of the island [, which comes] with its own private beach [, and balconies overlooking the] ocean [.]...

Reference
- angela merkel and husband spotted while on italian island holiday.

...

Baseline Approach
- angela merkel and her husband, chemistry professor joachim sauer, are spotted on their annual easter trip to the island of ischia, near naples.

...

Bottom-Up Summarization
- angela merkel and her husband are spotted on their easter trip to the island of ischia, near naples.

...

Figure 1: Example of two sentence summaries with and without bottom-up attention. The model does not allow copying of words in [gray], although it can generate words. With bottom-up attention, we see more explicit sentence compression, while without it whole sentences are copied verbatim.

herent to a stimulus (Buschman and Miller, 2007).

Motivated by this approach, we consider *bottom-up* attention for neural abstractive summarization. Our approach first selects a selection mask for the source document and then constrains a standard neural model by this mask. This approach can better decide which phrases a model should include in a summary, without sacrificing the fluency advantages of neural abstractive summarizers. Furthermore, it requires much fewer data to train, which makes it more adaptable to new domains.

Our full model incorporates a separate content selection system to decide on relevant aspects of the source document. We frame this selection task as a sequence-tagging problem, with the objective of identifying tokens from a document that are part of its summary. We show that a content selection model that builds on contextual word embeddings (Peters et al., 2018) can identify correct tokens with a recall of over 60%, and a precision of over 50%. To incorporate bottom-up attention into abstractive summarization models, we employ masking to constrain copying words to the selected parts of the text, which produces grammatical outputs. We additionally experiment with multiple methods to incorporate similar constraints into the training process of more complex end-to-end abstractive summarization models, either through multi-task learning or through directly incorporating a fully differentiable mask.

Our experiments compare bottom-up attention with several other state-of-the-art abstractive systems. Compared to our baseline models of See et al. (2017) bottom-up attention leads to an improvement in ROUGE-L score on the CNN-Daily Mail (CNN-DM) corpus from 36.4 to 38.3 while being simpler to train. We also see comparable or better results than recent reinforcement-learning based methods with our MLE trained system. Furthermore, we find that the content selection model is very data-efficient and can be trained with less than 1% of the original training data. This provides opportunities for domain-transfer and low-resource summarization. We show that a summarization model trained on CNN-DM and evaluated on the NYT corpus can be improved by over 5 points in ROUGE-L with a content selector trained on only 1,000 in-domain sentences.

2 Related Work

There is a tension in document summarization between staying close to the source document and allowing compressive or abstractive modification. Many non-neural systems take a select and compress approach. For example, Dorr et al. (2003) introduced a system that first extracts noun and verb phrases from the first sentence of a news article and uses an iterative shortening algorithm to compress it. Recent systems such as Durrett et al. (2016) also learn a model to select sentences and then compress them.

In contrast, recent work in neural network based data-driven extractive summarization has focused on extracting and ordering full sentences (Cheng and Lapata, 2016; Dlikman and Last, 2016). Nallapati et al. (2016b) use a classifier to determine whether to include a sentence and a selector that ranks the positively classified ones. These methods often over-extract, but extraction at a word level requires maintaining grammatically correct output (Cheng and Lapata, 2016), which is difficult. Interestingly, key phrase extraction while ungrammatical often matches closely in content with human-generated summaries (Bui et al., 2016).

A third approach is neural abstractive summarization with sequence-to-sequence models (Sutskever et al., 2014; Bahdanau et al., 2014). These methods have been applied to tasks such as headline generation (Rush et al., 2015) and article summarization (Nallapati et al., 2016a). Chopra et al. (2016) show that attention approaches that are more specific to summarization can further improve the performance of models. Gu et al. (2016) were the first to show that a copy mechanism, introduced by Vinyals et al. (2015), can combine the advantages of both extractive and abstractive summarization by copying words from the source. See et al. (2017) refine this pointer-generator approach and use an additional coverage mechanism (Tu et al., 2016) that makes a model aware of its attention history to prevent repeated attention.

Most recently, reinforcement learning (RL) approaches that optimize objectives for summarization other than maximum likelihood have been shown to further improve performance on these tasks (Paulus et al., 2017; Li et al., 2018b; Celikyilmaz et al., 2018). Paulus et al. (2017) approach the coverage problem with an intra-attention in which a decoder has an attention over previously generated words. However RL-based training can

be difficult to tune and slow to train. Our method does not utilize RL training, although in theory this approach can be adapted to RL methods.

Several papers also explore multi-pass extractive-abstractive summarization. Nallapati et al. (2017) create a new source document comprised of the important sentences from the source and then train an abstractive system. Liu et al. (2018) describe an extractive phase that extracts full paragraphs and an abstractive one that determines their order. Finally Zeng et al. (2016) introduce a mechanism that reads a source document in two passes and uses the information from the first pass to bias the second. Our method differs in that we utilize a completely abstractive model, biased with a powerful content selector.

Other recent work explores alternative approaches to content selection. For example, Cohan et al. (2018) use a hierarchical attention to detect relevant sections in a document, Li et al. (2018a) generate a set of keywords that is used to guide the summarization process, and Pasunuru and Bansal (2018) develop a loss-function based on whether salient keywords are included in a summary. Other approaches investigate the content-selection at the sentence-level. Tan et al. (2017) describe a graph-based attention to attend to one sentence at a time, Chen and Bansal (2018) first extract full sentences from a document and then compress them, and Hsu et al. (2018) modulate the attention based on how likely a sentence is included in a summary.

3 Background: Neural Summarization

Throughout this paper, we consider a set of pairs of texts $(\mathcal{X}, \mathcal{Y})$ where $x \in \mathcal{X}$ corresponds to source tokens $x_1, \ldots, x_n$ and $y \in \mathcal{Y}$ to a summary $y_1, \ldots, y_m$ with $m \ll n$.

Abstractive summaries are generated one word at a time. At every time-step, a model is aware of the previously generated words. The problem is to learn a function $f(x)$ parametrized by θ that maximizes the probability of generating the correct sequences. Following previous work, we model the abstractive summarization with an attentional sequence-to-sequence model. The attention distribution $p(a_j|x, y_{1:j-1})$ for a decoding step j, calculated within the neural network, represents an embedded soft distribution over all of the source tokens and can be interpreted as the current focus of the model.

The model additionally has a copy mechanism (Vinyals et al., 2015) to copy words from the source. Copy models extend the decoder by predicting a binary soft switch z_j that determines whether the model copies or generates. The copy distribution is a probability distribution over the source text, and the joint distribution is computed as a convex combination of the two parts of the model,

$$
\begin{aligned}
&p(y_j \mid y_{1:j\text{-}1}, x) = \\
&\quad p(z_j = 1 \mid y_{1:j\text{-}1}, x) \times p(y_j \mid z_j = 1, y_{1:j\text{-}1}, x) + \\
&\quad p(z_j = 0 \mid y_{1:j\text{-}1}, x) \times p(y_j \mid z_j = 0, y_{1:j\text{-}1}, x)
\end{aligned} \tag{1}
$$

where the two parts represent copy and generation distribution respectively. Following the *pointer-generator* model of See et al. (2017), we reuse the attention $p(a_j|x, y_{1:j-1})$ distribution as copy distribution, i.e. the copy probability of a token in the source w through the copy attention is computed as the sum of attention towards all occurrences of w. During training, we maximize marginal likelihood with the latent switch variable.

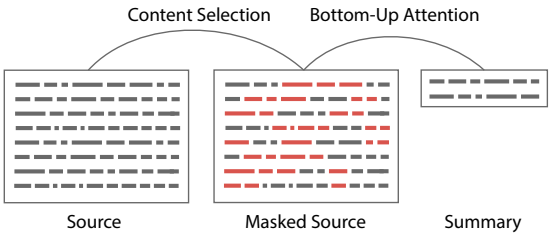

Figure 2: Overview of the selection and generation processes described throughout Section 4.

4 Bottom-Up Attention

We next consider techniques for incorporating a content selection into abstractive summarization, illustrated in Figure 2.

4.1 Content Selection

We define the content selection problem as a word-level extractive summarization task. While there has been significant work on custom extractive summarization (see related work), we make a simplifying assumption and treat it as a sequence tagging problem. Let $t_1, \ldots, t_n$ denote binary tags for each of the source tokens, i.e. 1 if a word is copied in the target sequence and 0 otherwise.

While there is no supervised data for this task, we can generate training data by aligning the summaries to the document. We define a word x_i as

copied if (1) it is part of the longest possible subsequence of tokens $s = x_{i-j:i:i+k}$, for integers $j \leq i; k \leq (n-i)$, if $s \in x$ and $s \in y$, and (2) there exists no earlier sequence u with $s = u$.

We use a standard bidirectional LSTM model trained with maximum likelihood for the sequence labeling problem. Recent results have shown that better word representations can lead to significantly improved performance in sequence tagging tasks (Peters et al., 2017). Therefore, we first map each token w_i into two embedding channels $e_i^{(\mathrm{w})}$ and $e_i^{(\mathrm{c})}$. The $e^{(\mathrm{w})}$ embedding represents a static channel of pre-trained word embeddings, e.g. GLoVE (Pennington et al., 2014). The $e^{(\mathrm{c})}$ are contextual embeddings from a pretrained language model, e.g. ELMo (Peters et al., 2018) which uses a character-aware token embedding (Kim et al., 2016) followed by two bidirectional LSTM layers $h_i^{(1)}$ and $h_i^{(2)}$. The contextual embeddings are fine-tuned to learn a task-specific embedding $e_i^{(\mathrm{c})}$ as a linear combination of the states of each LSTM layer and the token embedding,

$$e_i^{(\mathrm{c})} = \gamma \times \sum_{\ell=0}^{2} s_j \times h_i^{(\ell)},$$

with γ and $s_{0,1,2}$ as trainable parameters. Since these embeddings only add four additional parameters to the tagger, it remains very data-efficient despite the high-dimensional embedding space.

Both embeddings are concatenated into a single vector that is used as input to a bidirectional LSTM, which computes a representation h_i for a word w_i. We can then calculate the probability q_i that the word is selected as $\sigma(W_s h_i + b_s)$ with trainable parameters W_s and b_s.

4.2 Bottom-Up Copy Attention

Inspired by work in bottom-up attention for images (Anderson et al., 2017) which restricts attention to predetermined bounding boxes within an image, we use these attention masks to limit the available selection of the pointer-generator model.

As shown in Figure 1, a common mistake made by neural copy models is copying very long sequences or even whole sentences. In the baseline model, over 50% of copied tokens are part of copy sequences that are longer than 10 tokens, whereas this number is only 10% for reference summaries. While bottom-up attention could also be used to modify the source encoder representations, we found that a standard encoder over the full text was effective at aggregation and therefore limit the bottom-up step to attention masking.

Concretely, we first train a pointer-generator model on the full dataset as well as the content selector defined above. At inference time, to generate the mask, the content selector computes selection probabilities $q_{1:n}$ for each token in a source document. The selection probabilities are used to modify the copy attention distribution to only include tokens identified by the selector. Let a_j^i denote the attention at decoding step j to encoder word i. Given a threshold ϵ, the selection is applied as a hard mask, such that

$$p(\tilde{a}_j^i | x, y_{1:j-1}) = \begin{cases} p(a_j^i | x, y_{1:j-1}) & q_i > \epsilon \\ 0 & \text{ow.} \end{cases}$$

To ensure that Eq. 1 still yields a correct probability distribution, we first multiply $p(\tilde{a}_j | x, y_{1:j-1})$ by a normalization parameter λ and then renormalize the distribution. The resulting normalized distribution can be used to directly replace a as the new copy probabilities.

4.3 End-to-End Alternatives

Two-step BOTTOM-UP attention has the advantage of training simplicity. In theory, though, standard copy attention should be able to learn how to perform content selection as part of the end-to-end training. We consider several other end-to-end approaches for incorporating content selection into neural training.

Method 1: (MASK ONLY): We first consider whether the alignment used in the bottom-up approach could help a standard summarization system. Inspired by Nallapati et al. (2017), we investigate whether aligning the summary and the source during training and fixing the gold copy attention to pick the "correct" source word is beneficial. We can think of this approach as limiting the set of possible copies to a fixed source word. Here the training is changed, but no mask is used at test time.

Method 2 (MULTI-TASK): Next, we investigate whether the content selector can be trained alongside the abstractive system. We first test this hypothesis by posing summarization as a multi-task problem and training the tagger and summarization model with the same features. For this setup, we use a shared encoder for both abstractive summarization and content selection. At test time, we

apply the same masking method as bottom-up attention.

Method 3 (DIFFMASK): Finally we consider training the full system end-to-end with the mask during training. Here we jointly optimize both objectives, but use predicted selection probabilities to softly mask the copy attention $p(\tilde{a}^i_j|x, y_{1:j-1}) = p(a^i_j|x, y_{1:j-1}) \times q_i$, which leads to a fully differentiable model. This model is used with the same soft mask at test time.

5 Inference

Several authors have noted that longer-form neural generation still has significant issues with incorrect length and repeated words than in short-form problems like translation. Proposed solutions include modifying models with extensions such as a coverage mechanism (Tu et al., 2016; See et al., 2017) or intra-sentence attention (Cheng et al., 2016; Paulus et al., 2017). We instead stick to the theme of modifying inference, and modify the scoring function to include a length penalty lp and a coverage penalty cp, and is defined as $s(x, y) = \log p(y|x)/lp(x) + cp(x; y)$.

Length: To encourage the generation of longer sequences, we apply length normalizations during beam search. We use the length penalty by Wu et al. (2016), which is formulated as

$$lp(y) = \frac{(5 + |y|)^\alpha}{(5 + 1)^\alpha},$$

with a tunable parameter α, where increasing α leads to longer summaries. We additionally set a minimum length based on the training data.

Repeats: Copy models often repeatedly attend to the same source tokens, generating the same phrase multiple times. We introduce a new summary specific coverage penalty,

$$cp(x; y) = \beta \left(-n + \sum_{i=1}^{n} \max \left(1.0, \sum_{j=1}^{m} a^j_i \right) \right).$$

Intuitively, this penalty increases whenever the decoder directs more than 1.0 of total attention within a sequence towards a single encoded token. By selecting a sufficiently high β, this penalty blocks summaries whenever they would lead to repetitions. Additionally, we follow (Paulus et al., 2017) and restrict the beam search to never repeat trigrams.

6 Data and Experiments

We evaluate our approach on the CNN-DM corpus (Hermann et al., 2015; Nallapati et al., 2016a), and the NYT corpus (Sandhaus, 2008), which are both standard corpora for news summarization. The summaries for the CNN-DM corpus are bullet points for the articles shown on their respective websites, whereas the NYT corpus contains summaries written by library scientists. CNN-DM summaries are full sentences, with on average 66 tokens ($\sigma = 26$) and 4.9 bullet points. NYT summaries are not always complete sentences and are shorter, with on average 40 tokens ($\sigma = 27$) and 1.9 bullet points. Following See et al. (2017), we use the non-anonymized version of the CNN-DM corpus and truncate source documents to 400 tokens and the target summaries to 100 tokens in training and validation sets. For experiments with the NYT corpus, we use the preprocessing described by Paulus et al. (2017), and additionally remove author information and truncate source documents to 400 tokens instead of 800. These changes lead to an average of 326 tokens per article, a decrease from the 549 tokens with 800 token truncated articles. The target (non-copy) vocabulary is limited to 50,000 tokens for all models.

The content selection model uses pre-trained GloVe embeddings of size 100, and ELMo with size 1024. The bi-LSTM has two layers and a hidden size of 256. Dropout is set to 0.5, and the model is trained with Adagrad, an initial learning rate of 0.15, and an initial accumulator value of 0.1. We limit the number of training examples to 100,000 on either corpus, which only has a small impact on performance. For the jointly trained content selection models, we use the same configuration as the abstractive model.

For the base model, we re-implemented the Pointer-Generator model as described by See et al. (2017). To have a comparable number of parameters to previous work, we use an encoder with 256 hidden states for both directions in the one-layer LSTM, and 512 for the one-layer decoder. The embedding size is set to 128. The model is trained with the same Adagrad configuration as the content selector. Additionally, the learning rate halves after each epoch once the validation perplexity does not decrease after an epoch. We do not use dropout and use gradient-clipping with a maximum norm of 2. We found that increasing model size or using the Transformer (Vaswani et al.,

Method	R-1	R-2	R-L
Pointer-Generator (See et al., 2017)	36.44	15.66	33.42
Pointer-Generator + Coverage (See et al., 2017)	39.53	17.28	36.38
ML + Intra-Attention (Paulus et al., 2017)	38.30	14.81	35.49
ML + RL (Paulus et al., 2017)	39.87	15.82	36.90
Saliency + Entailment reward (Pasunuru and Bansal, 2018)	40.43	18.00	37.10
Key information guide network (Li et al., 2018a)	38.95	17.12	35.68
Inconsistency loss (Hsu et al., 2018)	40.68	17.97	37.13
Sentence Rewriting (Chen and Bansal, 2018)	40.88	17.80	**38.54**
Pointer-Generator (our implementation)	36.25	16.17	33.41
Pointer-Generator + Coverage Penalty	39.12	17.35	36.12
CopyTransformer + Coverage Penalty	39.25	17.54	36.45
Pointer-Generator + Mask Only	37.70	15.63	35.49
Pointer-Generator + Multi-Task	37.67	15.59	35.47
Pointer-Generator + DiffMask	38.45	16.88	35.81
Bottom-Up Summarization	**41.22**	**18.68**	38.34
Bottom-Up Summarization (CopyTransformer)	40.96	18.38	38.16

Table 1: Results of abstractive summarizers on the CNN-DM dataset.[2] The first section shows encoder-decoder abstractive baselines trained with cross-entropy. The second section describes reinforcement-learning based approaches. The third section presents our baselines and the attention masking methods described in this work.

2017) can lead to slightly improved performance, but at the cost of increased training time and parameters. We report numbers of a Transformer with copy-attention, which we denote CopyTransformer. In this model, we randomly choose one of the attention-heads as the copy-distribution, and otherwise follow the parameters of the big Transformer by Vaswani et al. (2017).

All inference parameters are tuned on a 200 example subset of the validation set. Length penalty parameter α and copy mask ϵ differ across models, with α ranging from 0.6 to 1.4, and ϵ ranging from 0.1 to 0.2. The minimum length of the generated summary is set to 35 for CNN-DM and 6 for NYT. While the Pointer-Generator uses a beam size of 5 and does not improve with a larger beam, we found that bottom-up attention requires a larger beam size of 10. The coverage penalty parameter β is set to 10, and the copy attention normalization parameter λ to 2 for both approaches. We use AllenNLP (Gardner et al., 2018) for the content selector, and OpenNMT-py for the abstractive models (Klein et al., 2017).[3].

[3]Code and reproduction instructions can be found at https://github.com/sebastianGehrmann/bottom-up-summary

[3]These results compare on the non-anonymized version of this corpus used by (See et al., 2017). The best results on the anonymized version are R1:41.69 R2:19.47 RL:37.92 from (Celikyilmaz et al., 2018). We compare to their DCA model on the NYT corpus.

7 Results

Table 1 shows our main results on the CNN-DM corpus, with abstractive models shown in the top, and bottom-up attention methods at the bottom. We first observe that using a coverage inference penalty scores the same as a full coverage mechanism, without requiring any additional model parameters or model fine-tuning. The results with the CopyTransformer and coverage penalty indicate a slight improvement across all three scores, but we observe no significant difference between Pointer-Generator and CopyTransformer with bottom-up attention.

We found that none of our end-to-end models lead to improvements, indicating that it is difficult to apply the masking during training without hurting the training process. The *Mask Only* model with increased supervision on the copy mechanism performs very similar to the *Multi-Task* model. On the other hand, bottom-up attention leads to a major improvement across all three scores. While we would expect better content selection to primarily improve ROUGE-1, the fact all three increase hints that the fluency is not being hurt specifically. Our cross-entropy trained ap-

Method	R-1	R-2	R-L
ML*	44.26	27.43	40.41
ML+RL*	47.03	30.72	**43.10**
DCA†	**48.08**	31.19	42.33
Point.Gen. + Coverage Pen.	45.13	30.13	39.67
Bottom-Up Summarization	47.38	**31.23**	41.81

Table 2: Results on the NYT corpus, where we compare to RL trained models. * marks models and results by Paulus et al. (2017), and † results by Celikyilmaz et al. (2018).

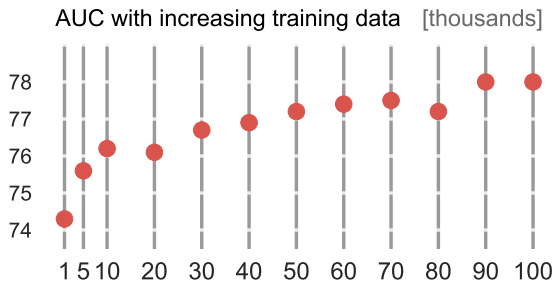

Figure 3: The AUC of the content selector trained on CNN-DM with different training set sizes ranging from 1,000 to 100,000 data points.

proach even outperforms all of the reinforcement-learning based approaches in ROUGE-1 and 2, while the highest reported ROUGE-L score by Chen and Bansal (2018) falls within the 95% confidence interval of our results.

Table 2 shows experiments with the same systems on the NYT corpus. We see that the 2 point improvement compared to the baseline Pointer-Generator maximum-likelihood approach carries over to this dataset. Here, the model outperforms the RL based model by Paulus et al. (2017) in ROUGE-1 and 2, but not L, and is comparable to the results of (Celikyilmaz et al., 2018) except for ROUGE-L. The same can be observed when comparing ML and our Pointer-Generator. We suspect that a difference in summary lengths due to our inference parameter choices leads to this difference, but did not have access to their models or summaries to investigate this claim. This shows that a bottom-up approach achieves competitive results even to models that are trained on summary-specific objectives.

The main benefit of bottom-up summarization seems to be from the reduction of mistakenly copied words. With the best Pointer-Generator models, the precision of copied words is 50.0% compared to the reference. This precision increases to 52.8%, which mostly drives the increase in R1. An independent-samples t-test shows that this improvement is statistically significant with t=14.7 ($p < 10^{-5}$). We also observe a decrease in average sentence length of summaries from 13 to 12 words when adding content selection compared to the Pointer-Generator while holding all other inference parameters constant.

Domain Transfer While end-to-end training has become common, there are benefits to a two-step method. Since the content selector only needs to solve a binary tagging problem with pretrained vectors, it performs well even with very limited training data. As shown in Figure 3, with only 1,000 sentences, the model achieves an AUC of over 74. Beyond that size, the AUC of the model increases only slightly with increasing training data.

To further evaluate the content selection, we consider an application to domain transfer. In this experiment, we apply the Pointer-Generator trained on CNN-DM to the NYT corpus. In addition, we train three content selectors on 1, 10, and 100 thousand sentences of the NYT set, and use these in the bottom-up summarization. The results, shown in Table 3, demonstrates that even a model trained on the smallest subset leads to an improvement of almost 5 points over the model without bottom-up attention. This improvement increases with the larger subsets to up to 7 points. While this approach does not reach a comparable performance to models trained directly on the NYT dataset, it still represents a significant increase over the not-augmented CNN-DM model and produces summaries that are quite readable. We show two example summaries in Appendix A. This technique could be used for low-resource domains and for problems with limited data availability.

8 Analysis and Discussion

Extractive Summary by Content Selection? Given that the content selector is effective in conjunction with the abstractive model, it is interesting to know whether it has learned an effective extractive summarization system on its own. Table 4 shows experiments comparing content selection to extractive baselines. The LEAD-3 baseline is a commonly used baseline in news summarization that extracts the first three sentences from an

	AUC	R-1	R-2	R-L
CNNDM		25.63	11.40	20.55
+1k	80.7	30.62	16.10	25.32
+10k	83.6	32.07	17.60	26.75
+100k	86.6	33.11	18.57	27.69

Table 3: Results of the domain transfer experiment. AUC numbers are shown for content selectors. ROUGE scores represent an abstractive model trained on CNN-DM and evaluated on NYT, with additional copy constraints trained on 1/10/100k training examples of the NYT corpus.

Method	R-1	R-2	R-L
LEAD-3	40.1	17.5	36.3
NEUSUM (Zhou et al., 2018)	41.6	19.0	38.0
Top-3 sents (Cont. Select.)	40.7	**18.0**	37.0
Oracle Phrase-Selector	67.2	37.8	58.2
Content Selector	**42.0**	15.9	**37.3**

Table 4: Results of extractive approaches on the CNN-DM dataset. The first section shows sentence-extractive scores. The second section first shows an oracle score if the content selector selected all the correct words according to our matching heuristic. Finally, we show results when the Content Selector extracts all phrases above a selection probability threshold.

article. Top-3 shows the performance when we extract the top three sentences by average copy probability from the selector. Interestingly, with this method, only 7.1% of the top three sentences are not within the first three, further reinforcing the strength of the LEAD-3 baseline. Our naive sentence-extractor performs slightly worse than the highest reported extractive score by Zhou et al. (2018) that is specifically trained to score combinations of sentences. The final entry shows the performance when all the words above a threshold are extracted such that the resulting summaries are approximately the length of reference summaries. The oracle score represents the results if our model had a perfect accuracy, and shows that the content selector, while yielding competitive results, has room for further improvements in future work.

This result shows that the model is quite effective at finding important words (ROUGE-1) but less effective at chaining them together (ROUGE-2). Similar to Paulus et al. (2017), we find that the decrease in ROUGE-2 indicates a lack of fluency and grammaticality of the generated summaries. A typical example looks like this:

> a man food his first hamburger wrongfully for 36 years. michael hanline, 69, was convicted of murder for the shooting of truck driver jt mcgarry in 1980 on judge charges.

This particular ungrammatical example has a ROUGE-1 of 29.3. This further highlights the benefit of the combined approach where bottom-up predictions are chained together fluently by the abstractive system. However, we also note that the abstractive system requires access to the full source document. Distillation experiments in which we tried to use the output of the content-selection as training-input to abstractive models showed a drastic decrease in model performance.

Data	%Novel	Verb	Noun	Adj
Reference	14.8	30.9	35.5	12.3
Vanilla S2S	6.6	14.5	19.7	5.1
Pointer-Generator	2.2	25.7	39.3	13.9
Bottom-Up Attention	0.5	53.3	24.8	6.5

Table 5: %Novel shows the percentage of words in a summary that are not in the source document. The last three columns show the part-of-speech tag distribution of the novel words in generated summaries.

Analysis of Copying While Pointer-Generator models have the ability to abstract in summary, the use of a copy mechanism causes the summaries to be mostly extractive. Table 5 shows that with copying the percentage of generated words that are not in the source document decreases from 6.6% to 2.2%, while reference summaries are much more abstractive with 14.8% novel words. Bottom-up attention leads to a further reduction to only a half percent. However, since generated summaries are typically not longer than 40-50 words, the difference between an abstractive system with and without bottom-up attention is less than one novel word per summary. This shows that the benefit of abstractive models has been less in their ability to produce better paraphrasing but more in the ability to create fluent summaries from a mostly extractive process.

Table 5 also shows the part-of-speech-tags of the novel generated words, and we can observe an interesting effect. Application of bottom-up attention leads to a sharp decrease in novel adjectives

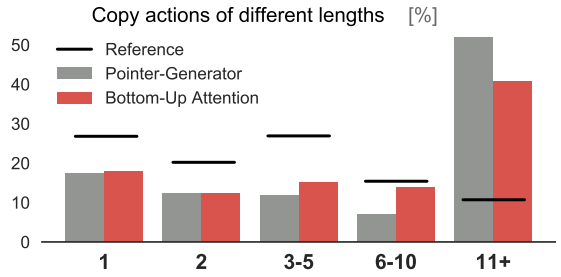

Figure 4: For all copied words, we show the distribution over the length of copied phrases they are part of. The black lines indicate the reference summaries, and the bars the summaries with and without bottom-up attention.

and nouns, whereas the fraction of novel words that are verbs sharply increases. When looking at the novel verbs that are being generated, we notice a very high percentage of tense or number changes, indicated by variation of the word "say", for example "said" or "says", while novel nouns are mostly morphological variants of words in the source.

Figure 4 shows the length of the phrases that are being copied. While most copied phrases in the reference summaries are in groups of 1 to 5 words, the Pointer-Generator copies many very long sequences and full sentences of over 11 words. Since the content selection mask interrupts most long copy sequences, the model has to either generate the unselected words using only the generation probability or use a different word instead. While we observed both cases quite frequently in generated summaries, the fraction of very long copied phrases decreases. However, either with or without bottom-up attention, the distribution of the length of copied phrases is still quite different from the reference.

Inference Penalty Analysis We next analyze the effect of the inference-time loss functions. Table 6 presents the marginal improvements over the simple Pointer-Generator when adding one penalty at a time. We observe that all three penalties improve all three scores, even when added on top of the other two. This further indicates that the unmodified Pointer-Generator model has already learned an appropriate representation of the abstractive summarization problem, but is limited by its ineffective content selection and inference methods.

Data	R-1	R-2	R-L
Pointer Generator	36.3	16.2	33.4
+ Length Penalty	38.0	16.8	35.0
+ Coverage Penalty	38.9	17.2	35.9
+ Trigram Repeat	39.1	17.4	36.1

Table 6: Results on CNN-DM when adding one inference penalty at a time.

9 Conclusion

This work presents a simple but accurate content selection model for summarization that identifies phrases within a document that are likely included in its summary. We showed that this content selector can be used for a bottom-up attention that restricts the ability of abstractive summarizers to copy words from the source. The combined bottom-up summarization system leads to improvements in ROUGE scores of over two points on both the CNN-DM and NYT corpora. A comparison to end-to-end trained methods showed that this particular problem cannot be easily solved with a single model, but instead requires fine-tuned inference restrictions. Finally, we showed that this technique, due to its data-efficiency, can be used to adjust a trained model with few data points, making it easy to transfer to a new domain. Preliminary work that investigates similar bottom-up approaches in other domains that require a content selection, such as grammar correction, or data-to-text generation, have shown some promise and will be investigated in future work.

Acknowledgements

We would like to thank Barbara J. Grosz for helpful discussions and feedback on early stages of this work. We further thank the three anonymous reviewers. This work was supported by a Samsung Research Award. YD was funded in part by a Bloomberg Research Award. SG was funded in part by NIH grant 5R01CA204585-02.

References

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2017. Bottom-up and top-down attention for image captioning and vqa. *arXiv preprint arXiv:1707.07998*.

Valerie Anderson and Suzanne Hidi. 1988. Teach-

ing students to summarize. *Educational leadership*, 46(4):26–28.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Duy Duc An Bui, Guilherme Del Fiol, John F Hurdle, and Siddhartha Jonnalagadda. 2016. Extractive text summarization system to aid data extraction from full text in systematic review development. *Journal of biomedical informatics*, 64:265–272.

Timothy J Buschman and Earl K Miller. 2007. Top-down versus bottom-up control of attention in the prefrontal and posterior parietal cortices. *science*, 315(5820):1860–1862.

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)*, volume 1, pages 1662–1675.

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

Jianpeng Cheng, Li Dong, and Mirella Lapata. 2016. Long short-term memory-networks for machine reading. *arXiv preprint arXiv:1601.06733*.

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

Sumit Chopra, Michael Auli, and Alexander M Rush. 2016. Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 93–98.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, volume 2, pages 615–621.

Alexander Dlikman and Mark Last. 2016. Using machine learning methods and linguistic features in single-document extractive summarization. In *DMNLP@ PKDD/ECML*, pages 1–8.

Bonnie Dorr, David Zajic, and Richard Schwartz. 2003. Hedge trimmer: A parse-and-trim approach to headline generation. In *Proceedings of the HLT-NAACL 03 on Text summarization workshop-Volume 5*, pages 1–8. Association for Computational Linguistics.

Greg Durrett, Taylor Berg-Kirkpatrick, and Dan Klein. 2016. Learning-based single-document summarization with compression and anaphoricity constraints. *arXiv preprint arXiv:1603.08887*.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. Allennlp: A deep semantic natural language processing platform. *arXiv preprint arXiv:1803.07640*.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. *arXiv preprint arXiv:1603.06393*.

Karl Moritz Hermann, Tomas Kocisky, 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*, pages 1693–1701.

Wan-Ting Hsu, Chieh-Kai Lin, Ming-Ying Lee, Kerui Min, Jing Tang, and Min Sun. 2018. A unified model for extractive and abstractive summarization using inconsistency loss. *arXiv preprint arXiv:1805.06266*.

Hongyan Jing and Kathleen R McKeown. 1999. The decomposition of human-written summary sentences. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 129–136.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. 2016. Character-aware neural language models. In *AAAI*, pages 2741–2749.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. *arXiv preprint arXiv:1701.02810*.

Chenliang Li, Weiran Xu, Si Li, and Sheng Gao. 2018a. Guiding generation for abstractive text summarization based on key information guide network. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, volume 2, pages 55–60.

Piji Li, Lidong Bing, and Wai Lam. 2018b. Actor-critic based training framework for abstractive summarization. *arXiv preprint arXiv:1803.11070*.

Peter J Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. 2018. Generating wikipedia by summarizing long sequences. *arXiv preprint arXiv:1801.10198*.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *AAAI*, pages 3075–3081.

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

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

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, Volume 2 (Short Papers)*, volume 2, pages 646–653.

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

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

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

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685*.

Evan Sandhaus. 2008. The new york times annotated corpus. *Linguistic Data Consortium, Philadelphia*, 6(12):e26752.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.

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

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)*, volume 1, pages 1171–1181.

Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Modeling coverage for neural machine translation. *arXiv preprint arXiv:1601.04811*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in Neural Information Processing Systems*, pages 2692–2700.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Wenyuan Zeng, Wenjie Luo, Sanja Fidler, and Raquel Urtasun. 2016. Efficient summarization with read-again and copy mechanism. *arXiv preprint arXiv:1611.03382*.

Qingyu Zhou, Nan Yang, Furu Wei, Shaohan Huang, Ming Zhou, and Tiejun Zhao. 2018. Neural document summarization by jointly learning to score and select sentences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 654–663.

Examples	Generated summary
Reference	green bay packers successful season is largely due to quarterback brett favre
S2S	ahman green rushed for 000 yards in 00-00 victory over the giants . true , dorsey levens , good enough to start for most teams but now green 's backup , contributed kickoff returns of 00 , 00 and 00 yards .
Content Selection	playoff-bound green bay packers beat the giants in the 00-00 victory . the packers won three games and six of each other .
Reference	paul byers , pioneer of visual anthropology , dies at age 00
S2S	paul byers , an early practitioner of mead , died on dec. 00 at his home in manhattan . he enlisted in the navy , which trained him as a cryptanalyst and stationed him in australia .
Content Selection	paul byers , an early practitioner of anthropology , pioneered with margaret mead .

Table 7: Domain-transfer examples.

A Domain Transfer Examples

We present two generated summaries for the CNN-DM to NYT domain transfer experiment in Table 7. S2S refers to a Pointer-Generator with Coverage Penalty trained on CNN-DM that scores 20.6 ROUGE-L on the NYT dataset. The content-selection improves this to 27.7 ROUGE-L without any fine-tuning of the S2S model.