CATS: Customizable Abstractive Topic-based Summarization

Anonymous EMNLP-IJCNLP submission

Abstract

Neural sequence-to-sequence models are a recently proposed family of approaches used in abstractive summarization of text documents, useful for producing condensed versions of source text narratives without being restricted to using only words from the original text. Despite the advances in abstractive summarization, custom generation of summaries (*e.g.* towards a user's preference) remains unexplored. In this paper, we present CATS, an abstractive neural summarization model, that summarizes content in a sequence-to-sequence fashion but also introduces a new mechanism to control the underlying latent topic distribution of the produced summaries. Our experimental results on the well-known CNN/DailyMail dataset show that our model achieves state-of-the-art performance.

1 Introduction

Automatic document summarization is defined as producing a shorter, yet semantically highly related, version of a source document. Solutions to this task are typically classified into two categories: Extractive summarization and abstractive summarization.

Extractive summarization refers to methods that select sentences of a source text based on a scoring scheme, and eventually combine those exact sentences in order to produce a summary. Conversely, abstractive summarization aims at producing shortened versions of a source document by *generating* sentences that do not necessarily appear in the original text. Recent advances in neural sequence-to-sequence modeling have sparked interest in abstractive summarization due to its flexibility and broad range of applications.

The majority of research on text summarization thus far has been focused on extractive summarization (Nallapati et al., 2017), due its simplicity compared to abstractive methods.

Beyond providing a generic summary of a longer passage of text, a system which would allow selective summarization based on a user's preference of topic would be of great value in an array of domains. For example, in the field of information retrieval, it could be used to summarize the results of a user search based on the content of the query.

Summarization is also extensively used in other domains such as concisely describing the gist of news articles and stories (Tombros and Sanderson, 1998; See et al., 2017), supporting the minute-taking process (Shang et al., 2018) in corporate meetings and in the electronic health record domain (Galkó and Eickhoff, 2018), to name a few.

In this paper, we introduce CATS, a customizable abstractive topic-based sequence-to-sequence summarization model, which is not only capable of summarizing text documents with an improved performance as compared to the state of the art, but also allows to selectively focus on a range of desired topics of interest when generating summaries. Our experiments corroborate that our model can selectively add or remove certain topics from the summary. Furthermore, our experimental results on a publicly available dataset indicate that the proposed neural sequence-to-sequence model can effectively outperform state-of-the-art baselines in terms of ROUGE.

The main contributions of this paper are:
(1) We introduce a novel neural sequence-to-sequence model based on an encoder-decoder architecture that outperforms the state-of-the-art baselines in the task of abstractive summarization on a benchmark dataset.
(2) We show how the attention mechanism (Bahdanau et al., 2014) may be used for simultaneously identifying important topics as well as recognizing those parts of the encoder output that are vital to be focused on.

The remainder of this paper is organized is as follows: Section 2 discusses related work on abstractive neural summarization. In Section 3, we introduce the CATS summarization model. In Section 4, we discuss our experimental setup and results comparing CATS to a broad range of competitive state-of-the-art baselines. Finally, in Section 5, we conclude this paper and present future directions of inquiry.

2 Related Work

Recent work approaches abstractive summarization as a sequence-to-sequence problem. One of the early deep learning architectures that was shown to be effective in the task of abstractive summarization was the Attention-based Encoder-Decoder (Nallapati et al., 2016) proposed by Bahdanau et al. (Bahdanau et al., 2014). This model had originally been designed for machine translation problems, where it defined the state of the art.

Attention mechanisms are shown to enhance the basic encoder-decoder model (Bahdanau et al., 2014). The main bottleneck of the basic encoder-decoder architecture is its fixed-sized representation ("thought vector"), which is unable to capture all the relevant information of the input sequence as the model or input scaled up. However, the attention mechanism relies on the notion that at each generation step, only parts of the input are relevant. In this paper, we build on the same notion to force our proposed model to attend to parts of the input which together represent a semantic topic.

Based on the Attention-based encoder-decoder architecture, several models were introduced. The Pointer Generator Network (PGN) (Vinyals et al., 2015) was applied by See et al. (See et al., 2017) to the task of abstractive summarization. This model aims at solving the challenge of out-of-vocabulary words and factual errors. The main idea behind this model is to choose between either generating a word from the fixed vocabulary or copying one from the source document at each step of the generation process. It incorporates the power of extractive methods by "pointing" (Vinyals et al., 2015). At each step, a generation probability is computed, which is used as a switch to choose words from the target vocabulary or the source document. Our model differs from the PGN firstly in the use of a different attention mechanism which forces the model to focus on certain topics when generating an output summary. Secondly, our model enables the selective inclusion or exclusion of certain topics in a generated summary, which can have several potential applications. This is done by incorporating information from an unsupervised topic model. By definition, topic models are hierarchical Bayesian models of discrete data, where each topic is a set of words, drawn from a fixed vocabulary, which together represent a high-level concept (Wang et al., 2008). According to this definition, Blei et al. introduced the Latent Dirichlet Allocation (LDA) (Blei et al., 2003) topic model. We further elaborate on the connection between this and our model in Section 3.

The work of (Paulus et al., 2017) is another approach which utilizes reinforcement learning to optimize ROUGE L, such that sub-sequences similar to a reference summary are generated. Similar to (See et al., 2017) they also use the pointer generator mechanism to switch between generating a token or extracting it from the source.

(Gehrmann et al., 2018) propose using a content selector to select phrases in a source document that should be part of a generated summary. Likewise, (Li et al., 2018) introduce an information selection layer to explicitly model the information selection process in abstractive document summarization. They perform information filtering and local sentence selection in order to generate summaries. The two latter approaches report best performances on the CNN/DailyMail benchmark. Our proposed model relies on information selection in the form of topics.

Existing neural models do not directly take advantage of the latent topic structure underlying input texts. To the best of our knowledge, this paper is the first work to include this source of information explicitly in a neural abstractive summarization model. The experimental section will demonstrate the merit of this approach empirically.

3 Proposed Model: CATS

3.1 Model Overview

Our abstractive summarization scheme CATS is a neural sequence-to-sequence model based on the attention encoder-decoder architecture (Nallapati et al., 2016). Additionally, we incorporate the concept of pointer networks (Vinyals et al., 2015) into our model, which enables copying words from the encoder output while also being able to generate words from a fixed vocabulary. Furthermore, we

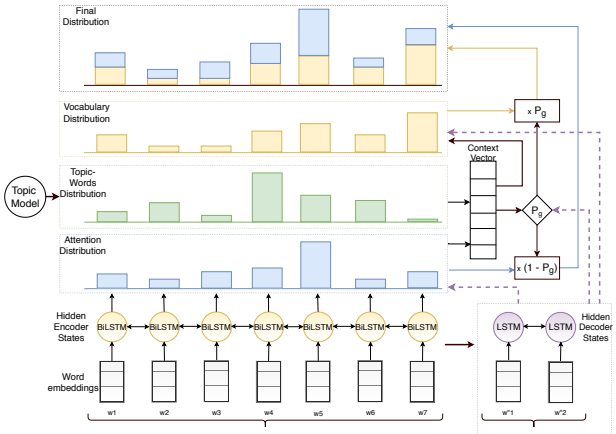

Figure 1: The architecture of our proposed model

introduce a novel attention mechanism controlled by an unsupervised topic model. This ameliorates attention by way of focusing not only on those words which it learns as important for producing a summary (as in the standard attention mechanism), but also by learning the topically important words in a certain context. We refer to this novel mechanism as topical attention. Over the encoder-decoder training steps, the model parameters adapt in a way to learn the topics of each document. During testing, when the model decoder generates summaries of test documents, it therefore no longer requires the input information from the topic model, as it learns a generalized pattern of the word weights under each topic.

We depict our model in Figure 1. In the following we describe the various components of our model.

3.2 Encoder & Decoder

The tokens of a document (i.e. extracted by a document tokenizer) are given one-by-one as input to the encoder layer. Our encoder is a single-layer Bi-directional Long Short Term Memory (BiLSTM) network (Graves and Schmidhuber, 2005). The network outputs a sequence of encoder hidden states h_i, each state being a concatenation of forward and backward hidden states, as in (Bahdanau et al., 2014).

At each decoding time step t, the decoder receives as input x_t the word embedding of the previous word (while training, this is the previous word of the reference summary and at test time it is the previous word output by the decoder) and computes a decoder state s_t. Our decoder is a single-layer Long Short Term Memory (LSTM) network (Greff et al., 2017).

3.3 Topical Attention

We propose the topical attention distribution a^t to be calculated as a combination of the usual attention weights as in (Bahdanau et al., 2014) and a "topical word vector" derived from a topic model. We use LDA (Blei et al., 2003) as the topic model of choice. Besides the experimentally shown robust performance (Blei et al., 2003), an important reason for selecting LDA over other topic models is that words under this model are always assigned probabilities between 0 and 1 and the sum of the probability scores of all words in each topic is 1. This facilitates the fusion of these scores with attention weights, which are then fed to a softmax function without the need for additional normalization steps.

In order to compute the topical attention weights, after training an LDA model using the training data, we map the target summary corresponding to each document to its LDA space. This gives us the strength of each topic in each target summary. Furthermore, since for each topic we also have the probability scores of each word in a fixed vocabulary $\mathcal{V}$, for a given document d we could calculate a *topical word vector* τ^d of dimension $|\mathcal{V}|$ considering all the words in that document, such that:

$$\tau^d = \sum_i P(topic_i|d) \cdot \tilde{\mathbf{w}}_i \quad (1)$$

where $P(topic_i|d)$ is the probability of each LDA topic being present in the target summary, and $\tilde{\mathbf{w}}_i$ is the $|\mathcal{V}|$-dimensional vector of probabilities $\tilde{w}_j = P(word_j|topic_i)$ of all words in vocabulary $\mathcal{V}$ under $topic_i$.

Then, for an input sequence of length K, we compute the final attention vector $a^t \in R^K$ at decoding step t as:

$$e_k^t = v^T \tanh(W_h h_k + W_s s_t + b_{attn}) \quad (2)$$

$$a^t = f(e^t, \tau^d) \quad (3)$$

where $e^t \in R^K$ is a precursor attention vector, $h_k \in R^n$ represents the k-th encoder hidden state and $s_t \in R^l$ the decoder state at decoding step t, while $v \in R^m$, $W_h \in R^{m \times n}$, $W_s \in R^{m \times l}$, $b_{attn} \in R^m$ are learnable parameters. Function f combines the topical word vector with the precursor attention vector. In order to combine the two,

we define f as the following distribution over the input sequence:

$$a^t = \frac{softmax(e^t) + softmax(\tilde{\tau}^d)}{2} \quad (4)$$

where $\tilde{\tau}^d \in R^K$ denotes the "reduced" topical word vector which is formed by selecting the K components of $\tau^d \in R^{|\mathcal{V}|}$ corresponding to the K words of the input sequence.

The attention distribution can be viewed as a probability distribution over the words from the source document, which tells the decoder where to look to produce the next word. Subsequently, the attention distribution is used to produce a weighted sum of the encoder hidden states, known as the context vector $h_t^* \in R^n$, as follows:

$$h_t^* = \sum_k a_k^t \cdot h_k \quad (5)$$

The context vector, which is a fixed-sized representation of what has been read by the encoder at this step, is concatenated with the decoder state s_t and the result is linearly transformed and passed through a softmax function to produce the final output distribution $P_{\mathcal{V}}(w)$ over all words w in vocabulary $\mathcal{V}$:

$$P_{\mathcal{V}} = softmax(V[s_t, h_t^*] + b) \quad (6)$$

where $V \in R^{|\mathcal{V}| \times (n+l)}$ and $b \in R^{|\mathcal{V}|}$ are learnable parameters.

3.4 Pointer Generator

We utilize the concept of pointer generators in our model, in order to give our model the flexibility of choosing between generating a word from a fixed vocabulary or copying it directly from source when needed.

We define p_g as a generation probability such that $p_g \in [0, 1]$. We calculate p_g for time step t from the context vector h_t^*, the decoder state s_t and the decoder input x_t as:

$$p_g = \sigma(w_{h^*}^T h_t^* + w_s^T s_t + w_x^T x_t + b_{pt}) \quad (7)$$

where vectors w_{h^*}, w_s, w_x, and scalar value b_{pt} are learnable parameters and σ is a sigmoid function.

Subsequently, p_g is used to linearly interpolate between copying a word from the source (specifically, to copy from the source document we sample over the input words using the attention distribution) and generating it from the fixed vocabulary using $P_{\mathcal{V}}$.

For each document, we define the union of the fixed vocabulary $\mathcal{V}$ and all words appearing in the source document as the "extended vocabulary". Using the linear interpolation described above, the probability distribution over the extended vocabulary is:

$$P(w) = p_g P_{\mathcal{V}}(w) + (1 - p_g) \sum_{\forall i: w_i = w} a_i^t \quad (8)$$

In Equation 8, we note that if a word w would be out-of-vocabulary, then $P_{\mathcal{V}}(w)$ would be equal to zero. Analogously, if w does not appear in the source document, then $\sum_{\forall i: w_i = w} a_i^t$ would be equal to zero. In expectation, the most likely words under this new distribution are the ones that both receive a high likelihood under the output distribution of the decoder, as well as much attention by the attention module. Words with a high likelihood under the initial output distribution, which however receive little to no attention, will be generated with a reduced probability, while words receiving much attention, even if they receive a low likelihood by the decoder or do not even exist in the vocabulary $\mathcal{V}$, will be generated with an increased probability.

Therefore, by being able to switch between out-of-vocabulary words and the words from the vocabulary, the pointer generator model mitigates the problem of factual errors or the lack of sufficient vocabulary in the output summary.

3.5 Coverage Mechanism

The coverage mechanism (Tu et al., 2016) is a method for keeping track of the level of attention given to each word at all time steps. In other words, by summing the attention at all previous steps, the model keeps track of how much coverage each encoding has already received.

This mechanism alleviates the repetition problem, which is a very common issue in recurrent neural networks with attention.

We follow (Xu et al., 2015) and define the *coverage vector* $c^t \in R^K$ simply as the sum of atten-

tion vectors at all previous decoding steps:

$$c^t = \sum_{i=0}^{t-1} a^i \tag{9}$$

First, the coverage vector is taken into account when calculating the attention vector by adding an extra term and modifying Equation 2 as follows:

$$e_k^t = v^T \tanh(W_h h_k + W_s s_t + c_k^t \cdot w_c + b_{attn}) \tag{10}$$

where $w_c \in R^m$ is a learnable parameter vector of the same length as v.

Second, following (See et al., 2017), we use the coverage vector to introduce an additional loss term, which is added to the original negative log-likelihood loss after being weighted by hyperparameter λ, to produce the following total loss at decoding step t :

$$\mathcal{L}_t = -\log P(w_t) + \lambda \sum_{i=0}^{k} \min(a_i^t, c_i^t) \tag{11}$$

This additional loss term encourages the attention module to redistribute attention weights by placing low weights to input words which have already received much attention throughout previous decoding steps. The overall loss for the entire output sequence of length T is the average loss over all T decoding steps.

3.6 Decoding

In order to generate the output summaries we use beam search. During evaluation of the model using the test data, contrary to training, we do not provide the model with any topical information from our trained LDA topic model. We believe that during training, the model parameters learn to best take advantage of the provided topical attention distribution, implicitly learning patterns of topic-words weights.

4 Evaluation

4.1 Dataset

We use the CNN/DailyMail dataset (Hermann et al., 2015; Nallapati et al., 2016), which contains news articles from the *CNN* and *Daily Mail* websites. The experiments reported in this paper are based on the non-anonymized version of the dataset, containing 287,226 pairs of training articles and reference summaries, 13,368 validation pairs, and 11,490 test pairs. On average, each document in the dataset contains 781 tokens paired with multi-sentence summaries (56 tokens spread over 3.75 sentences).

Similar to (Nallapati et al., 2016; See et al., 2017), we use a range of pre-processing scripts to prepare the data. This includes the use of the *Stanford CoreNLP* tokenizer to break down documents into tokens. For greater transparency and reproducibility of our results, we make all pre-processing scripts available together with our code base.

4.2 Baseline Models

We empirically compare CATS with several abstractive baselines as follows:

- *Attention-based encoder-decoder* (Nallapati et al., 2016).
- *PGN and PGN+Coverage* (See et al., 2017).
- *RL with Intra-Attention* (Paulus et al., 2017).
- *BottomUpSum* (Gehrmann et al., 2018).
- *InformationSelection* (Li et al., 2018).
- *ML+RL ROUGE+Novel, with LM* (Kryscinski et al., 2018).
- *UnifiedAbsExt* (Hsu et al., 2018).
- *RNN-EXT + ABS + RL + Rerank* (Chen and Bansal, 2018).

4.3 Evaluation Metrics

We evaluate our proposed model against the baseline methods in terms of $F_1\, ROUGE\, 1$, $F_1\, ROUGE\, 2$, and $F_1\, ROUGE\, L$ scores using the official Perl-based implementation of ROUGE (Lin, 2004), following common practice.

4.4 Experimental Results

We specify our model parameters as follows: the hidden state dimension of RNNs is set to 256, the embedding dimension of the word embeddings is set to 128, and the mini-batch size is set to 16. Furthermore, the maximum number of encoder steps is set to 400 and the maximum number of decoder steps is set to 100. In decoding mode (i.e. generating summaries on the test data) the beam search size is 4 and the minimum decoder size which determines the minimum length of a generated summary is set to 35. Finally, the size of the vocabulary that the models use is set to 50,000 tokens.

To train a topic model we run LDA over the training data. LDA returns M lists of keywords representing the latent topics discussed in the collection. Since the actual number of underlying

topics (M) is an unknown variable in the LDA model, it is important to estimate it. For this purpose, similar to the method proposed in (Griffiths and Steyvers, 2004; Bahrainian and Crestani, 2018), we went through a model selection process. It involves keeping the LDA parameters (commonly known as α and η) fixed, while assigning several values to M and running the LDA model for each value. We picked the model that minimizes $\log P(W|M)$, where W contains all the words in the vocabulary. This process is repeated until we have an optimal number of topics. The training of each LDA model takes nearly a day, so we could only repeat it for a limited number of M values. In particular, we trained the LDA model with values M ranging from 50 up to 500 with an increment of 50, and the optimal value on the CNN/Dailymail dataset was found to be 100.

Based on the setup described above, in the following present our experiments for evaluating our model.

4.4.1 Experiment comparing all models in terms of ROUGE

We first compare our proposed models against all baselines in terms of the F_1 ROUGE metrics presented in Section 4.3. The results of this comparison are given in Table 1.

As we observe in Table 1, our model with coverage outperforms all other models in terms of ROUGE 1. In order to verify the significance of the difference we conduct a statistical significance test based on the bootstrap re-sampling technique using the official ROUGE package (Lin, 2004). In the case of ROUGE 2 we achieve state-of-the-art performance in a tie with the 'BottomUpSum' approach of (Gehrmann et al., 2018). In the case of ROUGE L, (Paulus et al., 2017) reports the highest performance; however, this is due to their model loss function optimizing directly on the evaluation metric ROUGE L instead of the summarization loss. In fact, (Hsu et al., 2018) reports an experiment that shows summaries generated by the (Paulus et al., 2017) method achieve poorest readability scores as compared with a number of models including PGN and their own UnifiedAbsExt model, a finding which we also confirmed by comparing them with the output of our model (see Section 4.4.2). We note that we did not include the method of (Celikyilmaz et al., 2018) in our comparison, due to the fact that unlike most papers that use preprocessing scripts of (See et al., 2017) for the non-anonymized version of the dataset, they use different scripts. The effect of this difference on their LEAD-3 baseline remains unclear as they do not report it. Thus, their results may not be necessarily comparable with ours.

4.4.2 Human Evaluation of Summaries

We conduct a human evaluation in order to assess the quality of summaries produced by CATS+coverage in comparison with that of PGN+coverage (See et al., 2017) and summaries of RL with Intra-Attention (Paulus et al., 2017) provided by them, in terms of informativeness and readability of 50 randomly chosen summaries by the three models. By comparing the output produced by the three models, the three human assessors[1] assigned scores ranging from 1 to 5 to each summary, while blinded to the identity of the models. The average overall scores of each model are shown in Table 2.

Table 2: Human evaluation comparing quality of summaries on a 1-5 scale using three evaluator.

	Readability	Informativeness
CATS	**4.1**	**3.9**
PGN	3.5	3.3
RL+Intra-Attention	2.6	2.9

We observe that the summaries generated by our model are judged to be more readable and more informative.

4.4.3 Human Evaluation of Customizing Summaries

In this section, we report a human evaluation of CATS's capability to include only certain topics in a summary and exclude others. As mentioned earlier, CATS is the first neural abstractive summarization model that allows its users to selectively include or exclude latent topics from their output summaries. In order to demonstrate this feature, we remove a few topics from the output of the topic model, fine-tune the trained summarization model for a few additional training steps and analyze the effect. Our expectation is that the focus of certain output summaries which should usually contain those topics will change, while naturally the ROUGE values will decrease. For this experiment, we chose two topics and removed them from the summaries one at a time. The first topic is related to *health-care* and its top five keywords are "dr", "medical", "patients", "health",

[1]None of the assessors are affiliated with this paper.

Table 1: Results of a comparison between our proposed models against the baselines in terms of F_1 ROUGE metrics on the CNN/Dailymail dataset. Statistical significance test was done with a confidence of 95%. '*' means that results are based on the anonymized version of the dataset and not strictly comparable to our results.

Models	ROUGE 1 (%)	ROUGE 2 (%)	ROUGE L (%)
CATS (Ours)	38.01	16.35	34.87
CATS+coverage (Ours)	**41.73**	**18.64**	38.17
LEAD-3 Baseline	40.34	17.70	36.57
Attn. Enc-Dec (Nallapati et al., 2016)	35.46	13.30	32.65
PGN (See et al., 2017)	36.44	15.66	33.42
PGN+coverage (See et al., 2017)	39.53	17.28	36.38
RL with Intra-Attention (Paulus et al., 2017) '*'	41.16	15.75	**39.08**
BottomUpSum (Gehrmann et al., 2018)	41.22	**18.68**	38.34
InformationSelection (Li et al., 2018)	41.54	18.18	36.47
ML+RL ROUGE+Novel, with LM (Kryscinski et al., 2018)	40.19	17.38	37.52
UnifiedAbsExt (Hsu et al., 2018)	40.68	17.97	37.13
RNN-EXT + ABS + RL + Rerank (Chen and Bansal, 2018)	40.88	17.80	38.54

and "care". The second topic is related to *police arrests and charges* with its top five words being "charges", "court", "arrested", "allegedly", and "jailed". We randomly selected a total of 50 test documents that originally contained either of the above-mentioned topics. In order to do so we used the LDA model described in the beginning of Section 4.4. Using the LDA rankings of topics of source documents, we randomly chose 50 that contained either-mentioned topics and those topics were not their sole or primary focus but in the second rank. Three human judges evaluated whether the summaries generated by CATS with restricted topics showed exclusion or reduction of those topics or there was no major difference. They were instructed to look for existence of the top 20 words of each topic in particular, except for cases that one of these words is a part of a name (e.g. American Health Center). For each document, we take the majority vote of the human assessors as the final decision. The results of this experiment show that in 44 documents the topics were excluded, in four documents the topics were reduced and in two documents the majority vote showed no major difference.

Table 3 shows an example summary produced by CATS that was restricted not to include the *health-care* topic, next to a summary produced by CATS with no topic restriction as well as the corresponding reference summary. We observe that the focus of the summary is altered such that it focuses on the crime-related aspects rather than health-care in order to avoid using words such as "hospital", "patients" and "medicine".

4.4.4 Analysis of Repetition in Output Summaries

In this experiment we analyze the quality of the output summaries produced by our models and those produced by PGN and PGN+coverage in terms of repetition of text. A common issue with attention-based encoder-decoder architectures is the tendency to repeat an already generated sequence. In text summarization this results in summaries containing repeated sentences or phrases. As described in Section 2, the coverage mechanism is used to reduce this undesirable effect.

Here we compare our two models, CATS and CATS+coverage, to PGN and PGN+coverage in terms of n-grams repetition with n ranging from 1 to 6. For this purpose we train all four models with exact same parameters whenever applicable. The upshot of this experiment is reported in Figure 2. The scores reported in the figure are normalized average repetition scores over all output summary documents in the test set of the CNN/Dailymail dataset. We compute the scores by calculating the average of per-document n-gram repetition score $S_{rep,doc}$ over all test output documents, which is defined as $S_{rep,doc} = \frac{\#duplicate\, n-grams}{\#all\, n-grams}$.

We observe that our models demonstrate lower repetition of text in their output summaries compared with both PGN and PGN+coverage, which is confirmed by manual inspection of the output. This trend is consistent on all the tested n-grams.

We believe that the reason behind this phenomenon is that our model tends to focus not only on the few words in the input sequence which are assigned high attention weights, but also on other words which are topically connected with these words in a certain context. Firstly, this acts as an

Table 3: Comparison of a CATS generated summary next to a summary with restricted topics and the human-written reference summary[2].

CATS restricted with health-care topic	*CATS*	*Reference*
victorino chua , 49 , denies murdering tracey arden , 44 , arnold lancaster , 71 and derek weaver , 83 , and deliberately poisoning 18 others between 2011 and 2012 . chua has pleaded not guilty to 36 charges in all , including three alleged murders , one count of grievous bodily harm with intent , 23 counts of attempted grievous bodily harm with intent , eight counts of attempting to cause a poison to be administered and one count of administering a poison .	victorino chua , 49 , has given evidence for the first time and denied he tampered with saline bags and ampoules at stepping hill hospital in stockport . a nurse today told a jury he did not murder three hospital patients and poison almost 20 more at stepping hill hospital in stockport in order to kill and injure people he was caring for . chua denies murdering patients tracey arden , 44 , arnold lancaster , 71 and derek weaver , 83 , and deliberately poisoning 18 others between 2011 and 2012 .	victorino chua , 49 , denies murdering patients at stockport hospital in 2011 . filipino nurse also accused of poisoning 18 more at stepping hill hospital . denies injecting insulin and other poisons into bags of medicine on ward .

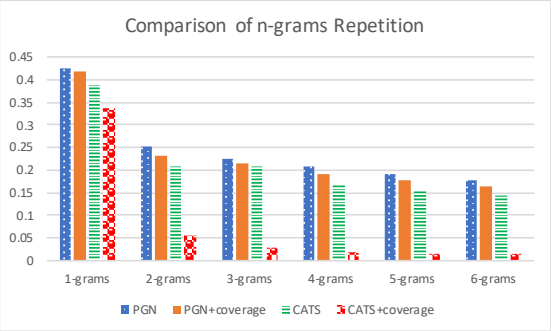

Figure 2: Experiment comparing the degree of n-grams repetition in our models versus that of the PGN and PGN+coverage baselines on the CNN/Dailymail test set. Lower numbers show less repetition in the generated summaries.

attention diversification and redistribution mechanism (an effect similar to coverage). Secondly, these topically connected words receive a higher generation probability (through Equations 6 and 8) and the model is more inclined to paraphrase the input.

The result of this experiment indicates that our topical attention mechanism may be a viable solution to the repetition issue in sequence generation based on encoder-decoder architectures.

5 Conclusions and Future Work

In this paper we present CATS, an abstractive summarization model that makes use of latent topic information in a source document, and is thereby capable of controlling the topics appearing in an output summary of a source document. This can enable customization of generated texts based on user profiles or explicitly given topics, in order to present content tailored to a user's information needs.

Our experimental results show that our CATS+coverage model achieves state-of-the-art performance in terms of standard evaluation metrics for summarization (i.e ROUGE) on an important benchmark dataset, while enabling customization in producing summaries.

CATS can serve as a foundation for future work in the domain of automatic summarization. Based on the results of this paper, we believe the future work on summarization systems to be exciting, in that a generated summary could be customized to users' needs. We envision three ways of controlling the focus of output summaries using our models: First, as demonstrated in the experiment in Section 4.4.3, certain topics could be disabled in the output of the topic model and be consequently discarded from output summaries. Second, a reference document could be provided to the topic model, its topics could be extracted and subsequently direct the focus of generated summaries. This is useful when a user wants to see summaries/updates primarily or only regarding issues discussed in an existing reference document. Third, content extracted from user profiles (e.g. history of web pages of interest) could be provided to the topic model, their salient themes extracted by the model and then taken into account whenever presenting users with summaries. All three directions are interesting future works of this paper.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly

learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Seyed Ali Bahrainian and Fabio Crestani. 2018. Augmentation of human memory: Anticipating topics that continue in the next meeting. In *Proceedings of the 2018 Conference on Human Information Interaction & Retrieval*, CHIIR '18, pages 150–159.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022.

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.

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

Ferenc Galkó and Carsten Eickhoff. 2018. Biomedical question answering via weighted neural network passage retrieval. In *European Conference on Information Retrieval*, pages 523–528. Springer.

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

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5-6):602–610.

K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber. 2017. Lstm: A search space odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, 28(10):2222–2232.

Thomas L Griffiths and Mark Steyvers. 2004. Finding scientific topics. *Proceedings of the National academy of Sciences*.

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. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 132–141.

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

Wei Li, Xinyan Xiao, Yajuan Lyu, and Yuanzhuo Wang. 2018. Improving neural abstractive document summarization with explicit information selection modeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 1787–1796.

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

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Ramesh Nallapati, Bowen Zhou, Cícero Nogueira dos Santos, Çaglar Gülçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016*, pages 280–290.

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

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1073–1083.

Guokan Shang, Wensi Ding, Zekun Zhang, Antoine Jean-Pierre Tixier, Polykarpos Meladianos, Michalis Vazirgiannis, and Jean-Pierre Lorré. 2018. Unsupervised abstractive meeting summarization with multi-sentence compression and budgeted submodular maximization. *arXiv preprint arXiv:1805.05271*.

Anastasios Tombros and Mark Sanderson. 1998. Advantages of query biased summaries in information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 2–10. ACM.

Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Modeling coverage for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1*.

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

Chong Wang, David Blei, and David Heckerman. 2008. Continuous time dynamic topic models. *Proc. of UAI*.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057.