CATS: Customizable Abstractive Topic-based Summarization

SEYED ALI BAHRAINIAN, AI Lab, Brown University, USA
GEORGE ZERVEAS, AI Lab, Brown University, USA
FABIO CRESTANI, Informatics Department, University of Lugano, Switzerland
CARSTEN EICKHOFF, AI Lab, Brown University, USA

Neural sequence-to-sequence models are the state-of-the-art approach used in abstractive summarization of textual 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 while also introducing a new mechanism to control the underlying latent topic distribution of the produced summaries. We empirically illustrate the efficacy of our model in producing customized summaries and present findings that facilitate the design of such systems. We use the well-known CNN/DailyMail dataset to evaluate our model. Furthermore, we present a transfer-learning method and demonstrate the effectiveness of our approach in a low resource setting, *i.e.* abstractive summarization of meetings minutes, where combining the main available meetings' transcripts datasets, AMI and ICSI, results in merely a few hundred training documents.

CCS Concepts: • **Computing methodologies → Neural networks**; *Latent Dirichlet allocation*; **Natural language generation**.

Additional Key Words and Phrases: sequence-to-sequence neural models, abstractive summarization,topical customization [1]

ACM Reference Format:
Seyed Ali Bahrainian, George Zerveas, Fabio Crestani, and Carsten Eickhoff. 2021. CATS: Customizable Abstractive Topic-based Summarization. *ACM Transactions on Information Systems* 1, 1, Article 1 (January 2021), 24 pages. https://doi.org/10.1145/3464299

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 *selects* sentences of a source text based on a scoring scheme, and combines 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. The majority of traditional research on text summarization has focused on extractive summarization [5, 27] due to its simplicity

[1]This article has some textual overlap with the PhD thesis of the first author [3].

Authors' addresses: Seyed Ali Bahrainian, AI Lab, Brown University, Providence, RI, USA; George Zerveas, AI Lab, Brown University, Providence, RI, USA; Fabio Crestani, Informatics Department, University of Lugano, Lugano, Ticino, Switzerland; Carsten Eickhoff, AI Lab, Brown University, Providence, RI, USA.

compared to abstractive methods. Recent advances in neural sequence-to-sequence modeling, however, have sparked interest in abstractive summarization due to its flexibility and broad range of applications.

Summarization is extensively used in domains such as news articles [33, 37], minute-taking in corporate meetings [35] or electronic health records [14], to name a few. Aside from providing generic summaries of passages of text, there are applications to Information Retrieval (IR) scenarios in which the retrieval system summarizes results rather than merely retrieve them. For instance, search engines are increasingly presenting summaries, mash-ups and digests of relevant documents in the form of natural language answers to user queries. Automatic summarization lends itself for key use cases in mobile search [1] and scenarios involving communication with search engines via voice. Previous research on voice-based search shows that merely reading out the textual output of a search engine result page is an insufficient interaction paradigm [32] for a user. Furthermore, the underlying components of a spoken conversational search system (where communication between user and system is mediated verbally through voice) will need to operate differently from a traditional IR system [12, 36]. A recent user study [38] on conversational search has observed the importance of document summarization when presenting results of users' spoken search queries. In fact, the ideal voice-based assistant would summarize the key points of particular relevance for a certain searcher. This paper presents a novel abstractive summarization framework as a first step towards this vision.

In this paper, we introduce CATS, a **C**ustomizable **A**bstractive **T**opic-based sequence-to-sequence **S**ummarization model, which is not only capable of summarizing text documents with high quality, 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 specific topics from the summary. Furthermore, our experimental results on a publicly available dataset indicate that the proposed neural sequence-to-sequence model can be effectively fine-tuned to perform abstractive summarization in a low-resource setting. Moreover, we discuss a number of findings in the process of developing an abstractive summarization model with the ability to customize summaries. The main contributions of this article are:

(1) We introduce a novel neural sequence-to-sequence model based on an encoder-decoder architecture which leverages topic modeling to perform customizable abstractive summarization.
(2) We introduce a novel attention mechanism [2] named *topical attention* that 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.
(3) We extensively evaluate our model in customizing summaries, general abstractive summarization, as well as summarization in low-resource settings.

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 showing the efficacy of CATS in custom generation of summaries. Furthermore, we present a transfer-learning approach to summarization of small size datasets and we conduct a ROUGE-based evaluation. In Section 5, we present a discussion on the potential use cases of CATS, other potential means of custom summary generation, and how the topical attention can be adapted to other sequence-to-sequence problems. Finally, in Section 6, we conclude with a discussion on future directions of inquiry.

2 RELATED WORK

Prior to the rise of neural sequence-to-sequence models there had been limited interest in the area of abstractive summarization. TOPIARY was an abstractive model proposed in 2004 by Zajic et al. [48] which showed superior results in the DUC-2004 task. This model used a combination of linguistically motivated compression techniques and an unsupervised topic detection algorithm that inserts keywords extracted from the article into the compressed output. Some other notable work in the task of abstractive summarization includes using traditional phrase table-based machine translation approaches [7] and compression using weighted tree transformation rules [11].

Recent work approaches abstractive summarization as a sequence-to-sequence problem. In this section, we first briefly review some of the most important research in this domain. In order to do so we divide the literature into two categories of models that are mostly trained from scratch while requiring lower computational resources for training and those models which are based on fine-tuning already existing models that exhibit high computational demand both for training the base models as well as fine-tuning. Then we focus on the use of topic models in previous abstractive summarization research.

2.1 Seq2seq Abstractive Summarization Models Trained from Scratch

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 [28] proposed by Bahdanau et al. [2]. This model had originally been designed for machine translation, where it defined the state of the art.

Attention mechanisms are shown to enhance the basic encoder-decoder model [2]. 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) [41] was applied by See et al. [33] 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" [41]. 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 [42]. According to this definition, Blei et al. introduced the Latent Dirichlet Allocation (LDA) [8] topic model. We further elaborate on the connection between this and our model in Section 3.

The work of [29] is another approach which utilizes reinforcement learning to optimize ROUGE L, such that subsequences similar to a reference summary are generated. Similar to [33] they also use the pointer generator mechanism to switch between generating a token or extracting it from the source.

Gehrmann et al. [15] propose using a content selector to select phrases in a source document that should be part of a generated summary. Likewise, [25] 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.

2.2 Seq2seq Abstractive Summarization Models developed by Fine-tuning Pre-trained Models

The introduction of Transformer architectures and their proven efficacy in various natural language sequence-to-sequence problems is the latest major shift in the automatic document summarization field. Here we briefly review some of the latest developments in the space.

One of the top Transformer-based models is UniLM (Unified Pretrained Language Model)[13] from Microsoft. "The model architecture of UNILM follows that of BERTLARGE" [13]. The GELU [20] activation is used as in the GPT [30] model. They use a 24-layer Transformer with $1,024$-dimensional hidden layers, and 16 attention heads, containing about $340M$ parameters. "UNILM is initialized by BERTLARGE, and then pre-trained using English Wikipedia and the BookCorpus" [13]. Subsequently, this model is fine-tuned using summarization training data.

Another important model in this category is the T5 (Text-to-Text Transfer Transformer) model from Google [31] that uses transfer-learning on the Transformer architecture introduced by Vaswani et al. [40]. The authors study a number of variants of the Transformer architecture and finally fine-tune them on different natural language processing tasks.

The next model that is noteworthy in this domain is BART [24] by Facebook. BART is a denoising autoencoder for pretraining sequence-to-sequence natural language processing models. BART is trained by "corrupting text with an arbitrary noising function, and learning a model to reconstruct the original text" [24]. Similar to the T5 model, BART too is based on the Transformer architecture proposed by Vaswani et al. [40] while using a number of noising approaches, such as token masking, token deletion, randomly shuffling the order of the original sentences and a novel in-filling scheme, where spans of text are replaced with a single mask token. The only major difference to the Transofrmer architechture is that, following GPT, the authors replace ReLU activation functions by GeLUs [20]. They also state that their proposed architecture "is closely related to that used in BERT, with the following differences: (1) each layer of the decoder additionally performs cross-attention over the final hidden layer of the encoder (as in the transformer sequence-to-sequence model); and (2) BERT uses an additional feed-forward network before word prediction, which BART does not" [24]. For text generation tasks such as abstractive summarization, BART is then fine-tuned on in-domain data.

The final model in this category that we review is ProphetNet [47], which currently represents the state-of-the-art in abstractive summarization. This model also utilizes the Transformer architecture [40]. The main difference of ProphetNet is changing the original sequence-to-sequence optimization problem of predicting the next single token into predicting the n next token simultaneously. They show that this approach outperforms all other baselines in abstractive summarization in terms of ROUGE scores.

2.3 Use of Topic Models in Summarization

There has also been previous work utilizing topic information in sequence-to-sequence problems such as neural response generation [45]. The work of Xing et al. uses a topic model named Twitter LDA which is used in responding to messages. Aside from the different objective, this work is different from ours in that firstly, Twitter LDA assumes the existence of only a single topic per document. This assumption may be true for tweet-length texts but will not hold in summarization of longer news articles. Secondly, the topic embeddings are derived from the source document and aggregated in a very different way than ours.

The use of LDA topic information in neural abstractive summarization has been considered by Wang et al. [43]. Our work fundamentally differs from theirs not only in that they use a reinforcement learning approach along with convolutional neural networks optimizing directly on ROUGE, but also that our proposed model learns topic embedding weights at training time and does not use any topic information at test time. Moreover, they use topic embeddings of a source document while we use the topics of a target summary. Additionally, previous research [22] shows that while optimizing on ROUGE naturally results in a high ROUGE score, the readability of summaries produced by such systems can be poor compared with that of methods optimizing summarization losses like the one proposed in this work.

In summary, topic information has been used in previous neural models as an input, and Wang et al. [43] argue that it results in the diversification of words appearing in summaries. However, the novelty of our approach lies in using topic information to systematically influence the output summary and steer the generation mechanism to focus on certain topics only, allowing us to remove or downweight unwanted topics from an output summary. The experimental section empirically demonstrates the merit of this approach, not only for customizing summaries, but also for achieving a high performance in terms of ROUGE scores. More importantly, we demonstrate via a user study that CATS can effectively control the topics present in a generated summary.

3 PROPOSED MODEL: CATS

3.1 Model Overview

Our abstractive summarization method CATS is a neural sequence-to-sequence model based on the attention encoder-decoder architecture [28]. Additionally, we incorporate the concept of pointer networks [41] into our model, which enables copying words from the source side while also being able to generate words from a fixed vocabulary. Furthermore, we 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

Prior to encoding, all documents are pre-processed in the same way as [33] where the Stanford CoreNLP package is used to tokenize sentences.

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 [16]. The network outputs a sequence of encoder hidden states h_i, each state being a concatenation of forward and backward hidden states, as in [2].

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 [17].

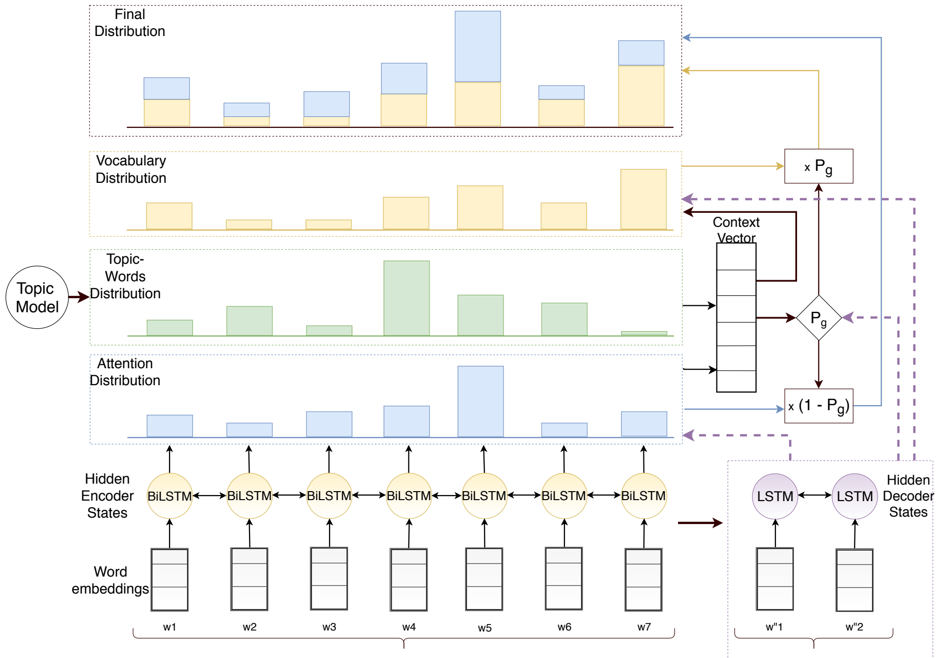

Fig. 1. The architecture of our proposed model.

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 [2] and a "topical word vector" derived from a topic model. We use LDA [8] as the topic model of choice. We chose LDA because: (1) it performs well as a component of CATS for yielding competitive summarization performance, (2) it is convenient to implement and use as its available in a few efficient topic modeling libraries, (3) and finally LDA assigns words, probabilities between 0 and 1 while the probability scores of all words in each topic sums up to 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:

$$\tau^d = \sum_i P(\text{topic}_i | d) \cdot \tilde{\mathbf{w}}_i \tag{1}$$

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

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

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

$$a^t = f(e^t, \tau^d) \tag{3}$$

where $e^t \in \mathbb{R}^K$ is a precursor attention vector, $h_k \in \mathbb{R}^n$ represents the k-th encoder hidden state and $s_t \in \mathbb{R}^l$ the decoder state at decoding step t, while $v \in \mathbb{R}^m$, $W_h \in \mathbb{R}^{m \times n}$, $W_s \in \mathbb{R}^{m \times l}$, $b_{\text{attn}} \in \mathbb{R}^m$ 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:

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

where $\tilde{\tau}^d \in \mathbb{R}^K$ denotes the "reduced" topical word vector which is formed by selecting the K components of $\tau^d \in \mathbb{R}^{|\mathcal{V}|}$ 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 $h_t^* \in \mathbb{R}^n$, as follows:

$$h_t^* = \sum_k a_k^t \cdot h_k \tag{5}$$

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 output distribution $P_{\mathcal{V}}(w)$ over all words w in vocabulary $\mathcal{V}$:

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

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

3.4 Pointer Generator

Another component of our proposed model is a copy mechanism [19]. The idea behind the pointer generator is to circumvent the limitations of pure abstraction when it comes to factual content such as names, dates of events, statistics and other content that requires copying from the source document to produce a correct summary. The basic encoder-decoder architecture often makes mistakes with people's names or other factual content while generating a summary. As a remedy, pointer networks [41] were introduced in the machine translation domain. 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:

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

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 $P_{\mathcal{V}}$ of Eq. (6).

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 final probability distribution over the extended vocabulary is:

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

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 [39] 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 [46] and define the *coverage vector* $c^t \in \mathbb{R}^K$ simply as the sum of attention 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:

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

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

Second, following [33], 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 :

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

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.

Table 1. Statistics of our meeting datasets.

	minutes	ave. #tokens per doc.	ave. #tokens per summary	minimum #tokens	median #tokens	maximum #tokens	#meetings
AMI	4868	5843	283	892	5998	11552	142
ICSI	3513	13080	449	2785	12605	22573	61
ADSC	NA	446	118	152	482	1383	45

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. As a result, at this stage the right side of Equation 4 turns into the $softmax(e^t)$ only. We believe that during training, the model parameters are optimized to best take advantage of the provided *topical attention* distribution, implicitly learning patterns of topic-words weights.

4 EVALUATION

In this section, we introduce our experimental setting, including details of our datasets, baseline models, and evaluation metrics. Finally, we present the experimental results.

4.1 Datasets

4.1.1 The CNN/DailyMail dataset. We use the CNN/DailyMail dataset [21, 28], 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). The non-anonymized version of the dataset was chosen as it presents a more realistic news wire summarization scenario.

Similar to [28, 33], 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.

4.1.2 The meetings dataset. For our empirical investigation, we compile the available datasets that have been used in previous work on meeting summarization.

For this purpose, we gathered data from the well-known AMI dataset[2] as well as the ICSI dataset[3] which are the only publicly available datasets of real-world meetings. AMI contains two categories of meetings between 2 to 4 participants. The first collection consists of freestyle meetings where the participants can decide on the topics of discussions, and targeted ones about designing technology products (*e.g.*, a remote control).

The ICSI dataset, on the other hand, contains weekly group meetings of academic groups of 3 to 10 participants. Both AMI and ICSI are face-to-face meetings that were initially audio recorded and then later transcribed. The reference summary of each meeting is then given by the manually created minutes that were taken by the original meeting participants.

We randomly divide the AMI and ICSI datasets in a 50-50 split to construct a training set as well as a test set. As a result, we end up with 101 real-world meetings as our test set and the remaining ones as the training set.

[2]http://groups.inf.ed.ac.uk/ami/download/
[3]http://groups.inf.ed.ac.uk/ami/icsi/download/

In order to increase the size of our training set we also add the Argumentative Dialogue Summary Corpus (ADSC) dataset[4] to our training set. The ADSC is composed of online conversations on topics of societal and political relevance such as gun control, gay marriage, the death penalty and abortion. Table 1 presents detailed statistics on all three datasets.

Challenges of Meeting Summarization: Most summarization research has focused on news documents for reasons of data availability. However, in addition to the small size of the existing meeting datasets, there are other aspects that make meeting summarization more challenging: (1) Most news articles are first-person narratives about a single event. Meetings, on the other hand, have a very different structure involving a dialogue between two or more parties. (2) Meetings are composed of spoken utterances between people, whereas their summaries and minutes are usually formulated from a third-person point of view by the human scribe. Therefore, meeting summarization also requires a change of structure from dialogue to a third-person narrative summarizing events. (3) Meetings can touch on multiple topics and are not restricted in terms of topical coherence. (4) Meeting transcripts include broken sentences, colloquial expressions, false starts and flawed grammar, all of which virtually never occur in carefully curated news articles. As an example, here is an excerpt from a meeting in one of the meeting datasets used in this paper which contains most of these flaws:

"mm-hmm . so sh . i 'm a bit confused about uh what 's the difference between the functional design and conceptual design ? uh i is it just uh more detail , uh as i understand it ? right . how how it will be done . so whe where do we identify the components of our uh product ? "

These issues are a common challenge of meeting transcripts and are noticeable in every meeting in the meeting datasets used in this article. Therefore, we also include the meetings dataset to also tackle a very different summarization problem as a low-resource example and show how to achieve reasonable results using our proposed model.

4.2 Baseline Models

In this section We empirically compare CATS with several abstractive baselines as follows:

- *Attention-based encoder-decoder* [28]: this abstractive model was one of the early encoder-decoder models which showed strong performance on summarization tasks.
- *PGN and PGN+Coverage* [33]: this model has been shown to effectively overcome the problem of OOV words.
- *RL with Intra-Attention* [29]: this model implements reinforcement learning to optimize summaries directly based on the evaluation metric ROUGE L. As a result, it is expected that this model would achieve a high ROUGE L performance.
- *BottomUpSum* [15]: this method uses a two-step process to generate a summary. First, it uses a content selector to identify phrases in a source document that should be part of the summary. Second, it generates a summary of the pre-selected phrases.
- *InformationSelection* [25]: this paper proposes to extend the basic attention-based encoder-decoder architecture with an information selection layer to explicitly model and optimize the information selection process. The proposed information selection layer consists of global information filtering and local sentence selection. After this step, a summary is generated using the selected sentences.
- *ML+RL ROUGE+Novel, with LM* [23]: this model aims at improving the level of abstraction of generated summaries, by generating novel sentences. In order to do so, they decompose the decoder into a contextual network that retrieves

[4] https://nlds.soe.ucsc.edu/node/30

relevant parts of the source document, and use a pre-trained language model that incorporates prior knowledge about language generation.

- *UnifiedAbsExt* [22]: this model combines extractive and abstractive summarization in an end-to-end learnable framework. Sentence-level attention is used to modulate the word-level attention such that words in less attended sentences are less likely to be generated.
- *RNN-EXT + ABS + RL + Rerank* [10]: in this model, first salient sentences are selected. Then the selected sentences are rewritten abstractively. These two steps are done using two separate neural networks. Furthermore, a sentence-level policy gradient method is used to bridge the non-differentiable computation between the two neural networks in a hierarchical way.
- *UniLM* [13]: As described in Section 2.2 UniLM is a language model whose architecture follows that of BERTLARGE and is also initialized by this model, but slightly modified its activation function and further fine-tuned for abstractive summarization.
- *T5* [31]: This work is also explained in Section 2.2. This model is also based on the Transformer architecture introduced by Vaswani et al. [40].
- *BART* [24]: BART is another top performing summarization model based on the Transformer architecture. The main contribution is the use of various noising technique for corrupting input text. For further details we refer to Section 2.2.
- *ProphetNet* [47]: The ProphetNet is yet another model based on the Transformer architecture explained in Section 2.2. The idea behind the ProphetNet is changing the original sequence-to-sequence optimization problem of predicting the next single token into predicting the n next token simultaneously.

4.3 Evaluation Metrics

Following standard practice, 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 [26]. Furthermore, by means of human evaluation, we assess the readability and informativeness of summaries generated by CATS, as well as CATS's capability to customize summaries given a set of topics.

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 truncated source lengths is set to 400 and the truncated target summary lengths is set to 100. In decoding mode (i.e. generating summaries on the test data) the beam size is 4 and the minimum target length which determines the minimum length of a generated summary is set to 35. Finally, the size of the vocabulary that CATS uses 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 parameter in the LDA model, it is important to estimate it. For this purpose, similar to the method proposed in [4, 6, 18], 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 the negative $\log P(W|M)$, where W contains all the words in the vocabulary of all the documents in the training data. 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.

The experiments reported in this paper were conducted using a Tesla V100 GPU with 18GB of RAM per node.

Based on the setup described above, in the following we present our experiments evaluating our proposed model against baselines.

4.4.1 Automatic Evaluation of Topic Customization. We first evaluate CATS in generating summaries on pre-defined topics. In order to do that we remove two topics from the output of the topic model, fine-tune the trained summarization model for a few additional training steps and compute the presence/absence of the two topics in the generated summaries.

The first topic is related to *health care* and its top five keywords are "dr", "medical", "patients", "health", and "care". The second topic is related to *police arrests and charges* with its top five words being "charges", "court", "arrested", "allegedly", and "jailed". Using the LDA model described in Section 4.4, we determine the topics of all human written summaries from the CNN/DailyMail test set. Our investigation shows that there are 752 human written summaries with the *health care* topic and 1,326 documents with the *police arrests and charges* topic. After we remove these two topics as explained above and generate summaries, we find out that the number of generated summaries of the same documents with the *health care* topic drops down to 64 and the number of generated summaries with *police arrests and charges* drops down to 255. This shows a significant decrease in the presence of the two topics in the generated summaries. Furthermore, as a reference point we examine the summaries produced by CATS without any topics removed. Our findings reveal that summaries produced by CATS have topic distributions very similar to those of human written summaries. Specifically, the number of documents containing the *health care* topic is 752 while the corresponding number for the *police arrests and charges* is 1317. These near-identical numbers were expected as CATS is trained to learn topics from target summaries.

Although, this automatic evaluation shows a clear effectiveness in removing topics from summaries, it does come with a certain limitation. For example, since different topics can share the same words among them, it might happen that certain shared words that belong to more than one topic cause an error in our evaluation. Moreover, the copy mechanism that is adopted in our model, may copy certain names from the source document that can contain words that form a topic to be removed, e.g. World Health Organization. This is the reason why the numbers of topic presences in the generated summaries although significantly lower, but cannot reach 0. Therefore, in the following subsection we also conduct a human evaluation of the customized summaries.

This experiment clearly showed the effectiveness of CATS in removing topics from summaries, when compared with both the human written summaries and the output summaries of the standard CATS.

4.4.2 Human Evaluation of Customizing Summaries. In this section, we describe the human evaluation results 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 to selectively include or exclude latent topics from the 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 number of additional training steps and analyze the effect. Our expectation is that the focus of certain output summaries which usually contain those topics will change, while naturally the raw ROUGE values are expected to decrease.

For this experiment, we chose the same two topics of the automatic evaluation 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", and "care". The second topic is related to *police arrests and charges* with its top five words being "charges", "court", "arrested", "allegedly", and "jailed". Using the topic rankings of source documents, which are provided by the LDA model described in Section 4.4, we randomly chose 100 documents from the dataset that contained either one of the aforementioned topics, given that those topics were not their sole or primary focus, but in the second rank. The reasoning is that, for example, if a news article would only cover a crime-related topic and the summarization system tries to exclude that topic from a summary, there are very few words left to form a meaningful summary. Thus, in order to systematically exploit the customization mechanism, our model also examines the topics of a given input article and determines whether excluding certain topics from its summary is feasible.

Five human judges evaluated whether the summaries generated by CATS with restricted topics showed exclusion or reduction of those topics or whether there was no major difference. In other words, for each given system-generated summary, its corresponding human-written summary and the original news article, human judges could select either full exclusion of a target topic, reduction of a target topic, or no meaningful change. 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, out of the 100 documents, the majority of the human judges find a full exclusion of a target topic in 87 documents, a reduction of the target topic in ten documents, and no major difference in only three documents. The Kappa agreement between the five human judges is 0.704.

Based on this experiment, we conclude that CATS can in most cases reliably customize summaries by controlling the topics that appear in them, and we attribute this capability to the *topical attention* mechanism. Our model is the first to bring customization of abstractive summaries in sequence-to-sequence architectures. Such feature, can be beneficial for editorial boards of publishers, e.g. news channels who would like to enforce policies regarding the topics of the content they publish. This can also be used at hospitals where doctors need to quickly obtain information from long electronic health-care records of patients regarding a certain illness. For example, a doctor attending a heart condition of a patient might not need information about a previously broken arm and therefore may would like to filter-out such irrelevant information.

Table 2 shows an example summary produced by CATS that was restricted not to include the *health care* topic, alongside a summary produced by CATS restricting the *crime* topic and CATS with no topic restriction, as well as the corresponding human-written reference summary. We observe that in the first two columns the focus of the summary is altered such that it focuses on the crime-related thematic rather than health care and vice versa in order to avoid using words such as "hospital", "patients" and "medicine" in the first column and words such as "murdering", "guilty", "charges", "denies" in the second column.

Table 3 shows another similar example where CATS is restricted not to include the *health care* topic and separately the *crime* topic.

We observe from the two examples that CATS generates summaries that read fluently in both topic-restriction and no-restriction modes.

4.4.3 The impact of topic model. In this section, we analyze the impact of the topic model in achieving summarization performance in terms of ROUGE. We already discussed how we train the LDA model in Section 4.4 using the training data. However, since the LDA model is unsupervised and can be trained in an online training process using new documents, we could also train it using both training as well as testing datasets. In this section we compare the performance of CATS in terms of ROUGE metrics in the situation where the unsupervised LDA topic model is trained only on training data compared with when it is trained on both training and testing datasets.

Table 2. Comparison of a CATS generated summary next to a summary with restricted topics and the human-written reference summary[5]. The words related to the crime topic are colored in red, while words related to health-care are in green.

CATS restricting health-care	***CATS restricting crime***	***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 he didn 't poison patients at stepping hill hospital in stockport . a nurse today told he did not poison hospital patients on his ward by contaminating their medicine with insulin .	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 .

Table 3. Comparison of a CATS generated summary next to a summary with restricted topics and the human-written reference summary[6]. The words related to the crime topic are colored in red, while words related to health-care are in green.

CATS restricting health-care	***CATS restricting crime***	***CATS***	***Reference***
darwin man is accused of using someone else 's employee registration number to pose as a fake employee at the aurukun primary health centre . he was charged on saturday with one count of fraud after cairns detectives made contact with him in the northern territory .	a 30-year-old darwin man posed as a nurse at the aurukun primary health centre on cape york during february and march . health authorities are searching through patient records after it was revealed man did not have the correct qualifications .	a 30-year-old darwin man is accused of using a female nurse 's registration number at the aurukun primary health centre on cape york during february and march . he was charged on saturday with one count of fraud after cairns detectives made contact with him in the northern territory . he was receiving a $ 100,000 annual salary and accommodation from queensland health in the six weeks he was at the hospital .	man , 30 , is accused of using a female nurse 's employee number to work . he worked for six weeks at aurukun primary health centre on cape york . man was charged with fraud after payroll raised the alarm with hospital . authorities are checking patient records to see who he interacted with .

In the results presented in Table 4, we observe that when the topic model is fine-tuned using the test data, the performance significantly improves in terms of ROUGE 1 and ROUGE L while showing slight improvement in terms of ROUGE 2. Therefore, we conclude that the training of the topic model is an essential factor in summarization performance.

4.4.4 Comparison in terms of ROUGE. In this section we compare our proposed model 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 5.

Table 4. Comparison between our model trained using LDA trained on training data against our model trained using LDA trained on both training and test data in terms of F_1 ROUGE metrics on the CNN/Dailymail dataset. Statistical significance test was done with a confidence of 95% and confirmed significance.

Models	ROUGE 1 (%)	ROUGE 2 (%)	ROUGE L (%)
CATS (LDA:training data)	41.76	18.69	38.21
CATS (LDA:training+testing data)	42.13	18.85	38.63

Table 5. Comparison between our proposed model against the baselines in terms of F_1 ROUGE metrics on the CNN/Dailymail dataset. '*' means that results are based on the anonymized version of the dataset and not strictly comparable to our results. The bottom four models utilize pre-trained Transformer-based architectures.

Models	ROUGE 1 (%)	ROUGE 2 (%)	ROUGE L (%)
CATS (Ours)	42.13	18.85	38.63
LEAD-3 Baseline	40.34	17.70	36.57
Attn. Enc-Dec (Nallapati et al. [28])	35.46	13.30	32.65
PGN (See et al. [33])	36.44	15.66	33.42
PGN+coverage (See et al. [33])	39.53	17.28	36.38
RL with Intra-Attention (Paulus et al. [29]) '*'	41.16	15.75	39.08
BottomUpSum (Gehrmann et al. [15])	41.22	18.68	38.34
InformationSelection (Li et al. [25])	41.54	18.18	36.47
ML+RL ROUGE+Novel, with LM (Kryscinski et al. [23])	40.19	17.38	37.52
UnifiedAbsExt (Hsu et al. [22])	40.68	17.97	37.13
RNN-EXT + ABS + RL + Rerank (Chen and Bansal [10])	40.88	17.80	38.54
UniLM (Dong et al. [13])	43.33	20.21	40.51
T5-small (Raffel et al. [31])	41.12	19.56	38.35
T5-largest (Raffel et al. [31])	43.52	21.55	40.69
BART (Lewis et al. [24])	44.16	21.28	40.90
ProphetNet (Yan et al. [47])	44.20	21.17	41.30

We can observe that our model outperforms all other non-Transformer-based models in terms of ROUGE 1 and ROUGE 2 while being behind the Transformer-based models (the bottom four models in the table). In order to verify the robustness of findings, we conduct a statistical significance test based on the bootstrap re-sampling technique using the official ROUGE package [26]. In the case of ROUGE L, [29] reports the highest performance among the non-Transformer-based models; however, this is due to their model loss function optimizing directly for the evaluation metric ROUGE L instead of the summarization loss. In fact, [22] reports an experiment that shows summaries generated by the [29] method achieve the poorest readability scores compared with a number of models including PGN and their own UnifiedAbsExt model, a finding which we also confirmed by comparing the output summaries with the output of our model (see Section 4.4.7). This indicates that optimizing on ROUGE L instead of the summarization loss adversely impacts the quality of the produced summaries. We discuss this point further in Section 4.4.7 where we qualitatively compare our generated summaries against that of [29].

We note that we did not include the method of [9] in our comparison, due to the fact that unlike most papers that use preprocessing scripts of [33] for the non-anonymized version of the dataset, they use different scripts. The effect of this difference on their LEAD-3[7] baseline remains unclear as they do not report it. Thus, their results may not be comparable with ours.

In this experiment, we conclude that among non-Transformer-based baselines our model achieves superior performance as compared with other baselines. However, the Transformer-based models outperform CATS in terms of ROUGE metrics. This is while the training time, computational resources, and the training dataset size used for preparing our model is only a small fraction of that of the Tranformer-based models. Let us take ProphetNet [47], the best performing model in terms of ROUGE, as an example. The authors explicitly mention that their model has been trained with a 160GB dataset, then with another 16GB dataset, and finally fine-tuned using the CNN /Dailymail dataset. However, our model has been only trained using the CNN/Dailymail dataset.

For the smaller versions of the Transformer models which, similar to our model, are also trainable from scratch, we report the results of the small T5 model as a point of reference. The reason for reporting only the T5 is that it is the only model for which the size-performance trade-off is explored by the original authors [31]. As we observe in Table 5, our proposed model outperforms the T5-small in terms of ROUGE 1 and ROUGE L but it lags behind in terms of ROUGE 2.

Besides the data efficiency of CATS, the design goal behind our model is the capability of customizing summaries based on given topic requirements. This is something that no other model discussed in this article has been shown to be capable of.

4.4.5 Comparing variations of CATS in terms of ROUGE. This section performs an ablation study, measuring the impact of individual CATS components on ROUGE scores. We first present the setup of CATS used in all experiments throughout this article followed by other variations to determine the effect of each component on the model's summarization performance:

(1) CATS: The standard setup of CATS using topical attention, as explained in Section 3. It focuses on topics of the target summaries at training time without using any topic information at test time. Additionally, CATS uses a coverage component as explained in the same section.
(2) CATS-Source-Topics: This variation uses topical attention focusing on topics *of source articles* at training time without using any topic information at test time.
(3) CATS-Source-Topics-TrainTest: This variation uses topical attention which focuses on topics of source articles during training, but differently from the above variations, also uses topic information of source articles at test time.
(4) CATS-No-Coverage: This variation of standard CATS omits the coverage mechanism.
(5) CATS-No-Topical-No-Coverage: We fully remove the topical attention of CATS and also remove the coverage mechanism. Under such settings CATS is reduced to a basic pointer generator network.

Table 6 presents the results of the ablation study. We observe that having a topical attention focusing on topics derived from target summaries during training time outperforms other variations of topical attention. We believe that focusing on topics of target summaries enables CATS to generate summaries precisely to the point as presented in the target summary. The fact that this variation outperforms all other variations may be caused by the model learning attention weights as a complement to the topic-words weights so precisely that providing this information at test time does not improve the summarization performance any further. As we remove the coverage mechanism or even the entire topical attention scheme, performance noticeably deteriorates.

[7] The LEAD-3 baseline is taking the first three sentences of an article as its summary. This baseline is commonly used in automatic summarization as a reference to evaluate a dataset.

Table 6. Ablation study between the full CATS model and a number of reduced/altered variants in terms of F_1 ROUGE metrics on the CNN/Dailymail dataset.

Models	ROUGE 1 (%)	ROUGE 2 (%)	ROUGE L (%)
CATS	42.13	18.85	38.63
CATS-Source-Topics	41.22	17.98	37.39
CATS-Source-Topics-TrainTest	40.88	17.73	37.12
CATS-No-Coverage	38.13	16.52	35.03
CATS-No-Topical-No-Coverage	36.44	15.66	33.42

Table 7. F_1 *ROUGE* scores on AMI/ICSI test sets.

	ROUGE 1	ROUGE 2	ROUGE L
CATS No-TL	12.13	1.54	11.15
CATS	**30.85**	**8.89**	**28.50**

4.4.6 *Low-resource Abstractive Summarization using Transfer Learning with CATS.*

In this section, we introduce a transfer-learning approach for abstractive summarization of a very small dataset of meetings transcripts. We first train CATS on the CNN/ DailyMail news dataset. Our transfer-learning approach is based on fine-tuning and adapting model parameters to the new task of meeting summarization.

As a result, after we pre-train CATS on the news dataset, we fine-tune it as follows: We feed our model with the meeting training dataset described in Section 4.1.2. We use a small learning rate to tune all parameters from their original settings to minimize the loss on the new task. Moreover, we increase the minimum number of tokens generated from 35 to 65 to account for the greater length of meeting transcripts and corresponding summaries.

Fine-tuning adapts the model's parameters to make it more discriminative for the new task, and the low learning rate is an indirect mechanism to preserve some of the representational structure learned in the news summarization task. Moreover, we expose CATS to the meeting training data for 50 epochs on the meeting training set with a batch size of 16.

Since our model utilizes LDA we need to add the training examples to the LDA model as well. That also changes the derived topics given to the topical attention mechanism.

We begin evaluating this approach by comparing our model in terms of the F_1 *ROUGE* metrics against our model when the transfer-learning approach described above is applied. Table 7 illustrates the results of this experiment.

As we can observe in the table, our model with transfer-learning significantly outperforms the model without transfer-learning in terms of ROUGE 1 and ROUGE L. Our statistical significance test is based on bootstrap re-sampling using the official ROUGE package [26] and confirms that the observed improvement over the baselines in terms of ROUGE metrics is significant with a confidence of 95%.

The most important finding of this experiment is the comparison of our model against its equivalent version without transfer-learning. The considerable improvement in performance corroborates that our transfer-learning approach is very effective in building a meeting abstractive summarization system, while producing summaries which are in a third-person-view and contain no colloquial expressions.

4.4.7 Human Evaluation of Summaries. We conduct a manual evaluation in order to assess the quality of summaries produced by CATS compared to the summaries of PGN+coverage [33] and RL with Intra-Attention [29], which were provided by the authors of these methods. We chose the RL with Intra-Attention since it was the only method optimizing on ROUGE L and thus had a higher ROUGE L. We examine informativeness and readability of 50 randomly sampled summaries. When comparing the output produced by the three models, the three human assessors[8] 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 8.

Table 8. Human evaluation comparing quality of summaries on a 1-5 scale using three evaluators.

	Readability	Informativeness
CATS	**4.1**	**3.9**
PGN+Coverage	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.

4.4.8 Analysis of Repetition in Output Summaries. In this experiment we analyze the quality of the output summaries produced by CATS 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 has been introduced to mitigate this undesirable effect, and we show that our model can reduce it even further.

We compare CATS to PGN and PGN+coverage in terms of n-grams repetition with n ranging from 1 to 6. For this purpose, and to exclude possible influence of better hyperparameter tuning, we train all three models using the optimal hyperparameters found for PGN+coverage, 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 the per-document n-gram repetition score, $S_{\text{rep,doc}}$, over all test output documents, where we define:

$$S_{\text{rep,doc}} = \frac{\#\text{duplicate n-grams}}{\#\text{all n-grams}} \tag{12}$$

We observe that our model exhibits drastically lower repetition of text in its 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. Although PGN+coverage was originally designed to overcome the repetition problem, the results of this experiment indicate that our proposed topical attention mechanism reduces repetition significantly.

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 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 is a very effective solution to the repetition problem in sequence generation based on encoder-decoder architectures.

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

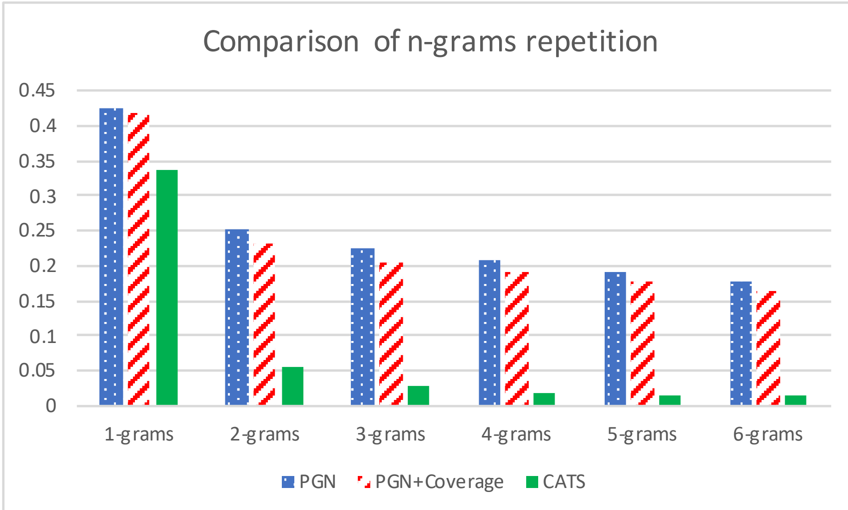

Fig. 2. Experiment comparing the degree of n-grams repetition in our model versus that of the PGN and PGN+coverage baselines on the CNN/Dailymail test set. Lower numbers show less repetition in the generated summaries.

4.4.9 *Readability experiment:*

This experiment is designed to measure the readability of the output summaries generated by the various models. For this purpose we use the Automated Readability Index (ARI) [34]. ARI is a measure for gauging how understandable a piece of text is. The results of the experiment, reported in Table 9, show that CATS yields superior readability compared to other models and variations. It is worth noting that CATS with topics removed performs very close to CATS in terms of automatic readability scores, suggesting high overall text generation quality. The table additionally presents basic statistics on average number of tokens per sentence as well as average number of characters per token.

Table 9. Comparing the performance of our model vs. PGN, with respect to readability of output summaries

	Ground-truth	CATS-without-coverage	CATS	CATS-with-topics-removed	PGN	PGN+coverage
ARI	28.40	23.43	34.14	23.86	22.59	23.66
Ave. # tokens per sentence	14.30	23.12	23.82	23.43	20.90	23.92
Ave. # chars per token	4.70	4.64	4.56	4.66	4.61	4.62

4.4.10 *Summary coherence experiment:*

This experiment is designed to measure the coherence of the output summaries generated by the various models. For this purpose we use the Normalized Pointwise Mutual Information (NPMI) which is an established measure for quantifying coherence between words. We compute the coherence of a summary by computing NPMI between all word pairs of every two consecutive sentences normalized by the number of sentences in the summary. Each sentence is identified by punctuation marks such as ".", "?" and "!". We formally define coherence of a summary s consisting of sentences $sent_1, \ldots, sent_n$ as:

$$coherence_s = (NPMI(sent_1, sent_2) + NPMI(sent_2, sent_3) + \cdots + NPMI(sent_{n-1}, sent_n))/n$$

This metric quantifies the relatedness of sentences of a document. In order to compute the coherence of summaries we remove stop words, punctuation marks as well as all non alphabetic tokens such as numbers. Then we compute the coherence produced by the different methods.

In this experiment we compare CATS against CATS with the crime topic removed. Table 10 shows the results of this experiment.

Table 10. Comparing the performance of CATS vs. CATS-with-topics-removed, with respect to coherence of output summaries

	CATS	CATS-with-topics-removed
Coherence	0.00754	0.00823

As we observe from the table CATS-with-topics-removed achieves a higher coherence score compared with CATS. This outcome was expected, since CATS aims for covering all topics present in a source article. Subsequently, since the NPMI score between words which come from different topics are lower, the overall coherence score is also lower. In the case of CATS-with-topics-removed, however, we observe that the summaries are more focused and therefore yield a higher coherence score.

In this experiment, we showed that when we remove a certain topic in summaries produced by CATS, we observe a higher coherence score.

5 DISCUSSION

In the previous sections we have presented and extensively evaluated CATS. In this section, we discuss the use cases of CATS in its current form, potentially significant improvements and modifications for future work, and, finally, the potential use of topical attention in other sequence-to-sequence neural architectures.

Prospective use cases of CATS: As previously mentioned, compared to transformer-based models that typically require large scale pre-training, CATS has the advantage of being trained on a relatively small dataset, while outperforming all baselines on the standard abstractive summarization task, except for the large-size variants of the transformer-based models. In addition to standard summarization, we also introduced and tackled the problem of topic-based summarization. We have qualitatively demonstrated the effectiveness of a fine-tuning method for custom-generation of summaries by focusing on a few topics and discarding others. In order to use this topic-based summarization feature of CATS in practice, it is currently necessary to fine-tune multiple instances of CATS beforehand, each including/excluding certain topics. These thematically customized models can be deployed on cloud infrastructure and be accessed through an API on demand, so as to serve specific information needs (e.g. a journalist covering only US - China relations as a part of international relations, or only trade as a part of US - China relations). Although deploying multiple specialized model instances in parallel is a paradigm widely used in industry (e.g. for machine translation between numerous language pairs), it comes with practical limitations with respect to infrastructure, maintenance and development time. In the following, we will discuss possible alternatives to fine-tuning for topic control, which is a topic of active, ongoing research.

Alternative topic control mechanisms for custom generation: A first solution to obviate the need for fine-tuning multiple instances, each focusing on a different set of topics, is to prepare a dataset with topic-specific summaries.

Such a dataset will contain articles and two or more summaries corresponding to each article, such that each summary focuses on only one (or a subset) of the few topics present in the document. In this way, during training, CATS or other similar sequence-to-sequence models will learn how to generate a summary focused on a topic (or subset of topics) indicated as input. To elaborate, each topic will be specified with a unique token which will be fed along with the input document tokens to the encoder, and the expected output of the decoder will be a summary with a focus on the corresponding topic(s). We are currently developing such a dataset and will soon release it as the first dataset on customized topic-based summarization to be used by the community for building advanced summarization systems. Interestingly, the existing fine-tuned CATS models can be used to generate the topic-specific summaries of this dataset.

A second, promising solution for controlling generation is to add a regularization term to the model's loss function in order to explicitly drive the attention mechanism to learn the distribution over input words as induced by the topic model. Specifically, during training we can use the KL divergence, Wasserstein distance or similar metrics which measure differences between distributions, to penalize the deviation between the precursor attention weights e^t (Eq. (2)) and the topical word distribution τ^d induced by a topic model (Eq. (1)). This method can potentially direct the model to attend to a source document in the same way as suggested by a distribution over words coming from a topic model. Moreover, certain topics can be turned off or on in the distribution.

The third possible solution that also relies on the dedicated dataset explained above (as the first solution) is to extract the topic-words distribution from the model's output summaries, and penalize its distance from the intended topic-words distribution specified by a user through a regularization term in the loss function.

Finally, a fourth solution is to train a CATS model as usual, but modify the beam-search text generation algorithm such that during inference it would assign higher probabilities for generating words that are indicated by a topic-words distribution. That is, a penalty term would be added to words that are likely to be generated by the normal beam-search but are not in line with a topic-words distribution indicated by a user.

In summary, we discussed a number of solutions that can be used to enhance the practicality and effectiveness of our topic-based, customizable summarization model. We believe that combining two or more of the above solutions can potentially result in a robust topic-based summarization. The above ideas are directions of our current research and future work.

Integrating the topical attention into other neural architectures: In the standard summarization experiments reported in the previous section, the concept of topical attention was shown to improve the quality of summaries compared to the same architecture without topical attention.

The recent advancements in abstractive summarization research has been mostly due to the advent of the transformer model. As discussed in Section 2, all recent top-performing summarization models are variants of the original Transformer model [40]. While in very recent work [44] the incorporation of topic models in transformer-based summarization systems is emerging as a beneficial component, we believe that our idea of topical attention can be directly used in transformer-based models even in its current form as presented in Equation (4) to mediate between the encoder and decoder as cross-attention. That is, the topic-words weights are integrated into the cross-attention weights. Adapting the topical attention mechanism to other transformer-based models, also taking into account the ideas presented in the previous paragraph, is the focus of our ongoing research.

6 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 CATS achieves performance superior to all non-transformer-based models in terms of standard evaluation metrics for summarization (*i.e.* ROUGE) on a standard benchmark dataset, while drastically reducing sequence repetition, and, crucially, enabling customization of produced summaries.

Moreover, we showed a transfer-learning approach for applying CATS to small datasets and low-resource cases.

CATS can serve as a foundation for future work in the domain of automatic summarization. Based on the results of this paper, we are optimistic about the potential of future summarization systems to generate summaries which are customized to users' needs. We envision three ways of controlling the focus of output summaries using CATS: First, as demonstrated in the experiment in Section 4.4.2, 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 or collection of documents. 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.

Finally, we are interested in exploring the use of dedicated, fully neural topic modeling modules, whose parameters are learned either using unsupervised pre-training or from scratch during end-to-end training of the sequence-to-sequence model.

ACKNOWLEDGMENTS

This research is supported in part by the NSF (IIS-1956221), SNSF (P2TIP2_187932), ODNI and IARPA via the BETTER program (2019-19051600004). The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of NSF, SNSF, ODNI, IARPA or the U.S. Government.

REFERENCES

[1] Mohammad Aliannejadi, Morgan Harvey, Luca Costa, Matthew Pointon, and Fabio Crestani. 2019. Understanding Mobile Search Task Relevance and User Behaviour in Context. In *Proceedings of the 2019 Conference on Human Information Interaction and Retrieval (CHIIR '19)*. New York, NY, USA, 143–151.

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

[3] Seyed Ali Bahrainian. 2019. *Just-In-Time Information Retrieval and Summarization for Personal Assistance*. Ph.D. Dissertation. Università della Svizzera italiana.

[4] 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)*. 150–159.

[5] Seyed Ali Bahrainian and Andreas Dengel. 2015. Sentiment analysis of texts by capturing underlying sentiment patterns. In *Web Intelligence*, Vol. 13. 53–68.

[6] Seyed Ali Bahrainian, Ida Mele, and Fabio Crestani. 2018. Predicting topics in scholarly papers. In *European Conference on Information Retrieval*. Springer, 16–28.

[7] Michele Banko, Vibhu O Mittal, and Michael J Witbrock. 2000. Headline generation based on statistical translation. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 318–325.

[8] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *J. Mach. Learn. Res.* 3 (2003), 993–1022.

[9] 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)*. 1662–1675.

[10] 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*. 675–686.

[11] Trevor Cohn and Mirella Lapata. 2008. Sentence compression beyond word deletion. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*. Association for Computational Linguistics, 137–144.

[12] Fabio Crestani and Heather Du. 2006. Written versus spoken queries: A qualitative and quantitative comparative analysis. *Journal of the American Society for Information Science and Technology* 57, 7 (2006), 881–890.

[13] Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems*. 13042–13054.

[14] Ferenc Galkó and Carsten Eickhoff. 2018. Biomedical Question Answering via Weighted Neural Network Passage Retrieval. In *European Conference on Information Retrieval*. Springer, 523–528.

[15] 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*. 4098–4109.

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

[17] 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 (2017), 2222–2232.

[18] Thomas L Griffiths and Mark Steyvers. 2004. Finding scientific topics. *Proceedings of the National academy of Sciences* (2004).

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

[20] Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415* (2016).

[21] 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*. 1693–1701.

[22] 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*. 132–141.

[23] 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*. 1808–1817.

[24] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461* (2019).

[25] 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*. 1787–1796.

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

[27] 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*.

[28] 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*. 280–290.

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

[30] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. [n.d.]. Improving language understanding by generative pre-training. ([n. d.]).

[31] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683* (2019).

[32] Nuzhah Gooda Sahib, Anastasios Tombros, and Tony Stockman. 2012. A comparative analysis of the information-seeking behavior of visually impaired and sighted searchers. *Journal of the American Society for Information Science and Technology* 63, 2 (2012), 377–391.

[33] 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*. 1073–1083.

[34] RJ Senter and Edgar A Smith. 1967. *Automated readability index*. Technical Report. CINCINNATI UNIV OH.

[35] 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* (2018).

[36] Paul Thomas, Daniel McDuff, Mary Czerwinski, and Nick Craswell. 2017. MISC: A data set of information-seeking conversations. In *Proceedings of the 1st International Workshop on Conversational Approaches to Information Retrieval.*

[37] 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.* ACM, 2–10.

[38] Johanne R Trippas, Damiano Spina, Lawrence Cavedon, Hideo Joho, and Mark Sanderson. 2018. Informing the Design of Spoken Conversational Search: Perspective Paper. In *Proceedings of the 2018 Conference on Human Information Interaction&Retrieval.* ACM, 32–41.

[39] 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.*

[40] 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.* 5998–6008.

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

[42] Chong Wang, David Blei, and David Heckerman. 2008. Continuous time dynamic topic models. *Proc. of UAI* (2008).

[43] Li Wang, Junlin Yao, Yunzhe Tao, Li Zhong, Wei Liu, and Qiang Du. 2018. A Reinforced Topic-aware Convolutional Sequence-to-sequence Model for Abstractive Text Summarization. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI'18).* 4453–4460.

[44] Zhengjue Wang, Zhibin Duan, Hao Zhang, Chaojie Wang, Long Tian, Bo Chen, and Mingyuan Zhou. 2020. Friendly Topic Assistant for Transformer Based Abstractive Summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP).* 485–497.

[45] Chen Xing, Wei Wu, Yu Wu, Jie Liu, Yalou Huang, Ming Zhou, and Wei-Ying Ma. 2017. Topic aware neural response generation. In *Thirty-First AAAI Conference on Artificial Intelligence.*

[46] 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.* 2048–2057.

[47] 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. *arXiv preprint arXiv:2001.04063* (2020).

[48] David Zajic, Bonnie Dorr, and Richard Schwartz. 2004. Bbn/umd at duc-2004: Topiary. In *Proceedings of the HLT-NAACL 2004 Document Understanding Workshop.* 112–119.