Spring 2022

Bridging the Gap Between Retrieval and Summarization

Connor Lennox
University of New Hampshire, Durham

Bridging the Gap Between Retrieval and Summarization

BY

CONNOR LENNOX

BS, University of New Hampshire, 2021

THESIS

Submitted to the University of New Hampshire

in Partial Fulfillment of

the Requirements for the Degree of

Master of Science

in

Computer Science

May, 2022

This thesis/dissertation has been examined and approved in partial fulfillment of the requirements for the degree of Master of Science in Computer Science by:

Thesis Director, Dietz, Laura, Assistant Professor, Computer Science

Petrik, Marek, Assistant Professor, Computer Science

Varki, Elizabeth, Associate Professor, Computer Science

On April 21, 2022

Original approval signatures are on file with the University of New Hampshire Graduate School.

TABLE OF CONTENTS

ABSTRACT

Information Retrieval is, at its core, a field focused on providing information to users to fulfill an information need. One of the most common use cases of Information Retrieval is document-level retrieval, which seeks to provide a collection of documents to the user that addresses their needs. In contrast to this, single document retrieval seeks to instead provide the user with a single document comprised of all required information. We seek to extend single document retrieval to single document generation, in which we use multiple source documents to create a new document which directly addresses the information need.

CHAPTER 1

Introduction

When automatically generating text one common issue is readability. Regardless of the information present, a poorly formatted or structured document is not useful for a user, who must then spend significant time to read, re-read, and fully understand the information being presented. In Chapter 2, we discuss this problem in more depth and present an entity-oriented method of structuring generated text. We focus on ordering sentences to avoid discordant ideas in adjacent sentences and promote a smooth reading experience for the user, where ideas naturally flow from one to the next. Specifically, we augment modern sentence ordering models with entity-related information and show significant improvement over the state-of-the-art.

The novelty of our ordering technique comes from the inclusion of entity information, and more specifically the method of incorporating said information. We seek to augment preexisting sentence-ordering models during fine-tuning, which avoids the requirement for large amounts of data with entity annotations.

Beyond the structuring and organization of articles, we must also consider the content contained therein. In chapter 3 we explore a multi-document summarization approach tailored to the article generation task. This approach operates in several stages, including information extraction, redundancy reduction, and the aforementioned information ordering. We note that our previous work on information ordering is directly applicable for the article generation task, being the final stage of the pipeline.

We show that our generated articles produce coherent, informative text and our "information-first" methodology allows for output statements to be traced directly to a source document. These techniques are a step towards more interpretable machine learning and verifiable models. In addition, our cluster-based technique addresses common problems in multi-document summarization.

CHAPTER 2

Query-focused Pointer Networks for Relevant Information Ordering

Abstract

Modern search interfaces require to sequence information into a meaningful order. Hence, merely ranking information by relevance is not enough. We study the task of ordering pieces of information into a relevant and coherent response to the search query. Pointer networks are a natural machine learning approach for ordering tasks. However, current approaches focus on text embeddings, without placing much emphasis on the relevance of information. We propose a novel approach that integrates information about the search query and relevant entities into pointer networks. This yields significant performance improvements with respect to ordering information in Wikipedia lead paragraphs.

2.1 Introduction

In information retrieval we discuss relevance mostly in the context of ranking information by relevance. By placing the most relevant information first, the second-most relevant information next, etc IR systems provide a list of information that is likely to be relevant, leaving it to the user to refine the selection. However, skimming a list is cumbersome on modern devices such as mobile phones or smart speakers, especially when the ranking switches back and forth between relevant topics.

In this work we address a usage scenario where relevant information is to be ordered in a meaningful way. Here *meaningful* implies that the result (if presented in this order) is coherent, relevant, and non-redundant. Given the vast body of work on retrieval and redundancy removal [5, 14], we assume that a non-redundant set of relevant information is readily available. Hence, in this work we focus on the prediction of a meaningful order.

Task. Given a query q and a set of K relevant passages $s \in \mathcal{S}$, predict an ordering of passages $s_1, s_2, \ldots, s_K$ so that when read in this order yields a coherent and relevant response to the query.

We define an ordering model $\mathcal{O}(s \in \mathcal{S}_{\succ} | \mathcal{S}_{\prec})$ that for each element s produces a probability of being placed next, given the order of preceding elements $\mathcal{S}_{\prec}$ and remaining elements $\mathcal{S}_{\succ}$.

When humans perform the passage ordering task, there are two main aspects that inform this process. The first is the *coherent* ordering, in which passages that logically flow together are to be placed adjacent to each other. The other aspect is the *relevant* ordering, where the focus is to present the most relevant information first. Often, some context needs to be provided before relevant information can be understood by the user—especially for complex, technical topics. Hence, passages elaborating the context need to be identified and ordered before passages containing the relevant information, otherwise the user will be confused while reading the result. In this paper we are making the following contributions.

- Developing Query-focused PtrNets, which incorporate query relevance into PtrNets for passage orderings that consider both coherence and relevance.
- Incorporate query-entity relevance information into the ordering process.
- A detailed empirical study, demonstrating a statistically significant 4.4% improvement on a dataset derived from Wikipedia.

2.2 Related Work

Passage ordering has been studied both with traditional methods and in the neural networks. Prior approaches use part-of-speech tags to identify patterns in entity appearances, assuming that entities are introduced first as a subject before being used as an object. A measure of "coherence" is derives as the spread of entities throughout the text [1, 13]. Other pre-neural approaches consider pairs of passages and recursively build an ordering from both passages and sub-orderings of passage pairs [3].

Most modern approaches to passage ordering depends on some forms of a text embedding, usually created with pre-trained contextual embedding models [45]. Cui et al. [7] suggest to order passages are embedded into a contextual format with a recurrent neural network, such as a Pointer Network. Pointer networks [39] produce a sequence of outputs from a given input set, by predicting pointers into that set. They have proven useful for passage ordering tasks, in which a set of passages is expected to be returned in an order that is coherent for the reader [26] as well as several other related domains [19, 35].

2.3 Background: Pointer Networks

Pointer Networks (PtrNets) [39] are a canonical framework for ordering a given set of elements $\mathcal{S}$. The approach produces a list of pointers $p_1, p_2, ...p_K$ which refer to elements s_j in the input set $\mathcal{S}$. The element referenced by pointer p_i is dependent on the settings of previous pointers $\mathcal{S}_\prec$ and initial state $\vec{h}_0$.

Pointer networks are based on recurrent neural networks where each step takes an input $\vec{x}_i$ and the previous state $\vec{h}_{i-1}$ to produce an output $\vec{y}_i$ and a new state $\vec{h}_i$. The output of each recurrent step is a probability distribution over the input elements, modeled via softmax. In the PtrNet, the output $\vec{y}_i$ represents a probability distribution over the references in the (unordered) set of remaining inputs $\mathcal{S}_\succ$ for the i'th pointer p_i. The preceding order $\mathcal{S}_\succ$ is represented by state h_{i-1}. When using these distributions to produce an ordering, we take the element with the highest probability: $p_i = \arg\max_{s \in \mathcal{S}_\succ} [\mathcal{O}(s \in \mathcal{S}_\succ | h_{i-1})]$.

To increase the accuracy of PtrNets, passages are often encoded before being provided to the pointer-network itself. In previous work, this is done with a sentence graph. A graph is created over the input passages, where edges represent some predicate on a passage pair, and a graph-based encoder is used to produce a dense representation of the passages [42].

2.4 Approach: Query-specific Pointer Networks

The PtrNet is a useful neural architecture for ordering text passages in a semantically coherent manner. The scope of our work is to develop query-focused PtrNets, which incorporate various notions of relevance with the goal of improving the order in which different pieces of information are presented to the user.

2.4.1 GAT: Representing Passages in Query Context

Based on a discussion in [45], we hypothesize that BERT embeddings of short passages might not be sufficient for ordering as they offer little contextual information with respect to the collection as a whole. Following ideas regarding attention-based sentence encodings from Cui et al. [7], we leverage a graph attention network [38], to learn a better passage representation as follows: We construct a symmetric sentence graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where each passage is represented by a node s, and a symmetric edge $(s_1, s_2) \in \mathcal{E}$ exists based on a predicate which is defined further below.

GAT: Using BERT passage representations as inputs $\vec{s}$, a new representation $\vec{z}_s$ for each passage s is estimated from the subset of context of other the passages $s' \in \mathcal{S}$ for which an edge $(s, s') \in \mathcal{E}$ exists in the graph. We implement this through neural graph attention networks, which use scaled dot product (aka QKV) attention [37] to govern a multi-head attention vector $\vec{a}$, with distinct vector-valued projection functions for each: $Q(\vec{s})$, and both $K(\vec{s'})$ and $V(\vec{s'})$:

$$\vec{z}_s = \sum_{s':(s,s')\in\mathcal{E}} \vec{a}\left(Q(\vec{s}), K(\vec{s'})\right) \cdot V(\vec{s'})$$

Shared entity edges: We hypothesize that two passages are likely to discuss related information when they share at least one entity. We guide the GAT encoder by including edges

$(s, s') \in \mathcal{E}$ if they share at least one entity. This adaptation leads to a passage representation that absorbs cross-passage entity-centric knowledge, which we believe is crucial for producing coherent orderings.

In an preliminary study without this guidance, we observed unhelpful representations due to over-smoothing. Yin et al. made similar observations [42].

Query-GAT: As motivated in Section 2.1, we want highly relevant information to be presented as early as possible. We include the query as another node q in the graph $\mathcal{G}$; then add an edge between the query q and passage node s to $\mathcal{E}$. We derive a new context-based representation z_q of the query. The effect is that the passage representations z_s can also incorporate similarities to the query.

Query-Init: While by default the state of the PtrNet is initialized with a zero vector $h_0 = \vec{0}$, we set the initial state $h_0 = z_q$ to the query's encoded representation.

2.4.2 Incorporating Entity Relevance

We hypothesize that users want to read about the most relevant entities first. We lean on the entity retrieval literature [34] to incorporate external information about the query-specific relevance of entities to predict whether a passage that mentions relevant entities should be promoted to the front.

Given a ranking of entities, an entity-based feature vector for each passage s is derived. For entities that are mentioned in the passage we note their presence, rank, and score in the entity ranking to compute the following features: number of relevant/non-relevant entities present in the passage $(\mathrm{f}_1, \mathrm{f}_2)$, mean/median of their entity scores $(\mathrm{f}_3, \mathrm{f}_4)$, and their reciprocal entity rank $(\mathrm{f}_5, \mathrm{f}_6)$.

Applying the ListNet learning-to-rank model [4] to the ordering task, we use a regression model with parameter $\vec{w}$ to govern the order model $\mathcal{O}_{\mathrm{rel}}(s \in \mathcal{S}_{\succ}) = p(s|\mathcal{S}_{\succ})$. The model is trained to predict the next passage s from a set of (remaining) passages $\mathcal{S}$ to maximize $p(s|\mathcal{S}) = \frac{\vec{w} \cdot \vec{\mathrm{f}}_s}{\sum_{s' \mathcal{S}} \vec{w} \cdot \vec{\mathrm{f}}_{s'}}$. Removing the maximizing passage from the set $\mathcal{S}$ to obtain $\mathcal{S}_{\succ}$, the

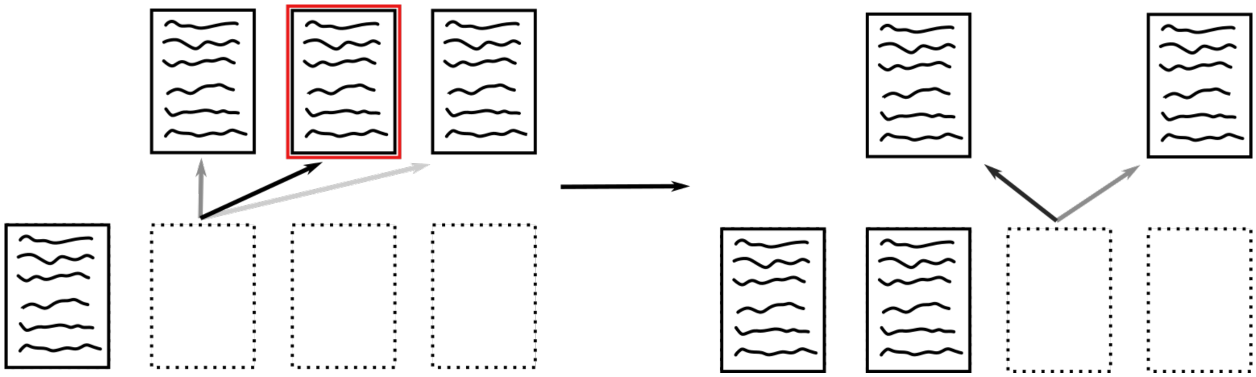

Figure 2.1: The PtrNet method of ordering passages. At each step in the ordering process, a distribution over remaining passages is created. We select the highest likelihood passage to fill the position in the sequence, and continue until all passages are consumed.

following passages are predicted. No further information about the preceding elements is considered by this model. We use the true passage order $\mathcal{S}^*$ for teacher-forcing [36] during training and optimize with a cross-entropy loss. During prediction, we obtain an efficient decoding for the passage ordering via beam search [40].

2.4.3 Overall System

While the ideas discussed above can be combined in different ways, in this paper we study the following system: We pre-train a PtrNet-based passage ordering model $\mathcal{O}_{\text{pre}}(s \in \mathcal{S}_{\succ} | \mathcal{S}_{\prec})$ using end-to-end training on all layers: (1) the PtrNet and (2) the latent representations $\vec{z}$ via graph attention networks to refine a "raw" BERT embedding of passages and the query. The graph $\mathcal{G}$ for the GAT includes the query as well as passages using shared-entity edges.

Fixing the pre-trained model $\mathcal{O}_{\text{pre}}$, we train a combined model end-to-end as

$$\mathcal{O}(s \in \mathcal{S}_{\succ} | \mathcal{S}_{\prec}) = \alpha \mathcal{O}_{\text{pre}}(s \in \mathcal{S}_{\succ} | \mathcal{S}_{\prec}) + (1 - \alpha) \mathcal{O}_{\text{rel}}(s \in \mathcal{S}_{\succ} | \mathcal{S}_{\prec})$$

Here the $\mathcal{O}_{\text{rel}}$ is the regression model to incorporate entity relevance data into passage features. Parameter α is trained along with regression parameters w using the training algorithm described in Section 2.4.2.

2.5 Evaluation

We empirically evaluate our Query-focused Pointer Network in comparison. There are no established IR-benchmarks for this ordering task. While in future our goal is to ordering longer passages, in this paper we focus on sentence-length passages. Despite the simplified setup, we demonstrate significant performance improvements when incorporating relevance.

Data. We create a benchmark, following ideas of automatic benchmark creation from Wikipedia articles [9]: We use the title as a query q, split the lead paragraph into passages s. We reserve the gold order of passages $\mathcal{S}^*$, and shuffle passages for the input set $\mathcal{S}$. Instances with less than three passages are dismissed.

We identify entities in each passage as hyperlinks provided by the Wikipedia editor and identify co-referent entity mentions through exact string matching (after lower-casing) of (1) the hyperlinked entity mention or (2) one of the top five inlink anchors (with at least 20 usages) to that entity's Wikipedia page.

A ranking of relevant entities is produced through unsupervised rank aggregation $\text{score}(e) = \sum_R \frac{1}{rank_R(e)}$ from a large set of rankings R provided with the TREC Complex Answer Retrieval track[1] which includes BM25, Query likelihood, RM3/ECM expansion using different query representations.

Training setup. We use Wikipedia articles provided in TREC CAR data[2] to train the model. We first pre-train the neural network for $\mathcal{O}_{\text{pre}}$ using 768-dimensional `BERT_base_uncased` as-is, on 50,000 Wikipedia lead paragraphs taken from fold 0 of the `train` split. Next, we train the combined model $\mathcal{O}$ (including $\mathcal{O}_{\text{rel}}$) on 117 lead paragraphs of `benchmarkY1-train`, and evaluate our approach on 133 lead paragraphs in `benchmarkY1-test`.

For pretraining $\mathcal{O}_{\text{pre}}$, we use 64 latent dimensions for $\vec{z}$ and $\vec{h}$ yielding a total of 158,144 parameters. Training of $\mathcal{O}(s \in \mathcal{S}_{\succ} | \mathcal{S}_{\prec})$ entails setting α and w (7 parameters). We use Adam

[1] We used `entity-page*` run files from http://trec-car.cs.unh.edu/inputruns/,

[2] Available at http://trec-car.cs.unh.edu/datareleases/

optimizer with learning rate of 0.004.

Evaluation metrics. We define two metrics for evaluating generated orderings:

Kendall's Tau Distance: The number of discordant pairs normalized by the total number of passage pairs (lower is better).

Perfect Match Ratio (PMR): The fraction of perfectly ordered passage sets.

Compared systems. We empirically compare our Query-focused Pointer Network to the variants of our approach where some components are deactivated (as denoted in Table 2.1) and the following baseline:

Sentence-state PtrNet: Replacing our GAT encoder with a Sentence-State LSTM as proposed by Zhang et al. [45].

Pairwise Learn-to-Order: A pair-wise ranking model trained to minimize discordant pairs $s \succ s'$ according to the gold order $\mathcal{S}^*$. Passage representations are generated with the GAT encoder.

2.6 Results

Table 1 reports the main results from experimentation. Our Query-focused PtrNet exhibits a statistically significant improvement over all other systems, and achieves competitive PMR performance.

When considering just the PtrNet and Query-focused PtrNet, we see a statistically significant improvement in Kendall's Tau, indicating that the inclusion of these features leads to a net improvement in sentence ordering performance.

The Pairwise model predicts a partial order of a pair of passages at a time, without considering the order or preceding passages (as in PtrNets). This lack of past consideration leads to worse results, demonstrating the importance of the PtrNet component.

Table 2.1: Evaluation results. ▼: marks significantly lower performance according than ours using a paired t-test significance tests $\alpha = 5\%$. Lower is better for Kendall's Tau distance, higher is better for Perfect Match Ratio (PMR). (S): Special Sentence-State encoder. (B): Uses the BERT representation of query as initial PtrNet state.

Model	Kendall's Tau	PMR	BERT	GAT	PtrNet	Query-Init	Entity-Rel
Pairwise Learn-to-Order	$0.344 \pm 0.016^{\blacktriangledown}$	0.084	X	X			
No Encoder PtrNet	$0.308 \pm 0.015^{\blacktriangledown}$	0.092	X		X	(B)	
No Query PtrNet	$0.300 \pm 0.015^{\blacktriangledown}$	**0.109**	X	X	X		
Sent-State LSTM PtrNet [45]	$0.307 \pm 0.015^{\blacktriangledown}$	0.084	(S)		X	X	
Entity Relevance Only $\mathcal{O}_{\text{rel}}$	$0.524 \pm 0.017^{\blacktriangledown}$	0.034					X
Pretrained PtrNet $\mathcal{O}_{\text{pre}}$	$0.287 \pm 0.014^{\blacktriangledown}$	0.084	X	X	X	X	
Query-focused PtrNet (ours)	$\mathbf{0.275} \pm 0.015 \star$	0.092	X	X	X	X	X

We see improvements in ordering results in methods that use a Graph-Attention encoder, but only in approaches that also contain a PtrNet component. This relationship indicates that the dense representations created by informing passages with information from one another is beneficial in a context where elements are being selected recurrently, but not in contexts where dense representations are directly compared.

Other PtrNets variations all see losses in terms of ordering accuracy. The usage of an encoder, query decoder initialization, and distinct learned parameters for each layer of the encoder work in concert to provide best performance. Removing any component leads to a reduced quality.

When ordering passages based solely on the entity relevance, we see very poor ordering scores, as the model does not capture notions of coherence. While the entity relevance information is necessary to achieve the best performance on ordering, it is not sufficient on its own.

2.7 Conclusion

When considering the passage ordering problem, we see that additional information provided beyond the passages themselves is beneficial. While passage content on its own is sufficient to inform large contextual embedding models, it is often difficult for these models to incorporate more specific information such as the entity-relevance we explore in this work.

Entities serve as an indicator of subject matter in passages. While other ordering models are capable of using the text of an entity to inform their output, they cannot leverage extended information such as the relevance of an entity to the query without an external signal such as what has been proposed in this work.

However, we find that entities on their own are insufficient to inform the ordering task. While they do lead to increased performance when used in conjunction with other passage ordering techniques, when used in isolation they lead to ordering results bordering on random.

We demonstrate statistically significant improvements in passage-ordering results when query-relevant information about passages and entities is incorporated. Specifically, we provide relevance features to fine-tune a pre-trained passage ordering model.

This work points in a new direction for IR when collating several relevant sources to provide a coherent response to inform the user.

CHAPTER 3

Multistage Cluster Summarization for Article Generation

Abstract

Modern Information Retrieval methods are capable of producing high-quality document rankings for a given search query. However, users of a retrieval system are often responsible for the selection and examination of retrieved documents and must read multiple documents to obtain all the information required for a full understanding of a topic. We seek to remove this responsibility from the user by instead constructing a single overarching document that incorporates information from all retrieved documents. Our multi-stage summarization approach avoids potential challenges that arise when summarizing large amounts of text by breaking the summarization problem into several subtasks. We first extract information from source documents, then eliminate redundancies, before finally ordering information in a logical manner. This approach is applied to the TREC-CAR Dataset, on which we demonstrate state-of-the-art results.

3.1 Task Description and Motivation

The TREC-Complex Answer Retrieval (CAR) shared task focuses on the retrieval of passages relevant to a given search query. [10]. We seek to provide an extension of this shared task which serves to produce a single concise, informative article from documents obtained by a retrieval system.

The queries found in the TREC-CAR shared task are often broad topics, such as "Natural

Resources" or "Horseshoe Crabs". Due to the breadth of available information on these topics, the information required for an informative article may be spread out among several more focused documents.

Users of a retrieval system are often responsible for the selection and examination of retrieved documents, and must read multiple documents to obtain all the information required for a full understanding of a topic. We seek to remove this responsibility from the user by instead constructing a single document that incorporates information from all retrieved documents, which will be sufficient to provide the necessary information independently.

This work seeks to provide the following contributions:

- Providing a method for creating concise yet informative summaries from a large amount of input documents.
- Extending prior work in Information Retrieval with a generative task to allow a single document to be presented to a user.
- A detailed analysis of generated articles, with an emphasis on comparing generated articles to a ground truth on a semantic level.

To motivate the experiments we present in our evaluation, we seek to answer two main research questions:

- **RQ1**: Does the usage of a multi-staged approach to summarization lead to better results than state-of-the-art single-stage methods?
- **RQ2**: How does the method of summarization impact the abstractiveness of generated text when compared to the input?

3.1.1 Task Statement

Given a query q and a ranking of documents for this query $\{d_1, d_2, \ldots, d_n\}$, generate an article a that is similar to a reference summary a^*.

We consider established evaluation metrics, such as ROUGE variants [21] and BERTScore [44], as well as perform a manual evaluation of generated text.

3.1.2 Challenges

In order to guide our design decisions and areas of focus for the duration of this work, we identify several potential challenges that our approach should be capable of handling:

Topical Breadth Retrieved documents potentially have large topical breadth, which may be necessary to cover all information required in a Wikipedia-style article. Wikipedia articles are structured into subtopics, represented by the top-level headings in the article. When constructing an article, we would like to ensure that information is grouped by subtopic, leading to the same organizational style as Wikipedia.

Text Quantity Document rankings can contain an arbitrarily large number of documents. The amount of input documents present in this context is significantly larger than what modern multi-document summarization systems are evaluated on: the DUC 2004 dataset contains 50 collections of only 10 documents each [42, 43]. As such, our system will need to be more robust to larger document collections than current state-of-the-art approaches.

Information Ordering When reading an article, it is desirable to have *coherent* text. Several approaches have been made to evaluate the coherence of text [1, 13]. As in our prior work, we consider the ordering of information to be an important consideration for improving document coherence. For this work, we consider information ordering both on a local (sentence) and global (subtopic) level.

3.2 Related Work

3.2.1 Abstractive Summarization

Abstractive summarization is a Natural Language Processing task where a summary of input documents is created by constructing new text with generative modeling. This differs from extractive summarization, in which summaries are produced by extracting fragments of the input document directly [14]. An extension of this task, query-focused abstractive summarization, provides additional context in the form of a query to the summarization model [18].

Modern abstractive summarization models rely on encoder-decoder architectures [31], where an encoder processes input tokens before a decoder produces a new sequence of output tokens [6]. Because these models have a limited vocabulary of tokens with which to generate an output sequence, it is common to provide a means for copying tokens of the input directly into the output sequence [12]. This has shown to be effective for names, places, and other proper nouns which are unlikely to appear in a vocabulary [41]. Following the advent of the Transformer model [37], attention-based models have become more common for summarization [11]. Information ordering is not considered in this task, as it is assumed that the information in the source article is already in a suitable order for coherency [28].

3.2.2 Multi-document Summarization

Multi-document summarization is a well-studied extension of abstractive summarization, where the input text comes from multiple source documents [22]. This summarization task naturally extends further to a retrieval setting, as a set of retrieved documents can serve as a set of input documents for summarization [24]. Modern transformer based summarizers have limitations on the amount of input tokens they can accept at one time [8]. This restriction has prompted alternative methods of performing summarization [27]: graph-based approaches to summarization have been explored by building graphs of input document paragraphs [20] or information triplets [30], which removes the need for neural networks to process large amounts of text simultaneously. These works mostly focus on small document sets of no more than 10 documents, which is significantly smaller than document sets produced by retrieval methods. Further, popular datasets for multi-document summarization often have narrowly defined topics, distinguishing them from the large topical breadth we see in retrieval sets [23].

3.3 Approach

In this section we describe our novel approach to article generation, Multistage Cluster Summarization. Our approach operates on clusters of input documents and consists of three main components: Document-Level Summarization, Redundancy Removal, and Information Ordering.

Modern multi-document summarization approaches have limitations on both number and length of documents that can be processed into a single summary. It is noted that with broad queries, such as "Natural Resources", which encompass a large breadth of information, the retrieved documents can be clustered into subtopical groups [15]. These groups are trained to correlate with the subsections of a Wikipedia-style article, where each subsection is a subtopic of the main article topic. We propose a method of handling large retrieved document sets by first clustering documents, and then considering each subtopic summary independently.

3.3.1 Document-Level Summarization

In many cases, the amount of text present in a cluster exceeds the maximum token limitations of modern NLP summarization models. To alleviate this concern, we propose to summarize each document in the cluster independently, before merging these subsummaries into a single coherent summary of the subtopic. By performing this procedure for each subtopic cluster, and then merging all subtopic summaries, a full article encompassing all relevant subtopics can be generated. We produce document summaries using a neural abstractive summarization model, which have been shown to have state-of-the-art performance [32].

3.3.2 Redundancy Removal

Given the number of documents present in each cluster, it is likely that some documents will contain redundant information. Effectively written articles are informative but also concise: this stands in contention with our combination of individual document summaries. In order to avoid including redundant information in our final subtopic summary, we use a method of eliminating this redundant information before merging all document summaries. First, we group document summaries according to a similarity metric using a clustering algorithm [2, 16]. Then, we concatenate all document summaries within a single group, and apply a second pass of the neural summarization model. This results in a single subsummary for the cluster of redundant information, which we substitute in place of all composing documents in our final subtopic summary.

3.3.3 Information Ordering

Finally, each subtopic summary must have its subsummaries put into a meaningful order for coherency. We achieve this by selecting an order that maximizes a similarity metric between adjacent subsummaries. Two similarity metrics are identified which are potential candidates for this step:

Entity-Adjacency Two subsummaries are considered similar if the set of entities mentioned in both subsummaries have high overlap [17].

Semantic Similarity Two subsummaries are considered similar if they have a high cosine similarity after being embedded by a contextual embedding model.

Our approach addresses the three main challenges in multi-document summarization outlined earlier. We eliminate the need for our summarization model to handle topical

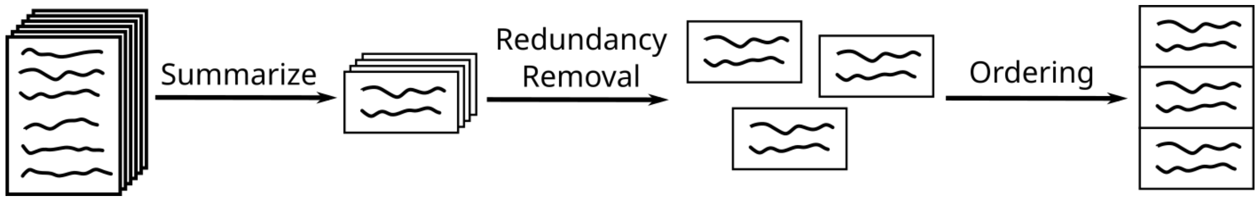

Figure 3.1: The multi-stage document summarization process for a single cluster of documents. We begin by extracting facts from each document via summarization. Then, redundant information is eliminated leaving a reduced set of output statements. Finally, statements are ordered into an output section for use in our final article.

breadth by pre-clustering input documents. The multi-stage summarization approach limits the amount of text provided to the summarization model at one time, avoiding restrictions on the input length present in modern summarizers. Our metrics for subsummary ordering explicitly attend to the information ordering problem after summarizing, something that is not present in other approaches.

3.4 Evaluation Framework

3.4.1 Dataset

We use the TREC-CAR [10] dataset to construct our data. The TREC-CAR dataset includes listings of Wikipedia subheadings and human-annotated relevance scores between the subheadings and passages from other Wikipedia articles. From these relevance scores we produce clusters of relevant passages for each subheading, which we use as the input to our model. This represents both an ideal retrieval system and an ideal clustering system. We create our ground truth from the true Wikipedia text for a given article, also present in the TREC-CAR dataset.

We produce 111 examples consisting of a document set, associated cluster information, and ground truth text. The average document set contains 4605 words, and the largest document set contains 13799 words.

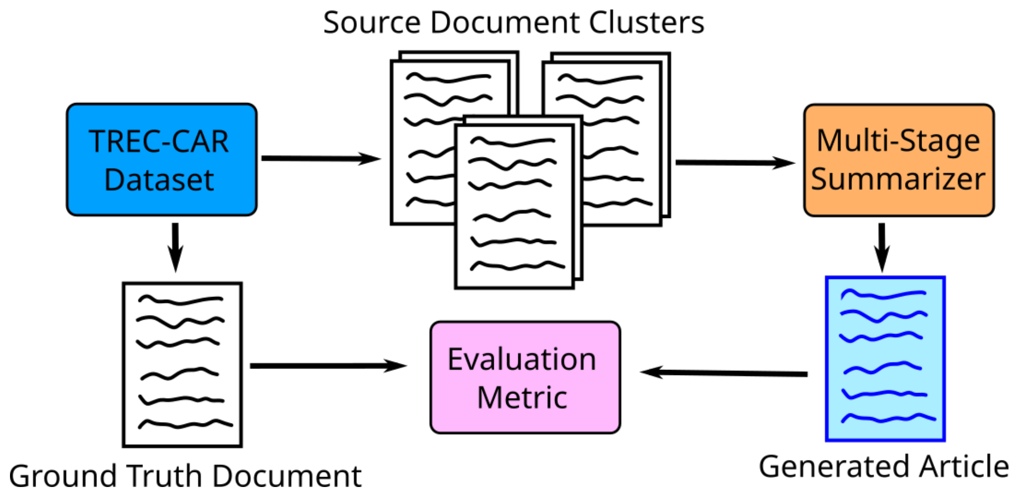

Figure 3.2: The evaluation framework for our model. Given a set of document clusters, our approach produces an output article. Given the ground-truth article from the TREC-CAR dataset, we are able to perform a variety of automatic and manual evaluations on the model.

3.4.2 Evaluation Metrics

In order to provide a complete evaluation of our system compared to other approaches from recent literature, we consider a variety of evaluation metrics, both automatic and manual. A common metric is ROUGE [21], however this metric has been shown to have significant drawbacks [29]. Another more recent metric is BERTScore [44], which calculates precision and recall between a generated summary and a reference summary through the use of a contextual embedding model. We two variants of ROUGE (ROUGE-1 and ROUGE-2) as well as BERTScore as our automatic evaluation metrics.

Beyond automatic metrics, we also perform a manual evaluation of the generated articles. Annotators are presented with a generated article and a list of five "important points" that should be discussed in the article. For each point in the list, they are asked to score the article from 0-3, with a 0 indicating that the generated article does not mention this point and a 3 indicating that the article fully explains the point. We average the score of each point for an article to produce a manual evaluation.

3.4.3 Baseline Methods

We compare our work to other abstractive summarization models from recent literature. Specifically, we compare to the methods Hierarchical Transformer [25] and LoBART [30].

- **Hierarchical Transformer** [25] is an abstractive summarization model which builds a ranking of input documents before proceeding with summarization. Each input document provides its own context, which are used in parallel via a graph-based summarization model.

- **LoBART** [30] takes advantage of pre-trained Transformer models while avoiding length limitations by restricting attention to local spans before doing an explicit content selection step.

In addition to these methods from literature, we also consider two heuristic baseline methods which perform no summarization. The first, **Raw Document**, concatenates all input documents in arbitrary order and presents this is the generated article. The other, **Raw Document Sampled**, randomly samples input documents to create an output document that is approximately equal in size to the summaries produced by other methods.

3.4.4 Our Methods

We seek to evaluate three variations of our model. All variations of our model follow the article generation procedure discussed prior, albeit with slight differences to some components:

Multistage Redundant The second stage of summarization has been disabled, and so there is no elimination of redundant information from the generated text before ordering subsummaries.

Multistage Entity-Order The subsummary ordering process depends on the entity-adjacency metric described above.

Model	ROUGE-1	ROUGE-2	BERTScore
Raw Document	0.163 ± 0.004	0.037 ± 0.002	0.818 ± 0.001
Raw Document Sampled	0.165 ± 0.004	0.027 ± 0.002	0.820 ± 0.001
Hierarchical Transformer [25]	0.074 ± 0.005	0.013 ± 0.001	0.770 ± 0.003
LoBART [30]	0.211 ± 0.005	0.052 ± 0.002	0.818 ± 0.001
Multistage Redundant (Ours)	0.179 ± 0.004	0.035 ± 0.002	0.812 ± 0.001
Multistage Entity-Order (Ours)	0.172 ± 0.003	0.028 ± 0.001	0.813 ± 0.001
Multistage SBERT (Ours)	0.172 ± 0.003	0.028 ± 0.001	0.813 ± 0.001

Model	Manual Evaluation	Avg. Document Length
Raw Document	-	3282 ± 252
Raw Document Sampled	1.44 ± 0.168	558 ± 39
Hierarchical Transformer	-	76 ± 9
LoBART	1.80 ± 0.139	827 ± 19
Multistage Redundant (Ours)	-	1041 ± 75
Multistage Entity-Order (Ours)	-	459 ± 25
Multistage SBERT (Ours)	1.54 ± 0.155	458 ± 26

Table 3.1: Results for ROUGE-1, ROUGE-2, BERTScore, and our Manual Evaluation. In addition, we provide the average generated document length for each method. Due to low automatic scores for the Hierarchical Transformer, we do not perform a manual evaluation of this method.

Multistage SBERT The subsummary ordering process uses SentenceBERT [33] as a contextual embedding model to produce a vector representation of each subsummary, and then optimizes the order to maximize cosine similarity between adjacent subsummary vectors.

3.5 Results

Results are provided in Table 3.1 for the two baseline methods, the two methods from literature, and three variations on our model.

These results serve to answer to our research question **RQ1**, pertaining the performance of our method compared to methods that do not perform a multi-staged summarization process. Both our model and LoBART surpass the results of both baseline methods, however the Hierarchical Transformer does not. Between our methods we see a correlation between longer output documents and higher scores in ROUGE-based metrics. However, this trend does not extend to LoBART, which surpasses our results with shorter documents than our Multistage Redundant model.

3.6 Discussion

3.6.1 Abstractiveness of Models

Even in abstractive summarization, it is common to use a copying mechanism to pull text directly from the source documents [41]. Due to this, it is possible for an abstractive summarization technique to perform similarly to an extractive method, while generating a minimal amount of "new" text. To evaluate how much text a method extracts directly from the source as opposed to generating, we use a similarity calculated using precision from ROUGE-2. A high ROUGE score between the generated text and the input suggests that large amounts of text were copied from the input documents into the final article. As such, we state that a model with high abstractiveness is one that has a low ROUGE score between its output and the input.

$$\text{Abstractiveness}(M) = 1 - ROUGE2(M)$$

ROUGE provides values between 0 and 1 as its output. Inverting this metric provides a measure of how abstractive a model is. This evaluation answers our research question **RQ2**, which focuses on the abstractiveness of other models compared to our multi-stage approach.

Model	Abstractiveness
LoBART	0.019 ± 0.001
Multistage Redundant (Ours)	0.223 ± 0.004
Multistage SBERT (Ours)	0.225 ± 0.004

Table 3.2: Models are evaluated for abstractiveness by comparing the generated text to the input documents. High similarities between the two indicates large amounts of copied text, which correlates with low abstractiveness.

From these results, we see that our approach provides a more abstractive generation than LoBART. Further, we see that when removing redundancies from generated article our model becomes slightly more abstractive.

3.6.2 Provenance of Output Facts

Throughout the entire process of our approach, it is possible to track the full path from a single input document to a statement in the generated text. As such, for each statement in the output article, the document(s) used to produce this statement can be identified. This is important for the verification of information presented by the model: the use of this provenance information can provide insight as to the source of questionable output statements during evaluation. In future work this may allow for identification of conflicting source documents, as well as detection of misinformation or otherwise adversarial documents.

3.7 Conclusion

We have explored a novel method of multi-document summarization that addressed common failure points that arise due to high volumes of text and large topical breadth in inputs. Our method is highly abstractive, and produces articles that are both informative and coherent.

Performing the article generation task via multiple subtasks has led to improvements in the scalability and interpretability of our approach. Our method can perform summarization over arbitrarily large document sets while still maintaining an unbroken trace from input to output, allowing the source of a given output statement to be determined.

This work suggests that explicit information extraction prior to summarization is beneficial for article generation. Beyond the benefit of intepretability, this information extraction step provides stronger assurances regarding the validity of output facts.

CHAPTER 4

Conclusion and Future Work

The goal of this thesis is to identify methods of effectively generating and presenting information to fulfill an information need. An entity-oriented method of information ordering and a robust method of article generation are discussed. Within single-document Information Retrieval systems, these methods provide a means for creating concise, coherent, and focused documents.

Our work in information ordering has led to state-of-the-art results when considering entity-relevance information alongside the standard Pointer Network approach to the ordering task. We find that even small collections of engineered features are sufficient to provide models with the necessary context to improve performance. Our features relate primarily to the relevance of entities to the information need, focusing the ordering process on the specific query of the user.

Our hierarchical approach to article generation allows for simultaneous processing of larger document sets than previous work in the field. Due to token-length limitations of current language models, the full content of a search result is infeasible to consider in its entirety. By breaking this problem into several small components, we are able to leverage state-of-the-art single document summarization models without the possibility of providing excess tokens.

The evaluation of our article generation approach demonstrates high-quality output documents and significantly abstractive text, with our output featuring language not found in the source documents themselves. This stands in contrast to other state-of-the-art methods, which either fail to process the large volume of tokens or generate the vast majority of their output via copying statements from the source text.

There still exists questions regarding query-specificity within the domain of summarization. While we depend on the query to inform the collection of source documents (via an information retrieval system), we do not consider it directly during the summarization process. It is hypothesized that incorporating the query could lead to more focused generated articles. This could potentially lead to an article generation method that more effectively addresses the information need of a user.

Another area that requires additional work is fact verification. While we currently assume source documents retrieved by our information retrieval are factually correct, it is possible that two documents will contradict each other. In this scenario, it is possible for our article generation method to write two contradictory statements. To avoid this, questionable statements should be avoided or alternative sources should be used to resolve the contradiction.

The work presented here addresses common problems in summarization and Natural Language Processing and serves as a step towards better fulfillment of information needs. The ultimate goal of information retrieval is to make information more accessible: we hope that our work is beneficial in the realization of this goal.

LIST OF REFERENCES

[1] Regina Barzilay and Mirella Lapata. Modeling local coherence: An entity-based approach. *Computational Linguistics*, 34(1):1–34, 2008.

[2] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

[3] Danushka Bollegala, Naoaki Okazaki, and Mitsuru Ishizuka. A bottom-up approach to sentence ordering for multi-document summarization. *Information processing & management*, 46(1):89–109, 2010.

[4] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: From pairwise approach to listwise approach. Technical Report MSR-TR-2007-40, April 2007.

[5] Jaime Carbonell and Jade Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 335–336, 1998.

[6] Kyunghyun Cho, Bart van Merriënboer, Çağlar Gulçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, 2014.

[7] Baiyun Cui, Yingming Li, Ming Chen, and Zhongfei Zhang. Deep attentive sentence ordering network. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4340–4349, 2018.

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

[9] Laura Dietz and Jeff Dalton. Humans optional? automatic large-scale test collections for entity, passage, and entity-passage retrieval. *Datenbank-Spektrum*, 20(1):17–28, 2020.

[10] Laura Dietz, Manisha Verma, Filip Radlinski, and Nick Craswell. Trec complex answer retrieval overview. In *TREC*, 2017.

[11] Travis R Goodwin, Max E Savery, and Dina Demner-Fushman. Flight of the pegasus? comparing transformers on few-shot and zero-shot multi-document abstractive summarization. In *Proceedings of COLING. International Conference on Computational Linguistics*, volume 2020, page 5640. NIH Public Access, 2020.

[12] Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640, 2016.

[13] Camille Guinaudeau and Michael Strube. Graph-based local coherence modeling. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 93–103, 2013.

[14] Vishal Gupta and Gurpreet Singh Lehal. A survey of text summarization extractive techniques. *Journal of emerging technologies in web intelligence*, 2(3):258–268, 2010.

[15] Sumanta Kashyapi and Laura Dietz. Learn the big picture: Representation learning for clustering. In *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*, pages 141–151, 2021.

[16] Renaud Lambiotte, J-C Delvenne, and Mauricio Barahona. Laplacian dynamics and multiscale modular structure in networks. *arXiv preprint arXiv:0812.1770*, 2008.

[17] Mirella Lapata and Regina Barzilay. Automatic evaluation of text coherence: Models and representations. In *IJCAI*, 2005.

[18] Md Tahmid Rahman Laskar, Enamul Hoque, and Jimmy Huang. Query focused abstractive summarization via incorporating query relevance and transfer learning with transformer models. In *Advances in Artificial Intelligence: 33rd Canadian Conference on Artificial Intelligence, Canadian AI 2020, Ottawa, ON, Canada, May 13–15, 2020, Proceedings*, pages 342–348, 2020.

[19] Jing Li, Aixin Sun, and Shafiq Joty. Segbot: A generic neural text segmentation model with pointer network. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 4166–4172. International Joint Conferences on Artificial Intelligence Organization, 7 2018.

[20] Wei Li, Xinyan Xiao, Jiachen Liu, Hua Wu, Haifeng Wang, and Junping Du. Leveraging graph to improve abstractive multi-document summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6232–6243, 2020.

[21] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

[22] Chin-Yew Lin and Eduard Hovy. From single to multi-document summarization. In *Proceedings of the 40th annual meeting of the association for computational linguistics*, pages 457–464, 2002.

[23] Kenneth C Litkowski. Summarization experiments in duc 2004. In *Proceedings of the HLT-NAACL Workshop on Automatic Summarization, DUC*, pages 6–7. Citeseer, 2004.

[24] Peter J Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. Generating wikipedia by summarizing long sequences. In *International Conference on Learning Representations*, 2018.

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

[26] Lajanugen Logeswaran, Honglak Lee, and Dragomir Radev. Sentence ordering and coherence modeling using recurrent neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[27] Congbo Ma, Wei Emma Zhang, Mingyu Guo, Hu Wang, and Quan Z Sheng. Multi-document summarization via deep learning techniques: A survey. *arXiv preprint arXiv:2011.04843*, 2020.

[28] Rada Mihalcea. Graph-based ranking algorithms for sentence extraction, applied to text summarization. In *Proceedings of the ACL interactive poster and demonstration sessions*, pages 170–173, 2004.

[29] Jun Ping Ng and Viktoria Abrecht. Better summarization evaluation with word embeddings for rouge. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1925–1930, 2015.

[30] Ramakanth Pasunuru, Mengwen Liu, Mohit Bansal, Sujith Ravi, and Markus Dreyer. Efficiently summarizing text and graph encodings of multi-document clusters. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4768–4779, 2021.

[31] Romain Paulus, Caiming Xiong, and Richard Socher. A deep reinforced model for abstractive summarization. In *International Conference on Learning Representations*, 2018.

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

[33] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, 2019.

[34] Ridho Reinanda, Edgar Meij, Maarten de Rijke, et al. *Knowledge graphs: An information retrieval perspective.* Now Publishers, 2020.

[35] Zhiqing Sun, Jian Tang, Pan Du, Zhi-Hong Deng, and Jian-Yun Nie. Divgraphpointer: A graph pointer network for extracting diverse keyphrases. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 755–764, 2019.

[36] N Toomarian and J Barhen. Fast temporal neural learning using teacher forcing. In *IJCNN-91-Seattle International Joint Conference on Neural Networks*, volume 1, pages 817–822. IEEE, 1991.

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[38] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[39] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. *Advances in neural information processing systems*, 28, 2015.

[40] Sam Wiseman and Alexander M Rush. Sequence-to-sequence learning as beam-search optimization. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1296–1306, 2016.

[41] Song Xu, Haoran Li, Peng Yuan, Youzheng Wu, Xiaodong He, and Bowen Zhou. Self-attention guided copy mechanism for abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1355–1362, Online, July 2020. Association for Computational Linguistics.

[42] Yongjing Yin, Linfeng Song, Jinsong Su, Jiali Zeng, Chulun Zhou, and Jiebo Luo. Graph-based neural sentence ordering. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 5387–5393. International Joint Conferences on Artificial Intelligence Organization, 7 2019.

[43] Jianmin Zhang, Jiwei Tan, and Xiaojun Wan. Adapting neural single-document summarization model for abstractive multi-document summarization: A pilot study. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 381–390, Tilburg University, The Netherlands, November 2018. Association for Computational Linguistics.

[44] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019.

[45] Yue Zhang, Qi Liu, and Linfeng Song. Sentence-state lstm for text representation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 317–327, 2018.

APPENDIX A

Article Generation Demonstration

We present an example article generated by our Multistage Cluster Summarization model. Data provided to the model follows the same format described in section 3.4.1. Article headings are extracted from the ground truth for ease of reading, however all body text is produced solely by the model.

A.1 Natural Resource

A.1.1 Classification

The term natural resource management is often used when dealing with a particular resource for human use rather than managing the whole ecosystem. A main objective of natural resources management is the sustainability for future examples include natural resource management, economic development, uses of traditional knowledge, genetic resources, health care, and education.

A.1.2 Extraction

Industries include establishments that develop the mine site, extract the natural resources, and/or those that process the mineral mined. Trends include sustainable use of natural resources and mineral resource extraction, agro-tourism, production of local food and niagara peninsula wine. The ministry of natural resources and wildlife is responsible for the protection of flora and fauna. It manages natural resource extraction in the province of quebec. Western australia's economy is largely driven by extraction and processing of mineral and petroleum commodities. The structure of the economy is closely linked to these natural resources, providing a comparative advantage. A reserve acts as a buffer zone that keeps ranching and extractive the tapajós-arapiuns extractive reserve was created by federal decree on 16 november 1998. it is classed as iucn protected area category vi the lagoa do jequiá marine extractive reserve was created in 2001. the reserve is administered by the chico mendes institute for biodiversity conservation.

A.1.3 Depletion of Resources

The conservation movement lobbies for protection of endangered species. Major environmental issues may include climate change, pollution, resource depletion etc. Recently, natural gas was discovered. Environmental problems include desertification, salination of fresh water, water-borne disease. Market failure could explain depletion of natural and social capital. He says natural capital is often undervalued by society since we are not fully. The security council is concerned at the actions and policies of former president charles taylor. It stressed the need to return the misappropriated funds and assets to liberia. Some markets can fail due to the nature of the goods being exchanged. This can cause underinvestment because developers cannot capture benefits from success.

A.1.4 Protection

In 2015, puerto rico's department of natural resources received back from the federal government 70 acres around the cove. Nwf acquired enough of the great swamp to protect the massive natural resource. Legislation championed by later secretary of the interior, stewart l. udall, was passed. In mexico, are considered ”protected natural areas” these include 34 biosphere reserves (unaltered ecosystems) 67 national parks. The reserve is administered by the chico mendes institute for biodiversity conservation. Or&r works on a cross-noaa team to protect and restore marine resources. The office conducts natural resource damage assessments by assessing environmental and economic injury. Or& the branford natural resources trust was founded in 1967. Proposition 84 passes 53.8

A.1.5 Management

The ciwm aims to advance the science, technical and practical aspects of wastes and resource management. The aim is to promote education, training, and research in the topic. Private sector's traditional role in environmental resource management is that of the recovery of natural resources. Private sector recovery groups include mining (minerals and petroleum), forestry and fishery organisations. The ministry issues progress reports on protection of the environment once in every three years.