Hierarchical Lay Text Summarization Of Wildfire Impacts for Systemic Engineering Practices

Vedant Sahai
vzs5356@psu.edu
Pennsylvania State University

Sathiyajith Kavindapadi Saravanan
sxk6394@psu.edu
Pennsylvania State University

ABSTRACT

This paper addresses the problem of lay text summaries from research papers related to wildfires and their impacts on the electrical grid infrastructure. Text summarization is a challenging task due to the need to condense large amounts of information while preserving its essential meaning. Despite significant advances in the field, existing techniques often have limitations such as bias, insufficient coverage, or low coherence. In this paper, we propose a hierarchical text summarization approach that combines extractive and abstractive techniques to address these limitations. Our experiments show that our hierarchical approach outperforms existing summarization techniques in terms of summarization quality, diversity, and coherence. The proposed method is capable of capturing a wide range of insightful information from the self-generated domain-specific corpus. Systemic engineering practices, such as strategies for wildfire mitigation can help decision-makers better understand the full scope of the impacts and make more informed decisions about how to address them.

ACM Reference Format:
Vedant Sahai and Sathiyajith Kavindapadi Saravanan. 2023. Hierarchical Lay Text Summarization Of Wildfire Impacts for Systemic Engineering Practices. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Electric utilities spend months preparing their wildfire mitigation strategies before implementing them with systemic operation procedures. Even in such situations, the fundamental goal of electric utilities is to retain as much of their crews' safety and the continuity of their customers' access to electricity. In the wildland-urban interface (WUI) [20] of Santa Clara, California, a wildfire was even simulated that was parametrically comparable to a wildfire that occurred in 2017. Although Pacific Gas and Electric Company [9] serves this region, artificial electrical feeder models that were obtained from public sources were superimposed upon this region and were thought to serve the local consumers. When wildfires were discovered and were moving across the northern counties, the mitigation option that was used as a general disruption of all feeder circuits. The domain-specific text cannot be understood by the mitigators and requires a lay-text summarization that can simplify what the problem is and how it can be handled. By employing text summarization to extract the relevant data, this goal can be further improved. This gives access to historical data and previous mitigation techniques, which can be used to influence current and future tactics.

Text summarization naturally entails transferring a source document's input word sequence to a target summary word sequence. Deep learning-based models that translate one input sequence into another output sequence, have recently demonstrated success in a wide range of challenges, including machine translation [5], speech recognition [4], and video captioning [22]. The Recurrent Neural Network (RNN) encoder-decoder model introduced in [5], which has achieved state-of-the-art performance in machine translation (MT), which is also a natural language job, is a very relevant model to our task within the framework of sequence-to-sequence models. But, The RNNs have the disadvantage of a vanishing gradient problem that occurs when gradients become too small during backpropagation, making it difficult for the network to learn long-term dependencies. LSTM overcomes this problem by incorporating memory cells that can store information over a longer period of time. There are few works using LSTM models [21], [12] and [7] that show that LSTM-based models can outperform traditional RNN-based models in text summarization tasks by producing more accurate and coherent summaries.

The earlier research on text summarization focused on extractive summarization and then the trends gradually shifted towards abstractive methods. Abstractive summarization thoroughly examines input material using NLP approaches to paraphrase important ideas into clear language using neural encoder-decoder architecture. This paper [1] proposes a local attention-based model that creates each word of the summary for each of the corresponding input sentences using a sequence-to-sequence model with an attention mechanism. The model's encoder parses the input text to create a series of hidden states. The decoder then creates the summary word by word while paying attention to the encoder's concealed states. When creating each word of the summary, the attention mechanism enables the decoder to concentrate on various elements of the input text. This model was further improved by [14] in which he replaced the encoder with an RNN encoder that was enhanced with lexical and statistical features such as named entity recognition (NER) and part-of-speech (POS) tags. The extractive models were outperformed by Abstractive models on a well-studied collection of summarization tasks as shown in [2], but limitations include short summaries and required paraphrasing of the sentences.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

The papers [8] and [17] provide an overview of recent advances in text generation using pre-trained language models (PLMs). Pre-trained language models such as BERT [3], PEGASUS [25], and GPT [18] were trained on large text data. These PLMs perform a wide range of language tasks, including language translation, summarization, question answering, and text generation. They have shown significant improvement in results. Most of these models are based on the Transformer architecture [23], which was introduced in 2017 by Google. The Transformer model was designed to tackle NLP tasks that involve transforming an input sequence into an output sequence, and it is based on an encoder-decoder framework that is similar to RNNs. Hugging face [24] library provides many of these pre-trained models to solve a wide range of NLP tasks. The authors highlight the importance of evaluation metrics for assessing the quality of the generated text, such as human evaluation, and automated metrics like ROGUE [11] and BLEU [16].

However Pre-trained PLMs (Language Models) are usually not optimized for specific tasks to achieve State of the Art results. Due to its generalistic approach to training, it fails to take into consideration the finer nuances of the domain-specific tasks, and hence fine-tuning is required. Thus in this paper, we propose a hierarchical approach to summarization that combines extractive summarization with abstractive summarization to produce a more accurate and informative summary of the impacts of wildfires on various aspects of society. This hierarchical approach provides a concise summary without compromising the domain-specific terminologies. The objective of this paper is to provide a tool for systemic engineering practices to aid decision-making processes related to wildfire management and mitigation. By providing a comprehensive and accessible summary of the impacts of wildfires, this approach can help decision-makers better understand the full scope of the impacts and make more informed decisions about how to address them.

2 DATASET

In order to conduct our research effectively, we had to narrow down our study to a specific case scenario of wildfires, as our computational resources were limited. We required an unstructured text corpus that specifically relates to this chosen scenario. To obtain this corpus, we curated a collection of articles from The *International Journal of Wildland Fire*, which is a publication that focuses on both fundamental and practical research related to wildfires. The journal covers a wide range of topics, including the ecological consequences of fire on stands and landscapes, fire modeling, and techniques for controlling fires. Because wildfires have a significant global impact, the journal adopts a global viewpoint in its publications. To extract data from the articles, we used the BeautifulSoup Python module, which is designed to scrape information from XML and HTML web pages. We extracted the abstracts and titles from all published studies that covered key aspects of wildfire occurrence. Our corpus comprised articles published in The International Journal of Wildland Fire from 1991 to 2023, totaling 1977 papers across 29 volumes. We then conducted a preliminary statistical analysis of the corpus to identify frequently occurring keywords and their frequency of recurrence.

Figure 1: Wordcloud

Through the use of EDA techniques, we were able to obtain a useful and relevant corpus for our research, respectively. Words like data, fire, wildfire, vegetation, and fuel occur more frequently as shown in figure 1. Among these 1977 documents, we have document lengths varying from around 5 to a maximum of 20 sentences and an average length of 10 sentences as shown in 2. Each document consists of *100-300* words and an *average of 200* words. There are only very less outliers in words. The count of words is normally distributed, and the count of chars varies from 500-1200 and an average of 1000 words. The distribution of words and characters is close to the mean value and the standard deviation is relatively small. These insights help us in finding what the summary must look like. Overall, the quality of the dataset is quite good with very less outliers and no skewed distribution of the features used.

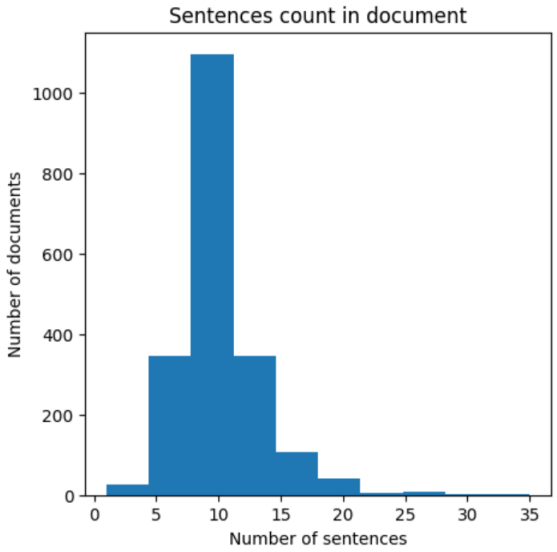

Figure 2: Distribution of count of sentences across documents

3 MODELS

3.1 TextRank

As described in [13], when applying graph-based ranking algorithms to natural language texts, TextRank first creates a representation of the text as a graph, connecting words or other text

elements with appropriate relationships. A vertex is added to the graph for each sentence in the text since the objective of sentence extraction is to rate full sentences. We are defining a "similarity" relation, where "similarity" is quantified as a function of content overlap, to construct connections (edges) between sentences. The relationship between two sentences can be thought of as a process of "recommendation": a sentence that addresses a particular concept in a text "recommends" that the reader refers to other sentences in the text that address the same concept, and so a link can be made between any two such sentences that share common content. The overlap of two sentences can be calculated by counting the common tokens between their lexical representations, or it can be subjected to syntactic filters that only count words belonging to a particular syntactic category. Additionally, we use a normalization factor and divide the content overlap of two sentences by the length of each sentence in order to prevent encouraging long sentences. Si = Wi1, Wi2, ..., WiNi, the similarity of Si and Sj is defined as:

$$Similarity(Si, Sj) = |Wk|WkSiWkSj|log(|Si|) + log(|Sj|)$$

The resulting graph is densely connected, and each edge has a weight that represents the degree to which different sentence pairs in the text are connected to one another. Due to the weighted graph representation of the text, we are employing the weighted graph-based ranking formulas described in Section 2.5. The graph can be shown as one of the following three types of graphs: (a) a straightforward undirected graph; (b) a directed weighted graph with edges set from a sentence to sentences that come after it in the text (directed forward); or (c) a directed weighted graph with edges set from a sentence to sentences that came before it in the text (directed backward). The top-scoring sentences are chosen to be included in the summary after the ranking algorithm has been applied to the graph and sentences are sorted in reverse order of their score. Figure 1 displays an example of text together with the weighted graph that was created for it. The final score computed for each vertex using the PR formula and applied to an undirected graph are also shown in the figure along with sample weights linked to the edges connecting to vertex 93. For inclusion in the abstract, only the sentences with the highest rank are chosen. Sentences with the ids 9, 15, 16, and 18 are retrieved for this sample article, producing a summary of around 100 words that, based on automatic evaluation metrics, comes in second place among summaries created by 15 different systems (see Section 4 for evaluation methodology).

3.2 MemSum

In the research paper [15], they have presented a multi-step episodic Markov Decision Process (MDP) model for extractive summarization. As shown in the 3, at each time step in an episode, they defined a sentence state made up of three sub-states: the local content of the sentence, its overall context within the document, and details about the extraction history, which included the previously chosen set of unordered sentences and the remaining sentences. The current state of the sentences is input into the policy network (agent) at each time step, which then generates scores that are used to decide whether to halt the extraction process or choose one of the remaining phrases to go into the candidate summary. In their multi-step policy, the agent updates the extraction history at each time step before choosing an action, in contrast to one-step episodic MDP-based models that encode the state information just once at the beginning of the episode. This method of state updating allows the agent to choose a sentence while taking the incomplete summary's content into account. They created an extraction agent based on LSTM networks to effectively encode local and global sentence states (Hochreiter and Schmidhuber, 1997). They employed fewer attention layers with relatively low dimensionality to encrypt the extraction history and to pick actions. Due to these decisions, their model can quickly be trained and can sum up lengthy texts like scientific papers or reports. The following are the contributions made by their work: 1) Extractive summarization should be thought of as a multi-step episodic MDP that is aware of the extraction history. 2) Demonstrate how our model behaves more robustly in the presence of document redundancy and can extract more compact summaries than models without historical knowledge. 3) The model outperforms GovReport datasets as well as extractive and abstractive summarization techniques. 4) Human reviewers believe that MemSum summaries are of higher quality than those produced by a competitive approach, particularly because they contain less repetition.

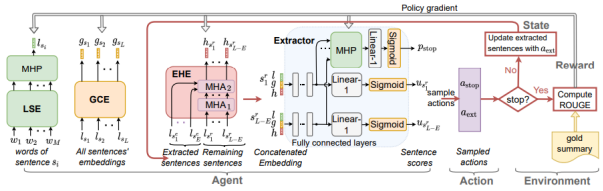

Figure 3: Arhitecture of MemSum as proposed in [6]

4 PRETRAINED LANGUAGE MODELS

4.1 T5

Developed by researchers at Google, T5 [19] is one of the versatile and multi-purpose Transformer models that demonstrated brilliant results across a variety of tasks not limited to Text Summarisation, Question Answering, translation, and Sentiment Analysis to name a few. The idea behind this model was to test the transfer worthiness of Transformers for various downstream NLP tasks; which methods work the best in what setting and for what tasks. Fundamentally, T5 adopts a unified framework and a mindset of generalized text-to-text approach and treats everything as a text generation problem rather than individual specific tasks. This unified framework and approach of input-output being only text (unlike BERT, where the input is text and the output is a class label), enabled the usage of a single model for any NLP task. This model was trained on the C4 dataset, which was double the magnitude of the Wikipedia dataset in the order of 750 GB. This feast of linguistic data endows T5 with a profound understanding of diverse language patterns, enabling it to grasp intricate syntactic and semantic nuances effortlessly.

Transformers have emerged as a powerful paradigm in NLP, enabling efficient parallel processing and capturing long-range dependencies in text. By employing transformers, T5 can effectively

model complex relationships between words or tokens, facilitating accurate and contextually rich text generation. This architecture has proven to be particularly beneficial for tasks that require understanding and generating coherent and contextually appropriate outputs, such as text summarization and machine translation. Another key aspect of the T5 model is its ability to handle different input modalities beyond traditional text. While originally developed for text-based tasks, T5 has shown promising results in multi-modal applications, such as image captioning and visual question answering. By incorporating visual information alongside textual inputs, T5 leverages its powerful pre-training and fine-tuning mechanisms to generate meaningful and contextually relevant outputs, effectively bridging the gap between language and vision.

By leveraging its extensive pre-training on a massive corpus of diverse text, T5 acquires a deep understanding of language patterns and semantic structures, enabling it to generate high-quality abstractions. During fine-tuning, T5 is trained specifically for abstractive summarization tasks using datasets that provide source texts and corresponding human-generated summaries. This process allows T5 to learn how to generate summaries that capture the key information while maintaining readability and coherence. The versatility of T5 also allows it to handle various document lengths and adapt to different domains and languages, making it a highly flexible solution for abstractive summarization.

4.2 Facebook BART

BART [10] harnesses the potential of deep learning to excel in various language-based tasks. It does so by pre-training a large neural network model using a novel methodology called denoising. This technique involves corrupting the input text and training the model to reconstruct the original, uncorrupted text. By learning to restore the original text, BART gains an unparalleled understanding of the underlying patterns and structures, enabling it to generate coherent and contextually accurate responses. BART's architecture shares similarities with BERT but incorporates two key differences. Firstly, each layer of the decoder in BART performs cross-attention over the final hidden layer of the encoder, similar to the transformer sequence-to-sequence model. Secondly, BERT includes an extra feed-forward network before word prediction, which is absent in BART.

BART's training methodology involves two key steps: (1) introducing textual corruption through an arbitrary noising function, and (2) training a model to accurately reconstruct the original text. This approach, employed in BART's pre-training phase, forms the foundation for its impressive language generation, translation, and comprehension capabilities. By corrupting the input text and subsequently training the model to restore the original content, BART develops a deep understanding of language structures, enabling it to produce coherent and contextually accurate outputs. Its ability to capture key information, identify important concepts, and retain the essence of the original text makes BART an invaluable tool for automating the summarization process. Whether it's news articles, research papers, or long-form documents, BART's fine-tuning potential combined with its language comprehension prowess empowers researchers and practitioners to develop highly efficient and accurate text summarization systems, streamlining information extraction and providing users with succinct and meaningful summaries of complex textual content.

4.3 Google Pegasus

Built upon the transformer-based framework, PEGASUS [26] introduces a two-step pre-training process. In the first step, it employs an extractive pre-training approach, where gap sentences are extracted from the source documents to create cloze-style training instances. These instances, along with the original documents, are then used for pre-training a transformer-based model. In the second step, the pre-trained model is fine-tuned using a denoising autoencoder objective, enabling it to generate coherent and accurate abstractive summaries.

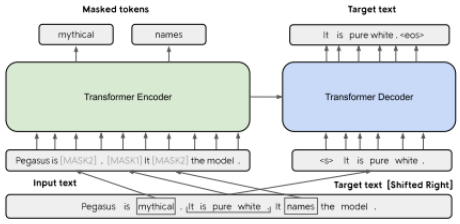

Figure 4: Architecture of Google Pegasus as proposed in [26]

PEGASUS's architecture as shown in Figure 4 capitalizes on both the strengths of extractive and abstractive methods, leading to enhanced summarization performance and a more comprehensive understanding of the source documents. Through its advanced transformer-based architecture, PEGASUS excels at understanding contextual relationships and generating coherent and fluent summaries that preserve key information. PEGASUS's abstractive summarization capabilities have proven effective across diverse datasets and domains, making it a promising tool for automating the summarization process and assisting users in extracting crucial insights from voluminous text documents.

Its effectiveness is evident in its impressive performance across various benchmarks and evaluation metrics. It consistently outperforms existing state-of-the-art models on popular summarization datasets like CNN/DailyMail and New York Times, demonstrating its ability to generate high-quality abstractive summaries. PEGASUS also exhibits superior performance in terms of content selection, coherence, and readability when compared to other abstractive summarization techniques.

5 TRAINING PARAMETERS

The Microsoft Azure and Google Colab GPU resources were used to train all of the models discussed in this work. It took less than 4 hours for each epoch for the pre-trained LLMs model and extractive models to run. The Wildfire dataset's training set has approximately 2k data points. Using the Huggingface API, all pre-trained models were loaded and trained using the criteria listed in Table 1 and 2. The full-length Rouge F1 metric that we used for the Wildfire

corpus was used to evaluate all of our models, with one important exception: in both the system and gold summary, we treated each highlight as a separate sentence.

Table 1: Training Parameters used for Abstractive Pretrained Models

Parameter	Value
Epochs	5
Batch Size	8
Learning Rate	$5.6e-5$
Weight Decay	0.01

Table 2: Training Parameters used for Extractive Models

PARAMETER	TEXTRANK	MEMSUM
Epochs	N/A	5
Batch Size	N/A	2
Stop Threshold	N/A	0.6
Weight Decay	N/A	0.999
Max Sequence length	N/A	100
Max Article Size	N/A	500
Max Summary Size	N/A	7
No of Documents	1700	1700

6 RESULTS

In our report, we utilized the standard ROUGE metric to evaluate the performance of our models. Specifically, we measured the F1 scores for three different ROUGE variations: ROUGE-1, ROUGE-2, and ROUGE-L. ROUGE-1 measures the overlap of individual words between the reference summary and the summary being evaluated. ROUGE-2, on the other hand, looks at the overlap of bigrams (pairs of consecutive words) between the two summaries. Finally, ROUGE-L measures the longest common sequence between the reference summary and the evaluated summary.

To obtain these measurements, we used the Huggingface wrappers around Google Research's implementation of ROUGE, as suggested in the literature by [11]. This allowed us to accurately and efficiently calculate the F1 scores for each ROUGE variation. Overall, our evaluation using the standard ROUGE metric provided valuable insights into the performance of our models, and we were able to draw meaningful conclusions based on the F1 scores obtained for each ROUGE variation.

6.1 Quantitative Analysis

The results of all fine-tuned models on the Wildfire dataset are shown in Table 3 and 4. From Table 3, we notice that for the extractive tasks the MemSum extractive model outperforms the TextRank algorithm because it successfully captures the semantic meaning, uses memory networks to retain key information, and chooses key sentences from the source document rather than creating new sentences.

Table 3: ROGUE Scores of Extractive Methods

		ROGUE 1	ROGUE 2	ROGUE L
MemSum	**Training**	63.90	55.23	63.90
	Testing	82.3129	74.6575	80.9523
TextRank	**Training**	76.4705	67.6056	74.5098
	Testing	11.6788	21.978	11.6788

As shown in Table 4, we notice that the BART outperforms the T5 and Google Pegasus transformer models especially when dealing with tasks that require the use of noisy or unstructured input data. like ours. The reason for this could be attributed to how closely connected BART's pre-training goals are with the challenge of producing high-quality text from noisy input. BART is pre-trained using objectives such as denoising autoencoder and masked language modeling, which make it well-suited for tasks where the input data is noisy or unstructured. In contrast, T5 and Pegasus were pre-trained using different objectives, such as a combination of language modeling and retrieval-based objectives, which may not be as effective for tasks involving noisy or unstructured input data.

Table 4: ROGUE Scores of Abstractive Methods

		ROGUE 1	ROGUE 2	ROGUE L
T5-Base	**Training**	46.0082	40.6726	44.8385
	Testing	44.0493	37.8972	42.373
Facebook BART	**Training**	51.2453	48.9532	51.1414
	Testing	48.3244	44.8785	47.6833
Google PEGASUS	**Training**	54.2501	41.3914	48.3438
	Testing	53.6497	40.6219	47.5999

Furthermore, the idea of hierarchical learning was in an effort to enhance the performance of our text-summarizing qualitatively and quantitatively. In order to achieve this, we chose the model that performed the best across all of the various approaches—one abstractive and one extractive. We examined the hierarchical approach after choosing the models, comparing its outcomes to those of the improved models. The outcomes demonstrated in Table 5 show that the hierarchical approach was able to nearly match the ROGUE scores of the SOTA fine-tuned models. However, on further analysis, we believe that the reason it wasn't able to overcome the quantitative results of the individual methods is maybe that when applying the extractive method on the original document, due to its shorter length, the summary only captures the key sentences but loses a lot on the context part and thus when the abstractive is applied in the hierarchy, the final summary generated isn't comprehensive enough and only contains the keywords in it. We believe merging the numerous documents might help the approach in extracting the keywords as well as retaining the context further helping the abstractive summarizer to generate better quantitative results.

Table 5: Comparison of ROGUE Scores

Model	ROGUE-1	ROGUE-2	ROGUE-L
Extractive (MemSum)	82.3129	74.6575	80.9523
Abstractive (BART)	48.3244	44.8785	47.6833
Hierarchical (BART + MemSum)	49.15	39.24	43.33

6.2 Qualitative Analysis

While quantitative metrics such as ROUGE scores from Table 3 and 4, suggest that extractive models outperformed abstractive models for text summarization, a qualitative analysis of the summaries revealed some limitations of both methods. The extractive summary tended to use technical language, which may be difficult for the average reader to understand, while the abstractive summary sometimes missed important information. To overcome these limitations, a combination of both methods was found to be effective. By using both extractive and abstractive methods, the resulting summary retained important information while also using more accessible language that could be understood by field engineers in their day-to-day work. This hybrid approach shows promise for improving the effectiveness of text summarization for a variety of applications. The same can also be seen in the original and predicated summaries below.

Original Document: *The simultaneous occurrence of wildfire can hinder firefighting effectiveness via multiple mechanisms that might explain historical resource demand trends. We validate this hypothesis by using data from the Monitoring Trends in Burn Severity (MTBS) project to determine if simultaneous wildfire occurrence is correlated with preparedness levels and examine potential changes in simultaneous wildfire activity over the 1984â€“2015 record. We explore patterns that are helpful for predicting simultaneous wildfires, such as seasonal variability in simultaneity, cross-regional correlations, and models of simultaneous wildfire occurrence based on dryness and lightning indicators. We show that simultaneous wildfire is at least as correlated with preparedness levels as other burned area measures and identify changes in simultaneous wildfire occurrence within the western and southern United States. Seasonal variation and spatial autocorrelation in simultaneous wildfire occurrence provide evidence of coupling of wildfire activity in portions of the western United States. Best-approximating models of simultaneity suggest that high levels of simultaneous wildfire often coincided with low fuel moisture and high levels of lightning occurrence. Model uncertainty was high in some contexts but, with only a few exceptions, there was strong evidence that the best model should include both a dryness and lightning indicator*

Original Summary: *This study investigates the correlation between simultaneous wildfire occurrence and firefighting effectiveness, using data from the Monitoring Trends in Burn Severity (MTBS) project. The results show that simultaneous wildfire is strongly correlated with preparedness levels and that there have been changes in their occurrence over the past few decades in the western and southern United States. The study also identifies patterns that can help predict simultaneous wildfires, such as seasonal variability and the relationship between dryness and lightning indicators. The best models of simultaneity suggest that high levels of simultaneous wildfire are associated with low fuel moisture and high levels of lightning occurrence.*

Extractive Summary: *The simultaneous occurrence of wildfire can hinder firefighting effectiveness via multiple mechanisms that might explain historical resource demand trends We explore patterns that are helpful for predicting simultaneous wildfires, such as seasonal variability in simultaneity, cross-regional correlations, and models of simultaneous wildfire occurrence based on dryness and lightning indicators respectively*

Abstractive Summary: *The simultaneous occurrence of wildfire can hinder firefighting effectiveness via multiple mechanisms that might explain historical resource demand trends. We explore patterns that are helpful for predicting simultaneous wildfires. We show that simultaneous wildfire is at least as correlated with preparedness levels as other burned area measures. We identify changes in simultaneous wildfire occurrence within the western and southern United States.*

Hierarchical Summary: *The simultaneous occurrence of wildfire can hinder firefighting effectiveness via multiple mechanisms that might explain historical resource demand trends. We validate this hypothesis by using data from the Monitoring Trends in Burn Severity (MTBS) project to determine if simultaneous wildfire occurrence is correlated with preparedness levels and examine potential changes in simultaneous wildfire activity over the 1984â€“2015 record.*

7 CHALLENGES

We faced significant challenges in developing the corpus for the electrical and wildfire-related domains. The primary obstacle was the lack of a suitable corpus, which made it exceedingly difficult to develop an effective summarization-based algorithm. As a result, we first needed to curate a corpus specific to these domains, which proved to be a challenging task in itself. The corpus we curated was non-homogeneous and specific to these domains, which presented further difficulties for our summarization-based algorithm. We found that domain-specific summarization tasks are particularly challenging for these algorithms to handle, given the nuanced and complex nature of the information involved.

Another major challenge was Choosing the right model for the task at hand is one of the primary issues in text summarization. It can be challenging to choose the model that will perform the best for a given dataset due to the abundance of models that are accessible. When choosing a model, it's crucial to take the type of input data, the desired output, and the available resources into account. For instance, using a basic sequence-to-sequence model or a simpler model like TextRank instead of a more complicated model like BERT or GPT may be preferable if the dataset is tiny. Dealing with resource limitations, particularly in terms of time and computer capacity, is another difficulty. Large language models like BERT or GPT can take a lot of time and require a lot of computational power to train. In such cases, it becomes necessary to use pre-trained models or to fine-tune a smaller model to achieve the desired level of performance.

8 CONCLUSION & FUTURE WORK

In this paper, we try to propose a novel domain-specific hierarchical lay text summarization pipeline by leveraging the transfer learning approach on the pre-trained Large Language Models. To conduct our research on the summarization of wildfires, we used a corpus of literature that we curated ourselves. This allowed us to carefully select and control the content of the corpus to ensure it was relevant to our study. We wanted to ensure that the corpus was representative of the topic of wildfires and contained a variety of information that could be used to generate informative summaries. As we analyzed the summaries, we paid close attention to the accuracy and quality of the summaries produced by our models. We were particularly concerned with ensuring that the summaries were semantically meaningful and accurately reflected the content of the corpus. We carefully tried to examine each summary to ensure that it conveyed the intended meaning and was consistent with the information presented in the corresponding text.

Despite the benefits of using text summarization, there are some potential drawbacks to be aware of. One such challenge is ensuring that the generated headlines are unbiased and factually accurate. This can be difficult to achieve depending on the quality of the training data used to develop the model. If the training data is not of high quality, then the generated headlines may not meet the desired standard. Another issue we encountered in our study is that the baseline results we obtained fell short of the state-of-the-art (SOTA). To address this, we plan to make improvements to the model architecture by either adding more layers or adjusting the existing layer hyperparameters. By doing so, we hope to maximize the performance of the model. In addition to improving the model architecture, we also plan to use combine different models and evaluate overall performance by running comprehensive robustness testing on a bigger dataset to determine how generalizable the workflow is.

In conclusion, the quality and correctness of the summaries produced by the hierarchical method, as well as the method we used to choose and analyze the corpus, contributed to assuring the validity and trustworthiness of our research findings.

REFERENCES

[1] Jason Weston Alexander M. Rush, Sumit Chopra. 2021. A Neural Attention Model for Abstractive Sentence Summarization. *arXiv preprint arXiv:1509.00685* (2021).

[2] Richard Schwartz Bonnie Dorr, David Zajic. 2021. Hedge Trimmer: A Parse-and-Trim Approach to Headline Generation. *arXiv preprint arXiv:1509.00685* (2021).

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[4] Dmitriy Serdyuk Philemon Brakel Yoshua Bengio Dzmitry Bahdanau, Jan Chorowski. 2021. End-to-end attentionbased large vocabulary speech recognition. *arXiv preprint arXiv:1509.00685* (2021).

[5] Kyunghyun Cho Dzmitry Bahdanau and Yoshua Bengio. 2021. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1509.00685* (2021).

[6] Nianlong Gu, Elliott Ash, and Richard H. R. Hahnloser. 2021. MemSum: Extractive Summarization of Long Documents using Multi-step Episodic Markov Decision Processes. *CoRR* abs/2107.08929 (2021). arXiv:2107.08929 https://arxiv.org/abs/2107.08929

[7] Dan Jurafsky Jiwei Li, Minh-Thang Luong. 2021. A Hierarchical Neural Autoencoder for Paragraphs and Documents. *arXiv preprint arXiv:1509.00685* (2021).

[8] Wayne Xin Zhao Junyi Li1, Tianyi Tang and Ji-Rong Wen1. 2021. Pretrained Language Models for Text Generation: A Survey. *arXiv preprint arXiv:1509.00685* (2021).

[9] Samarjeet Kaur, Vedant Sahai, Aditi Jaiswal, and Sayonsom Chanda. 2020. Knowledge Mining for Defining Systemic Engineering Practices. In *2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*. IEEE, 1346–1352.

[10] 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).

[11] Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*. Association for Computational Linguistics, Barcelona, Spain, 74–81. https://aclanthology.org/W04-1013

[12] Yang Liu. 2021. Fine-tune BERT for Extractive Summarization. *arXiv preprint arXiv:1509.00685* (2021).

[13] Rada Mihalcea. 2021. Graph-based Ranking Algorithms for Sentence Extraction, Applied to Text Summarization. *arXiv preprint arXiv:1509.00685* (2021).

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

[15] Richard H.R. Hahnloser Nianlong Gu, Elliott Ash. 2021. MemSum: Extractive Summarization of Long Documents Using Multi-Step Episodic Markov Decision Processes. *arXiv preprint arXiv:1509.00685* (2021).

[16] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics* (Philadelphia, Pennsylvania) *(ACL '02)*. Association for Computational Linguistics, USA, 311–318. https://doi.org/10.3115/1073083.1073135

[17] Diyah Puspitaningrum. 2015. A Survey of Recent Abstract Summarization Techniques. *arXiv preprint arXiv:1509.00685* (2015).

[18] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training. (2018).

[19] 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. *CoRR* abs/1910.10683 (2019). arXiv:1910.10683 http://arxiv.org/abs/1910.10683

[20] Engel D. Chen Y. Thompson S. Chassin D. Pratt R. Schneider, K. and J. Fuller. 2021. https://item.bettergrids.org/handle/1001/180. *arXiv preprint arXiv:1509.00685* (2021).

[21] Junyang Lin Xuancheng Ren Shuming Ma, Xu Sun. 2021. A Hierarchical End-to-End Model for Jointly Improving Text Summarization and Sentiment Classification. *arXiv preprint arXiv:1509.00685* (2021).

[22] Jeff Donahue Raymond J. Mooney Trevor Darrell Subhashini Venugopalan, Marcus Rohrbach and Kate Saenko. 2021. Sequence to sequence - video to text. *arXiv preprint arXiv:1509.00685* (2021).

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

[24] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771* (2019).

[25] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*. PMLR, 11328–11339.

[26] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020. PEGASUS: Pre-Training with Extracted Gap-Sentences for Abstractive Summarization. In *Proceedings of the 37th International Conference on Machine Learning (ICML'20)*. JMLR.org, Article 1051, 12 pages.