“Brevity is the soul of w(r)it”

Cheng Ding Xiang
A0232303Y, School of Computing
National University of Singapore

Ee Wen Lennard, Ong
A0034832L, School of Computing
National University of Singapore

Krishna Murthy Kannan
A0236178X, School of Computing
National University of Singapore

Kshitij Singh
A0248368N, School of Computing
National University of Singapore

***Abstract*—Text summarization is a crucial and challenging task in natural language. Creating a concise, coherent abstract that retains all relevant content is the natural end goal of this task. We consider abstractive summarization as two independent subtasks, knowledge extraction and text generation, to leverage the strength of large language models while retaining control and interpretability over generated output. Knowledge is extracted in the form of subject-verb-order (SVO) triples which are then used to generate natural language sentences via a transformer. We also evaluate our approach on metrics for text simplification for readability.**

I. Introduction

Condensing a vast body of text into a short collection of key points allows for improved comprehension. It enhances readability, especially for non-native speakers of a language. In the modern world, where data is big and information is aplenty, it is much more imperative to distil stuff down. Text summarization, formally the process of creating a concise and compact abstract for a given body, is thus a core task in natural language processing.

Extractive summarization seeks to rank the sentences in the given body of text and produces a collection of the most important sentences as the summary. While the methods vary in how ranking is done, this domain is mainly focused on understanding what constitutes a key sentence and relies on the inherent text structure for its generated output. *Abstractive summarization* meanwhile, adds the component of *text generation* to the problem of summarization. It relies on creating a semantic representation of the input text and generating output based on the most important sentences, no longer beholden to the original structure.

In this project, we explore a novel approach by dividing the task of summarization into two sub-tasks, namely *knowledge extraction* and *text generation*. We first seek to extract key information from a given body of text in the form of subject-verb-object (SVO) triplets and later create a natural language description for these triples. Treating it as a set of independent sub-tasks allows us to leverage strengths of both non-neural and neural methods, train specialised modules as well as interpret and explain our observations.

II. Related Work

TextRank [17] is a graph-based approach to extract key sentences from a given body of text. It creates a graph with each sentence assigned to a unique node, edges labeled with content overlap between sentences and feeds the graph into the PageRank [20] algorithm to generate importance scores for each sentence. LexRank [5] works on a similar sentence-based graph structure, employing eigenvector centrality to compute the relevant scores. Gong and Liu [8] posit an application of singular value decomposition to a term-sentence matrix, thus deriving the latent semantic structure of a body of text. They propose summarization by using the right singular vectors as a representative of importance scores for each topic in a sentence.

Since abstractive summarization requires an inherent semantic representation of language, neural networks based on sequential learning have been employed with reasonable success [3]. With the advent of the transformer architecture [31], large language models have received a widespread surge in adaptation. OpenAI proposed an extension with Generative Pre-trained Transformer (GPT) [23] and its subsequent iterations by fine-tuning the learned model parameters by training on downstream tasks. Google proposed Bidirection Encoder Representations from Transformers (BERT) [4] by pre-training the bidirectional transformer on a masked-language model objective. More recently, a Text-To-Text Transfer Transformer (T5) [24] was introduced as a unified framework that works solely with text strings.

PEGASUS [35] approaches abstractive summarization by pre-training a transformer for this task instead of fine-tuning an existing one. A self-supervised objective is created by masking important sentences and teaching the model to generate them from existing sentences. BART [13] employs a denoising autoencoder during its pre-training. It randomly shuffles the order of original sentences, coupled with replacing spans of text with a mask token, and is trained to recreate original text. While relatively state-of-the-art models for the task of summarization, BART and PEGASUS have since been superceded by SimCLS [16] which formulates summarization as a contrastive learning task.

A formal framework for triplets was created by World Wide

Web Consortium (W3C) in the form of Resource Description Framework (RDF) [18] for the purpose of metadata exchange between graphs. Since this creates a structured text representation, it can serve as input for text generation models. While rule-based approaches to transform such knowledge bases into text exist, they are inherently limited by either domain or structure [12] [34].

Vougiouklis et al. [32] propose an encoder-decoder framework to generate textual summaries from RDF triples. Inspired by the Sequence-to-Sequence framework [29], they employ a feedforward neural network to encode triples and pass it through a LSTM-based decoder to generate text. To bridge the gap between structured input and unstructured output, Li et al. [14] propose a novel anchor-to-prototype framework. They use multi-head attention to encode the triple representation, then retrieve prototypes to extract writing patterns and create output similar to those patterns using a LSTM-based decoder.

Wang et al. [33] introduce WikiGraphs, a set of Wikipedia articles paired with their corresponding knowledge graphs to facilitate the research in conditional text generation, graph generation and graph representation learning. This new dataset is notable for its significant scale compared to previous works. They accompany this with a baseline graph neural network and transformer models to perform text generation and text retrieval tasks that take graphs as input.

Shardlow [28] presents a survey of lexical simplification techniques aimed at replacing complex words in a sentence with simpler alternatives without changing its meaning. Since unsupervised techniques focus on the word, regardless of the surrounding context, they tend to yield spurious replacement candidates. LSBert [22] builds upon the advantages of representational model BERT and makes use of the wider context by looking at the whole sentence.

III. Method

In our study, we propose a two-step approach to abstractive summarization. Our core hypothesis is that the semantic essence of a text can be represented through a series of interconnected and interdependent SVO triplets, thus creating a graph of the information contained within the piece of text. Using these SVO triplets, we generate text to create the resultant summary.

Fig. 1. Overview of our approach to summarization

During the *triplet extraction* phase, we employ rule-based heuristics and dependency tree parsing methods to extract SVO triples from an article. This allows us to create a graphical representation of the relationships present in the given body of text, which serves as our foundation knowledge base.

We then proceed to the *triple-to-text generation* phase, wherein we recombine the extracted triplets to generate an abstractive summary for the article. We employ a finely-tuned large-scale language model(LLM) to generate coherent and grammatically correct sentences.

By decoupling the summarization process into two independent phases, our approach aims to balance strengths of both heuristics-based extractive summarization and neural abstractive summarization techniques. The first step allows us to have better control over the output and remain faithful to the original text. Meanwhile the second step allows for more natural flowing and coherent language, which can serve to lessen the language complexity of the original text. This should result in a palatable concise output.

A. Pre-processing

We work with the BBC News Summary [10] dataset as the corpus for our study due to its well-formatted and simple nature, with model handcrafted summaries to evaluate against. This dataset was created by re-purposing the BBC News dataset [9] originally created for benchmarks in classification. Since the language is quite formal and follows strict grammatical structure, we do not require explicit cleanup for individual tokens.

We perform tokenization, stopword removal, stemming and lemmatization on the available dataset. Additionally, we include a step for *co-reference resolution* across the entire corpus. This enhances the granularity of information within each individual sentence, thus ensuring accurate and meaningful dependency capture across different SVO triplets.

B. Triplet Extraction

Fig. 2. NER supplemented with title-based keyword enrichment

We hypothesize that the title of an article serves as an index for meaningful terms that should be part of a summary. As such, we extract keywords from the article title, and use them to enrich the article body tokens generated during *named-entity recognition* (NER). These entity tags will later be utilized

during our SVO generation process, thus strengthening the connection between the generated triplets and the overall theme of the article.

Our triplet structure is inspired by the WebNLG corpus [7] and utilizes the RDF schema. This allows us to use the WebNLG corpus as a source of training data to fine-tune our language generation model downstream.

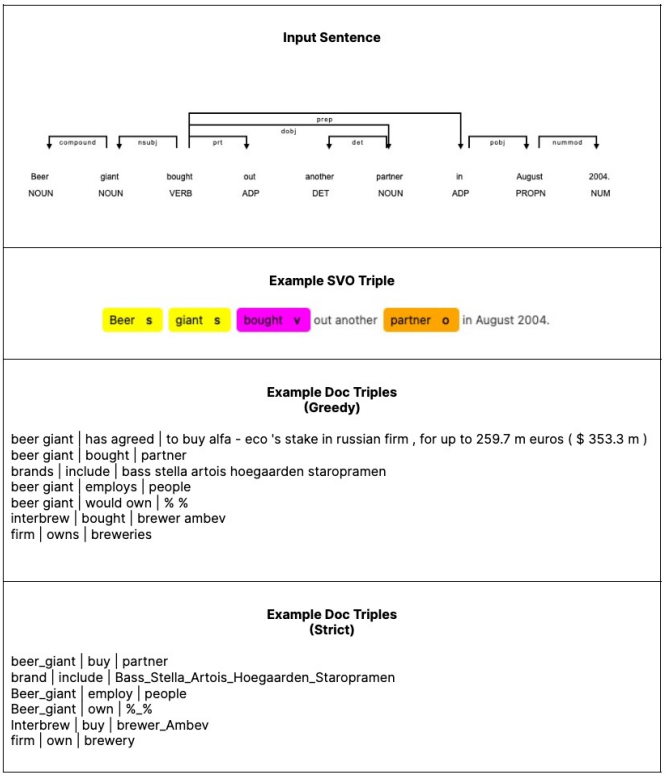

Fig. 3. SVO triplets, *greedy* and *strict* variants

We analyze the dependency structure of the input text and apply a set of heuristics grounded in linguistic rules to generate our SVO triplets. We utilize the Python library *Textacy*. An outline of the involved steps follows.

1) *Sentence* We iterate over the input at a sentence level.
2) *Verb-Subject-Object (Verb-SOS)* We initialize a dictionary, with verbs as its key and sets of associated subjects and objects as the corresponding values.
3) *Token* We parse sentences at a token level, analyzing each token based on its dependency label and part-of-speech tag.
4) *Subject or Object?* The dependency label determines whether a token is a subject or an object and adds it to the Verb-SOS dictionary if so. Subjects can be nouns or subordinate clauses. Objects can be nouns, introduced by a preposition or subordinate clauses.
5) *Subject/Object Relations* We expand subjects and objects to encompass related tokens, such as compound nouns, conjuncts, or tokens within a clause utilizing each token's subtree.
6) *Verb Conjuncts* We check for instances where multiple verbs might be linked via a conjunction (as an example, consider the sentence "I read and wrote this book"), updating the dictionary accordingly since such verbs share subjects and objects.
7) *SVO Triplet* Retrieve the extracted triplets from the Verb-SOS dictionary, each triplet a tuple in the order of subject-verb-object.

We developed two versions of triplet extraction, namely *greedy* and *strict*. These versions differ in the steps 5 and 7 in the above procedure.

The *greedy* variant follows token chains more liberally during subject and object expansion in step 5. This generates longer SVO triplets, thus giving rise to a more contextually rich set of triples. We further relax step 7 to allow non-entity and non-verb tokens in the subject and verb positions.

The *strict* variant is restricted to what might be considered the 'essential' triples. We limit token chains to a maximum size of 5 tokens, and remove auxiliary verbs. We naturally do not allow for non-entity and non-verb tokens in step 7, with the exception of subjects and objects that were a part of the title. We then proceed to lemmatize the non-entities.

C. Triple-to-text Generation

We now proceed to the task of converting our extracted set of triples into a coherent natural language summary. Since we are interested in an abstractive technique, we utilize language models here.

Text-To-Text Transfer Transformer (T5) We use the T5 model introduced by Raffel et al. [24]. This is a unified model that allows for natural language inputs and outputs. This model is pre-trained on Colossal Clean Crawled Corpus (C4) and shows remarkable performance across a range of natural language tasks.

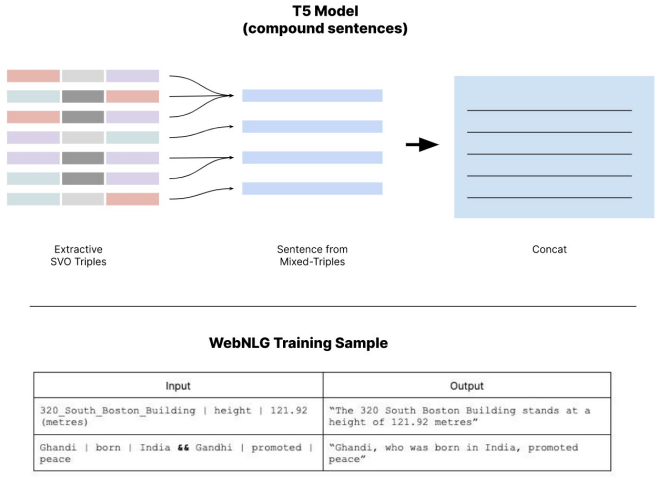

Fig. 4. Fine-tuned T5 generative process with sample training data from WebNLG

Since tuples are limited in structure, related information is often spread across different triplets. A coherent free-flowing

sentence can lead to multiple triplets, thus it is difficult to reconstruct a single sentence from a single triplet. As such, we fine-tune our T5 model by training it on the WebNLG corpus [7] which teaches it to combine sets of related tuples into natural language sentences. Since our *strict* variant of SVO tuples matches the format of RDF tuples in the WebNLG corpus, this fine-tuning suffices for transfer learning and allows us to utilize this model to create natural language text.

We group triplets on subject, and employ a probabilistic function to generate a varying range of 0 to 3 triplet sets that can serve as an input to our fine-tuned model. We believe this process allows the model to capture complex relationships and dependencies between entities in a structured and coherent manner.

DistilGPT2 Faced with constraints of small dataset and limited computational resources, we choose the lightweight model trained with the supervision of 124 million parameter version of GPT-2. Knowledge distillation [26] was utilized to develop this model.

Since DistilGPT2 provides a balance between performance and efficiency, we use it as a potential model to convert our generated triples to text. We use our *greedy* variant of generated triples for this model since they are most contextually rich in information. We perform a $4:1$ split of training and test for our corpus, thus fine-tuning this model.

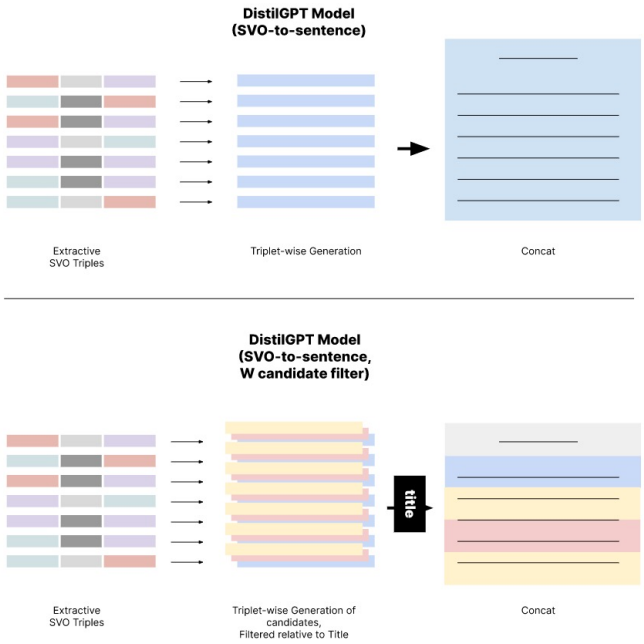

Fig. 5. Fine-tuned DistilGPT2 generative process

Since our current set of triplets is contextually rich, we are able to use isolated triplets to generate coherent sentences in this approach. To mitigate the hallucinatory aspects of the pre-trained model, which tends to introduce contextually incoherent or irrelevant information during text generation, we perform a filtering step by creating a set of candidate sentences for each triplet and selecting the one with the highest cosine similarity with the article's title vector. We believe that this step should allow for generation of summaries with contextual relevance to the title and avoid extraneous details.

Generative Pre-trained Transformer (GPT-3.5) Building upon the success of GPT-3 as introduced by Brown et al. [1], GPT-3.5 is an iterative series of models that were trained on a mixture of text and code. $gpt-3.5-turbo-0301$ is optimized for chat and available as an application programming interface (API).

```
=========Input Sys Prompt============
You are a triplet-to-paragraph generator.

#Brief:
Anything between [] is the task inputs.
# indicates the article title
"sub | veb | obj" describes a series of verb, object triples related to title.

#Task:
The task is to generate a short summary paragraph with the title at top.
It should be factually based only on the triples.
Inferences should only be made between the triples and the title.
Do not add embellishments.
Do organize the paragraph so it has a logical flow.
Keep it as simple and direct as possible.

#Example Input:
[#China now top trader with Japan

china | overtook | us
change | highlights | chinagrowing importance
trade | was hurt | factors
analysts | see | spurs
Japan trade surplus | grew | trade
Japan trade surplus | accounted | trade

#Example Output:
China now top trader with Japan

China has overtaken the US as Japan's top trading partner.
This change highlights China's growing importance in the region.
Trade was hurt by various factors, but analysts see this as a spur to
further growth. Japan's trade surplus grew as a result, with the surplus
accounting for a significant portion of the trade.

=========Input User Prompt============
[#UK 'risks breaking golden rule'

Treasury_spokesperson | dismiss | claim
UK | borrow | cash
UK | finance | UK_spending_project
UK | want | avoid_break_golden_rule
anything_world_economy | continue | grow_strongly
euro_zone | expect | pick_up_speed
export | stage | recovery
growth | set | accelerate
product | tip | 4.1_%_this_year
surge | have | effect
taxis | need | rise_by_about_£_10bn]
```

Fig. 6. Sample input prompt for GPT-3.5 Turbo

We note that text generation is a highly difficult task wherein the natural language sentence created from an input triple is often greater than the sum of its individual components. In essence, the role of context is crucial in generating coherent summaries. Thus, we perform an additional experiment by utilizing the readily available API to convert our generated triples into text.

We invoke the model to create coherent sentences based on input triples. We provide two sets of instructions, *system prompts* which describe the problem at hand and provide the model with a representative example, and *user prompts* which

contain the input data to process and generate a response for.

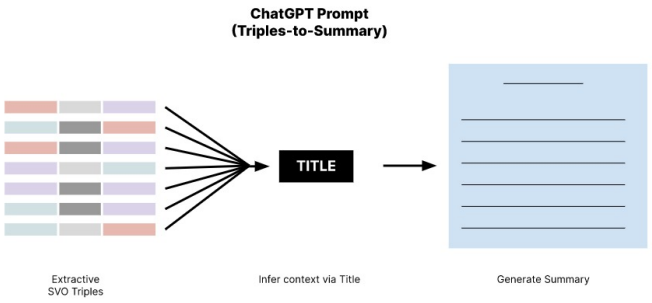

Fig. 7. GPT-3.5 Turbo generative process

D. Baseline

We compare our models against an unsupervised summarizer. For the baseline, we utilize an approach based on LexRank. Sentence-BERT [25] allows us to compute sentence-level embeddings. With sentences as the nodes in our graph, we compute pair-wise cosine similarities among these embeddings and finally pick the top-k sentences based on their degree centrality scores. The final output is comprised of most important sentences present in the original text.

Readability is one of our core motivations in this project. Creating coherent output by extracting the key information present in a text allows us to present a concise representation. However, we also wish to assess the syntactic simplicity of our output. Thus we also explore the task of *text simplification* by comparing the readability of our models against two baseline models.

Our first baseline simplification model is implemented in two steps, find *complex* words in a document and replace them with less *complex* words of an equivalent meaning. A word is considered *complex* if it is infrequently used, thus we use the Zipf scale developed by van Heuven et al. [30] to find appropriate *complex* words. These words are then searched for in the lexical database WordNet [19] and we replace them with similar but more frequently used words.

LSBert [22] provides us with our second baseline simplification model. First, we find *complex* words on our corpus using the Zipf scale. These words are then replaced with masked tokens and BERT [4] is asked to provide us with top-k suitable replacements. We choose an appropriate replacement by scoring them against the Zipf scale.

IV. Evaluation

For the task of *text summarization*, we evaluate our models against reference summaries. We additionally explore *text simplification* metrics by compraing the complexity of generated output by our models and two baseline models.

A. Text Summarization

Automated evaluation of textual summaries is a challenging task. Due to the free-flowing and expressive nature of natural language, there are no particular right and wrong answers. Subjectively, as long as a summary conveys the key points in a body of text while being coherent and concise, it is acceptable. However, creating quantitative metrics to capture *content*, *coherence* and *conciseness* is non-trivial. Fabbri et al. [6] spotlight these challenges in a survey on current automated metrics for text summarization.

ROUGE [15], often considered the standard for summary evaluation, measures the lexical content overlap between generated and reference summaries. ROUGE-N compares how many n-grams match between the two sets of summaries and can be extended to include multiple reference summaries. We use ROUGE-1 as our default metric.

TABLE I
ROUGE-1 scores across models

Model	*Precision*	*Recall*	*F1*
LexRank	**0.764**	**0.699**	**0.702**
T5	0.428	0.229	0.290
DistilGPT2	0.460	0.156	0.179
GPT-3.5	0.485	0.300	0.359

ROUGE-N is a recall-oriented metric. Intuitively, the recall component measures how many n-grams from reference summaries are present in the generated summaries. The precision component measures how many n-grams from generated summaries are present in reference summaries. And the F1 score is a combined quantitative measure. ROUGE-N is good to capture content however it fails to adequately measure conciseness and does not address coherence.

ROUGE-L is a variant that tries to introduce a measure of sentence coherence by capturing the match of longest common subsequence between generated and reference summaries. The matches need not be consecutive, however they are required to be in sequence. This helps mitigate issues in ROUGE-N evaluation whereby presence of n-grams in a grammatically incorrect order is sufficient to yield a high score. Furthermore, ROUGE-N suffers from semantic incoherence whereby switching the subject and object would not change the score even though it changes the meaning of that sentence. As such, we also evaluate our models against ROUGE-L to test for textual coherence.

TABLE II
ROUGE-L scores across models

Model	*Precision*	*Recall*	*F1*
LexRank	**0.521**	**0.480**	**0.480**
T5	0.428	0.229	0.290
DistilGPT2	0.342	0.088	0.106
GPT-3.5	0.253	0.158	0.188

ROUGE requires explicit token matches between output and reference summaries which would disadvantage abstractive neural summarizers which might paraphrase a sentence during text generation. Thus, we utilize another metric that captures token similarity using contextual embeddings. BERTScore [36] is a greedy evaluator that tries to maximize cosine

similarity between token embeddings generated by BERT. Since it does not rely on exact token matches, it is shown to correlate better with human judgements and the authors posit it is a strong indicator for model selection.

TABLE III
BERTSCORE FOR DIFFERENT MODELS

Model	*Precision*	*Recall*	*F1*
LexRank	**0.927**	**0.921**	**0.924**
T5	0.821	0.809	0.815
DistilGPT2	0.853	0.809	0.830
GPT-3.5	0.873	0.842	0.857

Bleu [21] is a precision-oriented metric that is also considered as a standard evaluation metric. It captures how many tokens in the generated output are also present in the reference. Since this could be abused by unnaturally short outputs consisting of most frequent tokens, this metric also assigns a multiplicative *brevity penalty* to outputs shorter than the reference. Since this exact match approach suffers from similar issues as ROUGE-N, moreover we believe that *brevity penalty* is an unintuitive and unfair metric for the task of summarization, thus we refrain from using Bleu as a metric for our study.

B. Text Simplification

We use a *simplicity index* which incorporates three distinct readability metrics, carefully chosen since they assess vital aspects of readability for non-native speakers. Higher scores indicate a more readable text. We apply this index to our model outputs to evaluate the readability of generated output, benchmarking it against original text and reference summaries.

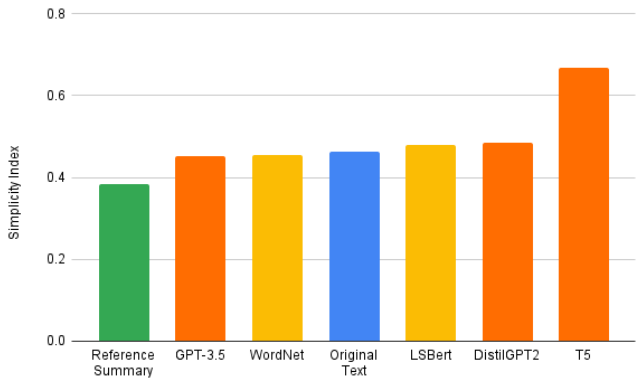

Fig. 8. *Simplicity Index* scores for different article versions

The metrics used in our index are as follows.

1) **Number of syllables** We calculate the Flesch Reading Ease Score [11], a metric based on the average number of syllables per word and the average number of words per sentence.
2) **Complexity of vocabulary** We opt for the Dale-Chall Readability Formula [2], a metric based on the number of 'difficult' words and the average sentence length. A word is considered 'difficult' if it can not be found on a list of 3000 common English words.
3) **Readability** We utilize the Automated Readability Index (ARI) [27], a metric that estimates readability based on the number of characters per word and words per sentence.

We also present a sample evaluation of the index across a representative set of articles from different spheres. We observe an intuitive correlation between the topics and our index values to reason that it is a suitable metric for evaluation.

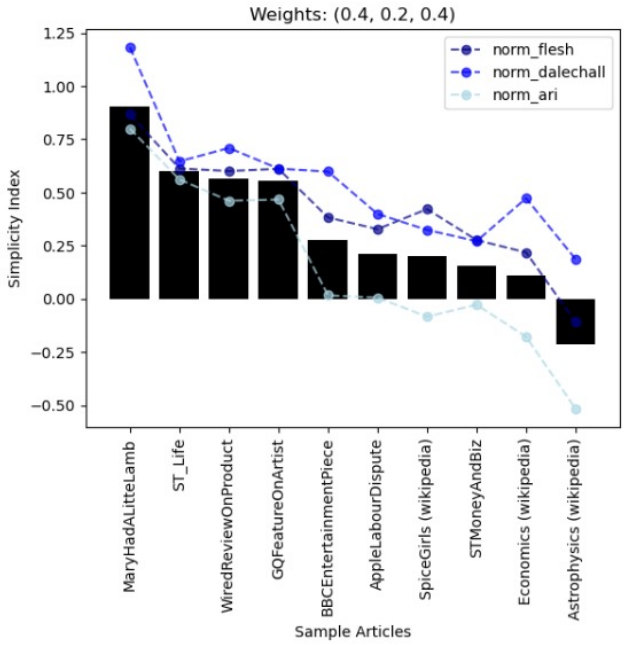

Fig. 9. Sample evaluation of *Simplicity Index* across various topics

V. DISCUSSION

We observe that our text summarization models are unable to match the unsupervised approach of LexRank. There is a distinct gap between the two approaches on ROUGE metrics, however they are much closer in performance on BERTScore. LexRank notably outperforms all three models across all evaluated metrics.

Extractive Summarization With its simplicity and ease-of-use, supplemented with its higher performance, we posit LexRank is a suitable baseline for models to evaluate against. We note that it requires a hyperparameter since we need to specify how many important sentences we need to extract. This can potentially lead to repetitive text if this parameter is set too large, or loss of information if it is too small. There is no principled way to choose this parameter and it is better to err on the side of caution by choosing more text to mitigate against loss of contextual information. We would once again like to highlight the inadequacy of ROUGE metric for the

task of *text summarization*, by noting that it tends to favour extractive approaches since it prefers exact n-gram matches.

Triplet Extraction Our project was aimed at creating a natural language summary by using key information extracted from text. While our output is often coherent, we notice it suffers from a lack of content. BERTScore evaluation indicates that the content of our output is worse than the extractive approach followed by LexRank. We are faced with the complex nature of language, which renders triplet extraction a challenging task. We observed that our heuristics did not transfer across articles, since different authors have different writing styles thus pigeonholing them into strict triples turned out to be unfeasible. Furthermore, strictly adhering to parsing the dependency tree resulted in loss of salient information like numbers, time and cause-effect. A loss of information at this step has downstream effects since the resultant summary lacks important context.

Transfer Learning We would like to highlight the challenges with transfer learning in the context of natural language tasks. T5 was trained on WebNLG dataset, however it was unable to bridge the gap between RDF and SVO schemas. We believed a generic pre-trained transformer would be able to specialize in this domain after fine-tuning, however we were faced with unsatisfactory results. DistilGPT was fine-tuned on a portion of our dataset, however it was not enough to create highly accurate sentences. Since GPT-2 is trained for text prediction, we believe we were unable to fine-tune it with enough datapoints to specialize to our task and a significant chunk of the output was inaccurate. GPT-3.5 was quite effective in the task of summarization and our instructive prompt was sufficient to serve as a *one-shot learning* instance. We observe that it achieved closest scores to LexRank on BERTScore metric, although it does perform poorly on ROUGE. We believe this highlights the strength and versatility of GPT-3.5 and the corpus used for its training which allows it to generalize to many subtasks at a reasonable level.

Hallucinations A significant challenge in *text summarizaiton* is to restrict hallucinatory effects of neural models. GPT-2 for example, is trained to predict words. While it is a suitable aide for auto-completion, speech recognition and generative language tasks, domain transfer to text summarization without suitable representative data presents a hurdle when it makes up words to preserve the coherence of a sentence. The same issues were encountered with T5 which had a larger dataset to fine-tune itself, however it still created outputs interlaced with made-up facts. Such hallucinations are highly undesirable for this task since our goal is to represent a certain set of information concisely, while the focus of these models tends to be fluency of language.

Text Simplification We observe that T5 presents the most readable summaries as reflected by our *simplicity index* scores. We reflect on its fine-tuning process which teaches the model to create succinct representative sentences from RDF triples, which carries over effectively to our task of summarization. We note that our baseline models do not perform well for simplification with WordNet proving inadequate in providing enough replacements for *complex* words. LSBert mitigates this by providing a richer candidate set of replacement words, thus generating text which scores higher on our index. Reference summaries were found to have the lowest scores in general, which conforms with our expectations since summaries are bereft of extraneous sentences and should contain only pertinent information.

Our source code is available at this Google Drive.

VI. Future Work

The road to effective text summarization is long and perilous, with several avenues to improve upon. Extractive approaches provide reasonable outputs and might be considered a baseline to improve upon in the future. However, they lack language fluency and flow. They might also require supervision to ensure lack of information is mitigated.

Abstractive approaches provide an exciting challenge to improve the state-of-the-art. The importance of creating a suitable dataset for training a model should not be underestimated. Models like GPT-3.5 present an interesting avenue for synthetic data generation to create more representative datasets in the future. Additionally, we would like to highlight the benefits of pre-training a model specialized in text summarization as opposed to fine-tuning an existing language model, since the task of summarization is sufficiently distinct from other natural language tasks.

Hallucinations serve as a major hurdle in current abstractive approaches to summarization which would further highlight issues of safety and ethics associated with artificial intelligence. While extractive approaches can provably guarantee accuracy of information, the same cannot be said for abstractive approaches. Language models trained on text prediction are biased w.r.t. training data and they can be unreliable if the output is not sufficiently filtered or monitored.

We would like to bring to note the inadequacy of existing evaluation metrics for text summarization. Quantitative handcrafted metrics like ROUGE fail to capture the diversity of language and penalize deviations in writing style. Meanwhile, recent neural metrics like BERTScore try to measure meaning but are not sufficiently interpretive. A task is only as good as the metric used to evaluate it, and we believe more research is required to create a representative evaluation metric for text summarization.

References

[1] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *CoRR*, abs/2005.14165, 2020.

[2] J.S. Chall and E. Dale. *Readability Revisited: The New Dale-Chall Readability Formula*. Brookline Books, 1995. ISBN 9781571290120.

[3] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078, 2014.

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

[5] Günes Erkan and Dragomir R Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479, 2004.

[6] Alexander R. Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. SummEval: Re-evaluating Summarization Evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409, 04 2021. ISSN 2307-387X.

[7] Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. Creating training corpora for nlg micro-planning. In *55th annual meeting of the Association for Computational Linguistics (ACL)*, 2017.

[8] Yihong Gong and Xin Liu. Generic text summarization using relevance measure and latent semantic analysis. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '01, page 19–25, New York, NY, USA, 2001. Association for Computing Machinery. ISBN 1581133316. doi: 10.1145/383952.383955.

[9] Derek Greene and Pádraig Cunningham. Practical solutions to the problem of diagonal dominance in kernel document clustering. In *Proceedings of the 23rd international conference on Machine learning*, pages 377–384, 2006.

[10] Anushka Gupta, Diksha Chugh, and Rahul Katarya. Automated news summarization using transformers. In *Sustainable Advanced Computing: Select Proceedings of ICSAC 2021*, pages 249–259. Springer, 2022.

[11] J Peter Kincaid, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Technical report, Naval Technical Training Command Millington TN Research Branch, 1975.

[12] Karen Kukich. Design of a knowledge-based report generator. In *21st Annual Meeting of the Association for Computational Linguistics*, pages 145–150, 1983.

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

[14] Ziran Li, Zibo Lin, Ning Ding, Hai-Tao Zheng, and Ying Shen. Triple-to-text generation with an anchor-to-prototype framework. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 3780–3786, 2021.

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

[16] Yixin Liu and Pengfei Liu. Simcls: A simple framework for contrastive learning of abstractive summarization. *arXiv preprint arXiv:2106.01890*, 2021.

[17] Rada Mihalcea and Paul Tarau. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411, 2004.

[18] Eric Miller. An introduction to the resource description framework. *Bulletin of the American Society for Information Science and Technology*, 25(1):15–19, 1998. doi: https://doi.org/10.1002/bult.105.

[19] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[20] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bring order to the web. Technical report, technical report, Stanford University, 1998.

[21] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[22] Jipeng Qiang, Ting Liu, and Yue Zhang. Lsbert: A simple framework for lexical simplification. *arXiv preprint arXiv:2006.14939*, 2020.

[23] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding with unsupervised learning. 2018.

[24] 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. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

[25] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

[26] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In *NeurIPS EMC^{2}Workshop*, 2019.

[27] RJ Senter and Edgar A Smith. Automated readability index. Technical report, Cincinnati Univ OH, 1967.

[28] Matthew Shardlow. View of a survey on lexical simplification. *Journal of Artificial Intelligence Research*, 65: 1–43, 2019.

[29] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Advances in*

neural information processing systems, 27, 2014.

[30] Walter J.B. van Heuven, Paweł Mandera, and Emmanuel Keuleers. Subtlex-uk: A new and improved word frequency database for british english. *Behavior research methods*, 44(3):Lexical, 2012.

[31] 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.

[32] Pavlos Vougiouklis, Hady Elsahar, Lucie-Aimée Kaffee, Christophe Gravier, Frédérique Laforest, Jonathon Hare, and Elena Simperl. Neural wikipedian: Generating textual summaries from knowledge base triples. *Journal of Web Semantics*, 52-53:1–15, 2018. ISSN 1570-8268. doi: https://doi.org/10.1016/j.websem.2018.07.002.

[33] Luyu Wang, Dongxu Zhang, Tingwen Liu, Yukun Huang, Chengyuan Wang, Xiang Zhang, and Jiawei Han. Wikigraphs: A wikipedia text-knowledge graph paired dataset. *arXiv preprint arXiv:2107.09556*, 2021.

[34] Sam Wiseman, Stuart M Shieber, and Alexander M Rush. Challenges in data-to-document generation. *arXiv preprint arXiv:1707.08052*, 2017.

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

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

VII. Appendix

We provide a sample of our outputs below. Following the title and the body of text, we showcase our extracted triplets and the generated summary.

A. (Sample 502) Algeria hit by further gas riots

Original Text Algeria suffered a weekend of violent protests against government plans to raise gas prices, local press reports.

Demonstrators in a number of regions blocked roads, attacked public buildings and overturned vehicles, newspapers including El Watan reported. The price of butane gas, a vital fuel for cooking, has risen to 200 dinars $2.77 per canister from 170 dinars. Even before the hike, failing economic conditions had been fanning resentment in some of Algeria's poorest regions.

Demonstrators took to the streets last week when the cost change was first announced, but police seemed to have restored order. According to local press reports, trouble flared up again on Saturday and carried on into Sunday. El Watan said that a number of hot spots centred on the villages and towns close to Bouira, about 100 kilometres (60 miles) south of the capital Algiers. Among the other main areas affected were the western Tiaret region and Sidi Ammar in the east of the country, Agence France Presse (AFP) reported. Riots also flared up in the Maghnia region close to the border with Morocco in the west, AFP said. Butane gas and fuel oil are used as the main source of fuel to heat homes and cook food in Algeria's remote mountain areas.

Extracted Triplets

```
Algeria | suffer | weekend
demonstrator | block | road
demonstrator | attack | building_vehicle
condition | fan | resentment
demonstrator | take | restore_order
police | seem | restore_order
El_Watan | say | number
butane_gas_fuel_oil | use | food
butane_gas_fuel_oil | cook | food
```

T5 The celestial body known as Algeria has a length of 45 days. The number of the ingredients of El Whitan, a company where El Wan uses, is food. The food found in the UK is the use of the butanegasfuel. The resentment of a building in use is resentment. The building of the 2160-floored Apollo Bellator is located in the country of the. The order of origin of advantage of a building of the Buddha is restore order. The backup order of the Polician politicatic body is the restoration order of which, is located in the rest of order.

DistilGPT2 The company claims that the cheap fuel used by the Iraqi army and its Iraqi government are now used to fuel the country's thirst for oil. The company also said it was halving its production to 2,000 vehicles, and halved its output to 5,500 vehicles.

GPT-3.5 Over the weekend, Algeria suffered from violent gas riots, according to local newspaper El Watan. A large number of people who use butane gas fuel oil to cook food were affected by the shortage, which has fueled resentment among the population. Demonstrators attacked a building and blocked roads, with some taking matters into their own hands to restore order. Eventually, police were able to help restore order.

LexRank + WordNet simplification Algeria suffered a weekend of violent protests against government plans to raise gas prices, local press reports. Riots also flare up in the Maghnia region close to the border with Morocco in the west, afp said. Even before the hike, failing economic conditions had been fan resentment in some of Algeria's hapless regions. Demonstrator in a number of regions blocked roads, attacked public buildings and overturn vehicles, newspapers including El Watan reported. Butane gas and fuel oil are used as the main source of fuel to heat homes and cook food in Algeria's remote mountain areas.

LexRank + LSBert simplification Algeria suffered a weekend of violent protests against government plans to raise gas prices, local press reports. Riots have flared in in the gao region close to the border with Morocco in the west, AFP said. Even before the hike, failing economic development was been fanning resentment in some of Algeria's poorest regions. Demonstrators in a number of regions blocked roads, attacked public building and overturned vehicles, newspapers including El Watan reported. Butane gas and fuel water is used as the main source of fuel to heat homes and cook food in Algeria's remote mountains.

B. (Sample 510) Lacroix label bought by US firm

Original Text Luxury goods group LVMH has sold its loss-making Christian Lacroix clothing label to a US investment group.

The Paris-based firm has been shedding non-core businesses and focusing on its most profitable brands including Moet & Chandon champagne and Louis Vuitton. LVMH said the French designer's haute couture and ready-to-wear labels had been purchased by the Falic Group for an unspecified sum. The Falic Group bought two cosmetics labels from LVMH in 2003. The sale of the Lacroix label comes as many fashion houses are struggling to make money from their expensive haute couture ranges. The Florida-based Falic group, which also runs a chain of 90 duty free stores in the US, said it planned to expand the brand by opening new stores. Mr Lacroix said he planned to stay at the label he founded in 1987 although exact details are still to be confirmed.

Extracted Triplets

```
group_LVMH | sell | Lacroix_label
group_LVMH | shed | business
couture_label | purchase | US_investment_group
US_investment_group | bought | cosmetic_label
```

T5 The USInvestment Group was bought on the cosmetic label. The US is the location of the Coustial body known as the US investment. Lacroix is a seller of the group LVMH which is a company based in the business of the Lacroixland.

DistilGPT2 LG Card has also sold its own LVMH label to LG Group, which has a stake in LG. The deal was sealed in October 2004 after it emerged that cosmetics giant Dove and Dove were both bought by US investment and group bought. The LSE's investment and group planned to increase Lacrox label in 2003 by expanding its new headquarters in London, and by closing new factories. The pair have been linked with the possibility of a possible takeover, but have yet to make an announcement.

GPT-3.5 The Lacroix label, a renowned couture brand, has been purchased by a US investment group. The sale was made after LVMH, the parent company of the Lacroix label, shed the business as part of a divestment strategy. The US investment group's purchase of the cosmetic label is a coup for the company and marks a significant change in ownership for the iconic French brand.

LexRank + WordNet simplification LVMH said the French designer's haute couture and ready-to-wear labels had been purchased by the Falic Group for an unspecified sum. The Falic Group bought two cosmetics labels from LVMH in 2003. The sale of the Lacroix label comes as many fashion houses are struggling to make money from their expensive haute couture ranges. Luxury goods group LVMH has sold its loss-making Christian Lacroix clothing label to a US investment group. The Florida-based Falic group, which also runs a chain of 90 duty free stores in the US, said it planned to expand the brand by opening new stores.

LexRank + LSBert simplification LVMH said the French designer's haute look and ready-to-wear companies have been purchased by the Falic Group for an unspecified sum. The Falic Group had two cosmetics labels from LVMH in 2003. The sale of the house also comes as many fashion houses are struggling to make money from their expensive haute couture ranges. Luxury goods group guess has sold its loss-making Christian green clothing label to a US investment group. The Florida-based Falic group, which also runs a chain of 90 tax free stores in the US, said it planned to expand the brand by opening new stores.