Final Paper: Text Summarization

Scott Nelson, Will Judy, Ari Kanevsky

Abstract

In this study, we evaluate the capabilities of extractive and abstractive text summarizers, including the latest GPT-4 model. We compare the performance of three models: Latent Semantic Analysis (LSA; extractive), T5 (abstractive), and GPT-3.5-turbo (abstractive). We hypothesize that LSA will perform the worst, T5 will achieve moderate performance, and GPT-3.5-turbo will exhibit the best performance. Additionally, we expect T5 and GPT-3.5-turbo to have more similar outputs, as they are both abstractive models. Our comparison uses ROUGE scores to assess the similarity between the outputs of the three models on both simple and complex input texts. Our findings show that T5 and GPT-3.5-turbo outputs are more similar than LSA and GPT-3.5-turbo outputs, and the performance ranking is as follows (best first): GPT-3.5-turbo, T5, LSA. We also observe that the complexity of the input text affects the similarity between the model outputs. For simpler texts, LSA and T5 outputs are more similar, while for more complex texts, T5 and GPT-3.5-turbo outputs exhibit greater similarity. We conclude with suggestions for future work, specifically investigating the performance of GPT-4 compared to these models.

1.0 Introduction

Text summarization is an interesting problem in Natural Language Processing, and requires the models to have a sufficient understanding of the text to convey the overall meanings, ideas, and themes of the text in an accurate, concise way. We propose an analysis of three different methods for text summarization: Latent Semantic Analysis (LSA), Text to Text Transfer Transformer (T5), and OpenAI's General Pre-Trained Transformer 3.5-turbo (GPT-3.5-turbo). First, the Latent Semantic Analysis works by initially constructing features of a TF-IDF matrix, then performing dimensionality reduction, and capturing the most important concepts. This method is rooted in advanced linear algebra and is a simpler approach when compared to neural nets and transformers. T5 Abstractive Summarization is a transformer architecture that is pre-trained on a large corpus of text; in our case, T5 was trained on the Billsum and CNN Daily Mail datasets from HuggingFace [Wolf et al. (2020)]. It is pre-trained in a masking process where the goal is to predict and identify masked words in a text sequence given the context of the sentence. The model is then fine-tuned on pairs of text and their corresponding summaries to perform abstractive summarization. The final model we test, GPT-3.5-turbo, is a General Pre-trained transformer model from OpenAI with a total of 154 billion parameters. The model has been fine-tuned using RLHF (Reinforcement Learning with Human Feedback), on a variety of different tasks, and it can effectively perform summarization - in fact, summarization is one of GPT's most rudimentary capabilities.

We developed a script that runs summarization using all three of these models and uses ROUGE scores to evaluate the quality of the summarization. We hypothesize that LSA will perform the worst empirically, as it is just extractive summarization, and T5 will perform moderately, while GPT-3.5-turbo will outperform the other models in terms of the quality of the summary. We also quantitatively evaluate the summary similarities to one another using a ROUGE score between model summaries.

2.0 Related Work

The abstractive summarization task has been a new undertaking coming from agents summarizing long documents. In 2018, abstractive summarization was introduced by Celikyilmaz et. al in Deep Communicating Agents for Abstractive Summarization. Their approach attempted and defined abstractive summarization using recurrent neural networks in a bidirectional long short term memory model with a contextual encoder and decoder for attention. They achieved ROUGE scores of 41.69, 19.47, 39.08 on ROUGE-1, ROUGE-2, and ROUGE-L. They demonstrated that the process can have multiple metrics for evaluation and salient attention mechanisms can perform well without manual evaluation.

Another important approach involves masking methods in a PEGASUS model from Zhang et al. where they trained a transformer architecture with masking tokens or sentences with overlapping similarity and providing a synthetic sentence which generates a composite representative sentence in the output. This approach from 2020 achieved state of the art ROUGE performance on all of their 12 summarization tasks spanning news, science, stories, instructions, emails, patents, and legislative bills. Furthermore, the human evaluation validated that their model summaries achieve human performance on multiple datasets.

In T5, Raffel et al. (2019) generalized the text-to-text framework to a variety of NLP tasks and showed the advantage of scaling up the model size to 11 billion parameters. Including a pre-trained corpus, introducing C4, a massive text corpus derived from Common Crawl, which was used in a variety of different models. T5 was pre-trained with randomly corrupted text spans of varying mask ratios and sizes of spans. T5 achieved state of the art performance on benchmark tasks including abstractive summarization.

Lastly, the work of OpenAI in "Improving Language Understanding by Generative Pre-Training is a seminal paper that provides context for our gold standard approach to abstractive summarization. Learning effectively from raw text is crucial for reducing the reliance on supervised NLP. Additionally, this addresses the limitations of researching in NLP with manually labeling data. This poses a challenge to researchers because there is a lack of consensus on optimization objectives and transfer methods. Researchers at OpenAI propose a semi-supervised approach, a combination of unsupervised pre-training and supervised fine-tuning. Their goal is to learn a universal representation that transfers with little adaptation to a wide range of tasks. The transformer architecture, as previously stated, is used for its strong performance in handling long-term dependencies and robust transfer performance across diverse tasks. To conduct transfer between tasks (e.g., natural language inference, question answering, semantic similarity, and text classification), the input is processed as a task-specific input, allowing the model to maintain the context and relationships between parts of the input and of course the task domain.

3.0 Methodology & Comparative Study

To further investigate the performance between models, we needed to come up with a system to effectively compare the results. As summarization is intrinsically a subjective task, there's no precise way to objectively score a summary. What we can do, however, is compare the results to one another using a ROUGE score. ROUGE, which stands for Recall-Oriented Understudy for Gisting Evaluation, is a widely-used evaluation metric for summarization tasks that quantifies the agreement between generated summaries and reference summaries. It's designed to measure the quality of a generated summary by comparing it to one or more reference summaries, typically created by human annotators. using n-gram overlap. It includes variants like ROUGE-N, ROUGE-L, and ROUGE-W, which consider different aspects of summary quality such as n-gram matches, longest common subsequences, and weighted n-gram matches, offering insights into the performance of summarization models from various perspectives, all of which we employed for our analysis.

There are several reasons to consider ROUGE for our chosen evaluation metric for summarization tasks. ROUGE offers a quantitative way to measure the quality of a generated summary compared to human-created reference summaries. This allows us as researchers to easily compare and rank different summarization algorithms. Second, ROUGE is a computationally efficient method that can be applied and scaled to large evaluation tasks, making it suitable for evaluating summarization models that are trained and tested on massive datasets. Lastly, ROUGE is a widely used benchmark and has been adopted by the research community which makes it easy to compare for external validity.

However, there are also good reasons to consider being cautious about the limitations of the ROUGE scoring method. First, it is based on n-gram overlap and does not consider the semantic meaning of the text. As a result, it may not always align with human judgment, especially when summaries convey the same meaning using different words. Second, the quality of reference summaries can significantly impact the ROUGE scores. If reference summaries are not well-written or do not cover all aspects of the source text, the evaluation results may not be reliable. Furthermore, summary references done by humans are subjective and rely on some degree of expert knowledge which itself can vary and influence the results. Lastly, ROUGE does not explicitly evaluate the structure and coherence of generated summaries, which are important aspects of summary quality from a human perspective. Despite these limitations, ROUGE remains a valuable, popular metric for text summarization tasks due to its quantitative nature, scalability, and widespread adoption in the research community.

Using the ROUGE method, we can evaluate how much overlap there is between summarizer outputs on the same input text. Generally, we expect the summaries to be quite similar and choose generative sentences that should convey the same overarching representation for the input text. As our hypothesis was centered around the performance hierarchy (where we believed the LSA would perform worst, T5 would perform moderately, and GPT would perform best), we developed a summarization and evaluation pipeline to assess a variety of input texts.

The pipeline started with a popup TKinter GUI window, where the user could choose text from numerous mediums: Local .txt file, website URL, copy & paste or typing into a text box, and even speech to text recognition powered by Google Cloud Speech Recognition API. This input structure was intended to make summarizing easy and efficient for any user. The selected text was then sent into each of our three models, where the summaries were computed and saved.

4.0 Data & Setup

In this section, we will discuss the data and setup used for implementing and evaluating the LSA, T5, and GPT-3 models in our study. Each of these models has its own set of tasks and is built on a diverse and extensive set of data to form the foundation of their intelligence. Fine-tuning and supervision enable researchers to adjust the models for specific tasks and track the alignment between the model's performance and the intended research objectives. We will provide an overview of the specific data sources and settings used for training, fine-tuning, and evaluating the models.

4.1 Latent Semantic Analysis

LSA is a widely-used natural language processing technique for extracting and representing the underlying semantic structure in a text corpus. It employs dimensionality reduction, specifically Singular Value Decomposition (SVD), to achieve this goal. While LSA is not a specific model with predefined experiments, it is often applied to various text dataset for different purposes such as information retrieval, document clustering, and in our case: summarization. Commonly used datasets for tuning include the 20 Newsgroups collection of 20,000 newsgroup documents evenly distributed among 20 newsgroups, the Reuters-21578 dataset of news documents, and TREC retrieval tasks using a wide variety of text collections.

For text preprocessing, the LSA method uses tokenization to divide the text into tokens, removing stopwords that do not carry significant semantic information, then incorporating stemming or lemmatization to truncate duplicate words of the same etymology by removing inflections (stemming) or converting them into their lemma. Next, Term Frequency-Inverse Document Frequency (TF-IDF) computes the TF-IDF weight for each term in the document to emphasize important terms related to the document and also reduce the impact of common terms.

The LSA unsupervised learning uses evaluation metrics like Precision, Recall, and F1 score for text classification, Normalized Mutual Information, Adjusted Rand Index, or Purity for clustering tasks, and Mean Average Precision or Discounted Cumulative Gain for information retrieval tasks.

Lastly, the experimental settings of the LSA model involve choosing the number of dimensions to retain after dimensionality reduction (k). A smaller k value leads to more compressed, general representations, while a larger k preserves more of the original structure. The choice for the value of k is determined by experimentation or by choosing application specific requirements. The setup for the deterministic LSA model start with creating a term-document matrix using the TF-IDF values, then apply singular value decomposition to the term-document matrix, resulting in three matrices: term matrix (U), singular value matrix (S), and document matrix (V^T). Next, these matrices are truncated, and the top k dimensions are retained for the further analysis or application-specific tasks such as document similarity, clustering, classification, or summary.

4.2 Text-to-Text Transfer Transformer

The T5 model is focused on transferability, so the data required to achieve the researcher's goals must be sufficiently large and representative of the variety of information. To do this, researchers leveraged the Colossal Clean Crawled Corpus (C4) from Common Crawl [Raffel et al. (2019)]. It provides "web extracted text" in a process that produces around 20TB per month of scraped data. The text is not quite natural language because it includes titles, menu items, and other boiler-plate text. Researchers used heuristics to filter the vast dataset to ensure the inclusion or discrimination of natural language based on punctuation, sentence length, nefarious subjects/topics/words, etc. to get the cleaned, English dataset. The resulting dataset was approximately 750GB.

The Text to Text Transfer Transformer (T5) is a pre-training framework designed for various NLP tasks, which employs a unified text-to-text format for both pre-training and fine-tuning. In the original T5 paper, the authors experimented with a wide range of tasks and datasets to demonstrate the versatility and effectiveness of the model [Kudo & Richardson (2018)]. The datasets used in our experiments include the GLUE benchmark, which consists of nine NLP tasks, such as sentiment analysis, textual entailment, and semantic similarity. Additionally, we employ the SuperGLUE tasks, designed for more challenging undertakings like natural language inference, question-answering, and coreference resolution. The SQuAD question-answering dataset was used for question-answering tasks, where the model is required to answer questions within a provided context. For translation, the WMT translation task set includes English-German and English-French translation. Lastly, we used the CNN/Daily Mail summarization task, in which the model must generate summaries of news articles.

For text preprocessing, the primary preprocessing step in T5 is converting all tasks into a text-to-text format. In this format, the input and output are considered as sequences of text. This approach simplifies the overall architecture, as the model uses the same training mechanism for both pre-training and fine-tuning. Additionally, the input text is tokenized using the SentencePiece library, which employs subword tokenization to handle out-of-vocabulary tokens and reduce the overall vocabulary size [Kudo & Richardson (2018)].

For data splits and evaluation metrics, the datasets used in the experiments are typically split into training, validation, and test sets. The training set is used for model training, while the validation set is used for hyperparameter tuning and early stopping. The test set is used to evaluate the final model performance. The evaluation metrics used in T5 vary depending on the task, but examples include: Accuracy for classification tasks like sentiment analysis, BLEU score for translation tasks, ROUGE score for summarization tasks, and F1 score for question-answering tasks.

Lastly, for the experimental settings, T5 is trained using a denoising autoencoder approach, where the input text is corrupted by randomly masking tokens, and the model must reconstruct the original text. The model is trained using maximum likelihood estimation, and the learning rate is adjusted using a cosine learning rate schedule with a linear warm-up phase.

During the fine-tuning phase, the model is trained on task-specific datasets. The authors experimented with various model sizes, ranging from small models with 60 million parameters to large models with 11 billion parameters [Raffel et al. (2018)]. To ensure a fair comparison, the authors control for computational resources during fine-tuning, ensuring that larger models are not given a disproportionate advantage.

The experiments demonstrated that the T5 model performs competitively across a wide range of NLP tasks, setting new state-of-the-art results on several benchmarks [Raffel et al. (2019)].

4.3 Generative Pre-trained Transformer 3

GPT-3 is a large-scale model developed by OpenAI (Radford et al. 2020), designed to perform a wide range of natural language processing tasks. GPT-3 is pre-trained on a diverse and extensive dataset and fine-tuned for specific tasks using a smaller, task-specific dataset.

The Datasets used for the experiments include the WebText dataset, which contains a vast collection of web pages extracted from outbound links found in Reddit posts with a minimum of three karma points [Radford et al. (2019)]. The dataset encompasses a wide variety of topics and genres, such as news articles, blog posts, and web pages. For fine-tuning and evaluation, GPT-3 uses task-specific datasets similar to the T5, including GLUE, SuperGLUE, SQuAD, WMT, and CNN/Daily Mail.

For text preprocessing, the GPT-3 model first employs the Byte Pair Encoding (BPE) tokenization method for text preprocessing. BPE reduces the overall vocabulary size by dividing words into smaller subword units, which helps manage out-of-vocabulary tokens and allows for better generalization. GPT-3 does not require converting tasks into a text-to-text format like T5, as it is designed to generate contextually relevant text based on the provided input prompt. GPT-3 uses the same methodology for data splits and the Accuracy, BLEU, ROUGE, and F1 scores for evaluation.

Lastly, the experimental settings for the GPT-3 model start with the unsupervised method of maximum likelihood estimation, with the Transformer architecture as its foundation. The model learns to generate contextually relevant text based on the input prompt. GPT-3 is available in several sizes, ranging from smaller models with tens of millions of parameters to the largest model with 175 billion parameters. During fine-tuning, GPT-3 is trained on task-specific datasets with only a few gradient updates, which is known as "few-shot learning." This approach enables the model to adapt to specific tasks with minimal fine-tuning and demonstrates GPT-3's ability to generalize effectively. The experimental results showed that GPT-3 achieved state-of-the-art performance on several NLP tasks and benchmarks [Radford et al. (2020)].

5.0 Results & Evaluation

5.1 Performance Hierarchy Hypothesis

In our experiments, we observed a performance hierarchy among the models, with GPT-3 consistently outperforming the other models in terms of summary quality and coherence. T5 followed as the second-best performer, while LSA lagged behind in terms of summary helpfulness and context preservation. This performance hierarchy supports our hypothesis that more advanced models such as GPT-3 would be better at summarization tasks than older techniques like LSA.

5.2 Impact of Text Complexity on Model Similarity

We found that the complexity of the input text had a significant impact on the similarity of summaries generated by the models. When working with Basic English texts, which are designed to be simpler and more accessible to non-native speakers, the models produced summaries with higher ROUGE similarity scores compared to the summaries generated from standard English Wikipedia articles. This finding suggests that the models have an easier time summarizing and understanding simpler texts, leading to more similar and potentially more accurate summaries.

5.3 Other Relevant Findings

Our experiments also highlighted the limitations of the models in terms of token count and processing capacity. GPT-3, despite its superior performance, was limited by the maximum token count of 4096 tokens. T5, on the other hand, had a more restrictive token limit of 512 tokens in our GUI implementation, which significantly impacted the quality and coherence of its summaries. This limitation could potentially be addressed by processing larger texts in smaller chunks and re-summarizing the resulting summaries, although this approach may introduce additional issues, such as loss of context and nuance.

5.4 Evaluation of Models

GPT-3 performed best in ROUGE score comparison, summary helpfulness, and lucidity but was limited by a 4096-token count. LSA was faster but produced incomplete or out-of-context summaries, rendering its methods insufficient. T5 provided a good summary of topics but had inconsistent coherence, possibly hindered by the 512-token sentence limit.

Our approach for analysis required valid measurement tools. However, evaluating text summaries involves subjectivity. We compared texts with congruent topics but different complexity levels, using standard and Basic English Wikipedia articles on Artificial Intelligence. Basic English, or simple English, is a controlled language, designed to aid non-native English speakers with comprehension by using shorter sentences and easier words and grammar. We expect that Basic English will lead to higher similarity ROUGE scores between summarizers.

These similarity percentages represent the degree of similarity between the summaries generated by different models (LSA, T5, GPT) using unigram and bigram comparisons. Unigram similarity compares individual words, while bigram similarity compares pairs of consecutive words. The overall similarity provides a more comprehensive comparison by considering both unigram and bigram similarities.

Presenting the ROUGE score comparisons for each model when comparing standard, en.wikipedia summaries.

1. LSA vs. T5:
   - Unigram similarity: 18%: There is an 18% overlap in the single words used in the summaries generated by SA and T5.
   - Bigram similarity: 0%: There is no overlap in the word pairs used in the summaries generated by SA and T5.
   - Overall similarity: 11%: The overall similarity, taking into account both unigrams and bigrams, is 11%.
2. LSA vs. GPT:
   - Unigram similarity: 8%: There is an 8% overlap in the single words used in the summaries generated by LSA and GPT.
   - Bigram similarity: 0%: There is no overlap in the word pairs used in the summaries generated by LSA and GPT.
   - Overall similarity: 7%: The overall similarity, taking into account both unigrams and bigrams, is 7%.
3. T5 vs. GPT:
   - Unigram similarity: 19%: There is a 19% overlap in the single words used in the summaries generated by T5 and GPT.
   - Bigram similarity: 2%: There is a 2% overlap in the word pairs used in the summaries generated by T5 and GPT.
   - Overall similarity: 12%: The overall similarity, taking into account both unigrams and bigrams, is 12%.

Presenting the ROUGE scores for comparison between models when comparing Basic English, simple.wikipedia summaries:

4. LSA vs. T5:
    - -Unigram similarity: 33%
    - -Bigram similarity: 32%
    - -Overall similarity: 33%
5. LSA vs. GPT
    - -Unigram similarity: 5%
    - -Bigram similarity: 0%
    - -Overall similarity: 4%
6. T5 vs. GPT
    - -Unigram similarity: 15%
    - -Bigram similarity: 1%
    - -Overall similarity: 8%

These differences in similarity could be attributed to the distinct approaches and architectures of the models. LSA is a statistical technique that relies on the frequency of words and their co-occurrence, while T5 and GPT are neural network models that use a more sophisticated understanding of language. The disparity in their summarization outputs highlights the variations in the ways these models capture and convey the essential information from the input text. GPT would usually produce the largest summary output, and T5 had options for sentence output count, so notably, verbose or larger summaries would cause the unigram and bigram similarity to increase.

These results indicate that there is some overlap in the summaries generated by the models, with T5 and GPT showing the highest similarity. However, the similarity percentages are relatively low, suggesting that each model is generating summaries using different word choices and phrasings due to differences in the underlying algorithms, training data, or other factors. We observed higher overlap in unigram and bigram similarity for simpler Basic English texts. From this, we argue that documents with higher complexity levels will have distinct and unique summaries, while simpler articles will exhibit less variability between summarizer outputs.

6.0 Conclusion & Future Work

Our investigation focused on evaluating and comparing summarizer models, a challenging task due to subjectivity and contextual factors. Natural language conveys nuanced information, and summarization requires omitting some details. We found that complex documents typically have higher variance between summarization techniques, and a consistent bigram and unigram overlap is desirable for validity. As AI becomes more ubiquitous, responsible usage becomes increasingly important. Summarizers have various applications, such as aiding reading comprehension or providing executive overviews. Summarizing simple, basic text results in stronger replicability and validity.

Implications of our results suggest that understanding the limitations and strengths of different summarization models is crucial for their responsible and practical application. By recognizing the variability in summarization outputs, users can make informed decisions when utilizing AI-generated summaries.

Future research directions include using simple, Basic English as a control measure, comparing segmentation methods for longer documents, and investigating the impact of document length on summarization. Researchers should also explore larger models, such as GPT-4, and assess the relationship between model size and summarization capability. Additionally, integrating expert knowledge into abstractive summarization through knowledge-based systems could improve contextual understanding and overall performance.

Potential applications and extensions of our work involve optimizing summarization models for various industries, enhancing educational tools, and developing personalized content summarization systems. By understanding the nuances and limitations of different summarization models, we can better tailor AI-generated summaries for specific user needs and contexts.

Citations

1. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., ... & Liu, P. J. (2019). Exploring the limits of transfer learning with a unified text-to-text transformer. arXiv preprint arXiv:1910.10683.
2. Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training. URL https://s3-us-west-2. amazonaws. com/openaiassets/research-covers/languageunsupervised/language_understanding_paper. pdf.
3. Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. Journal of the American Society for Information Science, 41(6), 391-407.
4. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in Neural Information Processing Systems, 30, 5998-6008.
5. Sennrich, R., Haddow, B., & Birch, A. (2016). Neural machine translation of rare words with subword units. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 1715-1725.

6. Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., & Bowman, S. R. (2018). GLUE: A multi-task benchmark and analysis platform for natural language understanding. arXiv preprint arXiv:1804.07461.
7. Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016). SQuAD: 100,000+ questions for machine comprehension of text. Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, 2383-2392.
8. Hermann, K. M., Kocisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., & Blunsom, P. (2015). Teaching machines to read and comprehend. Advances in Neural Information Processing Systems, 28, 1693-1701.
9. Ogden, C. K. (1930). Basic English: A general introduction with rules and grammar. Kegan Paul, Trench, Trubner & Company.
10. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., & Brew, J. (2020). Transformers: State-of-the-Art Natural Language Processing. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (pp. 38-45). Association for Computational Linguistics.