# Tell Me What I Don't Know: Generating Selective Abstract Summaries

**Dylan Mair**
UC Berkeley / Berkeley, CA
IHS Markit / Singapore
dmair@berkeley.edu

**Ernesto Martinez**
UC Berkeley / Berkeley, CA
LA County DPSS / Los Angeles, CA
futureperfect16@berkeley.edu

## Abstract

Incident reports may contain categorical data and free text descriptions. This paper simulates abstract summarization of such reports to uniquely capture all of this content. CNN stories from the *CNN / Daily Mail* summarization task dataset is cleaned up to use as a proxy for our incident reports.

A baseline T5 transformer model is generated with a small pre-trained model, the CNN training stories and associated reference summaries. Summaries of CNN test stories are found to have ROUGE scores comparable to prior work. Single sentence reference summaries are also modeled to measure the reduction in ROUGE scores that result from shorter summaries.

A sentence is separated from each reference summary to represent 'known' categorical data for exclusion from predicted summaries. A naïve model built without known sentences generates summaries with a lower ROUGE score, with no detectable improvement in ROUGE scores. Filtering the results of our baseline model to remove generated summary sentences that resemble our known data were also unsuccessful. However a novel solution that appends each 'known' sentence to its input story ahead of modeling was successful, significantly improving the ROUGE score for the remaining sentences.

## 1 Introduction

The confidential incident reports produced by a customer support organization combine natural language description and specific facts of the engagement - such as 'who' carried out 'what' with 'whom' (see Table 1). The description can be very noisy, such as an email trail, and may duplicate the facts. The facts alone lack the insights that the Sales team seek from carefully crafted summaries

| | |
|---|---|
| *CSA:* **Adam** | *Customer:* **Shell** |
| *Activity:* **Training** | *Products*: **GEPS** |
| *Attendees:* **Coen, Anouk, Mae, Lee, Lars, Elke** | |
| *Description*: Second GEPS training for Shell, 6 attendees this time and interacting during the call with questions, one user commented that the GEPS product would be excellent for her day to day workflow when looking at competitor monitoring. I showcased the Upstream Intelligence Beta and the global impact special interest tags and alluded to the API but there was no direct interest from users. | |
| *Sales Summary:* **Adam trained 6 users at Shell on GEPS.** GEPS is seen as excellent for competitor monitoring. These users were not interested in APIs. | |

Table 1: Example incident report and its summary prepared for Sales. Known data is shown **in bold**.

| |
|---|
| LIMA , Peru -LRB- CNN -RRB- – Seven children and two teachers were killed Monday when a bridge collapsed in southern Peru , according to a health department official .<br>Fifty-five others were injured in the incident , which occurred near a school in Peru 's Ayacucho province , said Director Maria Torrealba .<br>Further details were not immediately available , nor were the conditions of those injured in the incident .<br>Journalist Maribel Salas contributed to this report |
| *@highlight* **Fifty-five others were injured in the collapse** |
| *@highlight* The incident has occurred near a school in Peru 's Ayacucho province |
| *@highlight* The conditions of those injured in the incident is not immediately available |

Table 2: Example CNN story with a highlight **in bold.**

of incident reports. Summarizing over 10,000 reports every year is impractical. Delegating the task produces inconsistent results. We seek to automate this summarization, combining facts and description without duplication.

The incident reports are proprietary so we sought a similar dataset in the public domain. No incident reports were found that provided categorical fields, free text descriptions and summaries. Crime reports come close but descriptions were heavily redacted for privacy and summaries were rigidly structured, more categorical than insightful. Since the 'known' facts needed to ultimately be represented as a 'known' summary sentence we selected the tokenized CNN stories dataset from the *CNN / Daily Mail* summarization task, using its multiple summaries to represent both known information and reference summaries. An example story is shown in Table 2. The 'known' information in Table 1 is likely to form a much more formulaic sentence than the much more random 'known' information in Table 2; unfortunately we cannot leverage that pattern while using the CNN dataset.

We seek a process that summarizes the story without stating what we already know. Applying our learnings from the CNN dataset to our incident reports we hope to automate summarization of incident reports to showcase insight on bugs, enhancements, data quality issues and competitor intelligence that keep sales engaged with these opportunities.

## 2 Background

Our task includes abstract summarization; summary metrics; and steering our answer away from a specific outcome.

Raffel et al. (2020) uses the T5 transformer to generate abstract summaries. This Transformer architecture uses self-attention mechanisms to solve multiple text processing problems. That paper evaluates generated summaries against reference summaries using ROUGE metrics. We also begin with the same metrics. Most importantly this paper describes a baseline for our task built using comparable compute power.

Ganesan (2018) notes that ROUGE does not capture synonymous concepts. ROUGE compares n-gram overlap of words. This may be an odd choice of abstract summarization metric, if we hope to leverage word embeddings more deeply. Such a metric is left for future work; we did attempt to use tf-idf vectorization of summary sentences as an alternative to ROUGE (with little success so far).

Rachman et al. (2019) pursued a very similar research topic with a view to creating summaries of updated articles that exclude information that had previously been summarized. The authors focus on making sentence comparisons, beginning with an initial summary created by manually selecting sentences (a kind of extractive summarization). Sentence similarity is determined using Maximum Marginal Relevance (MMR) and TextRank. Once a final set of summary sentences is chosen the resulting multisentence is evaluated using ROUGE. The authors recognize that ROUGE-2 and ROUGE-SU4 considers the order of words, while TextRank similarity considers a one-gram word. The project had a ROUGE-2 score of 15.21. Our own work settled on a combination of ROUGE and tf-idf for comparing sentences. We had considered MMR (Carbinell and Goldstein, 2017) for summarization of stories before using T5 (and ROUGE) to take advantage of Raffel et al. (2020) existing benchmarks.

These benchmarks included reference to the current 'state of the art' ROUGE-1-F score of 43.33 for the summarization task, published by Dong et al. (2019). However the CNN/DM dataset was combined with 4 million articles in the Gigaword dataset. This might not seem a fair comparison but it does demonstrate the value of having a much larger corpus.

Similarly, Rachman et al. (2019) achieved ROUGE-1-F scores of 42.05 by further pre-training the t5-base pre-trained dataset on a variety of tasks, not just summarization, before continuing with the modeling exactly as described later in this paper aside from adding a beam search. The multi-task pre-training amounted to an artificial data set size of 2,620,000, vastly greater than we were able to digest for this project. Fortunately our baseline from this paper scores only half the ROUGE score.

## 3 Methods

The *CNN / DM* dataset is widely used to benchmark abstract summarization performance. The dataset was originally created by Hermann et al. (2015) by crawling the Cable News Network and Daily Mail websites. The process was recreated by Nallapati et al. (2016) and then See et al. (2017) provided a list of links to a web archive. The tokenized

| Model | Multisentence Baseline | Single Sent. Summaries | Raffel et al (2020) | Naïve Exclusion | Repatriated Highlight |
|---|---|---|---|---|---|
| Pre-trained model | t5-small | t5-small | t5-base | t5-small | t5-small |
| Pre-trained parameters | $\sim$ 60M | $\sim$ 60M | $\sim$ 220M | $\sim$ 60M | $\sim$ 60M |
| Sent. per ref. summary | 2-5 | 1 | 1-5 | 1-4 | 1-4 |
| Training pairs | 89,904 | 322,487 | 287,226 | 322,487 | 322,487 |
| Validation pairs | 1,182 | 3,054 | 13,368 | 3,054 | 3,054 |
| Testing pairs | 1,081 | 1,081 | 11,490 | 1,639 | 2,720 |
| Batch size | 8 | 8 | 128 | 8 | 8 |
| Beam size | 4 | 4 | 1 | 4 | 4 |
| Training epochs | 2 | 1 | unknown | 1 | 2 |
| Total fine-tuning steps | 44,952 | 40,311 | 262,144 | 29,073 | 153,205 |
| Training loss | 0.2051 | 0.0835 | 2 | 0.0888 | 0.1325 |
| Av. ROUGE-1-F | 26.00 | 13.96[1] | 19.24[2] | 16.68 | 20.07 |

[1] average maximum score per article of 22.38

[2] using the validation dataset

Table 3: Results of modeling. The Raffel et al. (2020) CNN/DM dataset baseline is added for comparison.

data was later made available by Jaffer Wilson[3] however training and testing labels only existed for the full binary-format CNN plus DM dataset. The text-format CNN dataset was labelled by accessing every test and validation url and page and matching it to a tokenized story. While onerous, this close attention identified a number of issues for cleaup.

The 92,579 CNN stories cleanup included:

- 59 stories had only one highlight. We need to simulate both known and unknown results.
- 221 stories were landing pages for audio-visual reports with duplicate (boilerplate) highlights. These were removed. Including these could produce memorization instead of generalization (Elangovan et al., 2021).
- 7 duplicated stories were removed. The duplicates did not refer to different events. Processing duplicates may also lead to memorization. Testing training stories against training stories would likely identify further duplicates.
- 119 empty and very short (less than 3 words) stories were removed.

We finished with 89,904 training stories, 1,182 validation stories and 1,081 testing stories.

ROUGE scores comparing each pair of sentences from an article's reference were found to generate high scores. In just two cases this revealed duplicate summary sentences. It was however quite informative to see very high ROUGE scores attributed to summary sentences telling very different facts. For instance, the following reference summary sentences produced a ROUGE-F-1 score of 0.846:

”Phillip Garrido is serving a sentence of 431 years to life in prison”

”Nancy Garrido is serving a sentence of 36 years to life in prison”

For each of the models presented here, we began with a pretrained T5 model implemented by Huggingface[4]. It consists of a six module self-attention network. All models were trained for one or two epochs (time constraints prohibited a greater number of epochs). All including the literature used SentencePiece for the WordPiece vocabulary (32,000 word pieces). A summary of other hyper-parameters used for generating the models is shown in Table 3.

All models used a learning rate of 0.001, matching the baseline by Raffel et al. (2020). We conducted a short experiment to explore the effect of different learning rates. We trained a multisentence model using only 100 stories six times, changing only the learning rate for each iteration. Out of the six learning rates considered (0.001, 0.0005, 0.001, 0.005, 0.01, 0.05), 0.001 resulted in the lowest test loss.

### Multisentence Baseline

The pre-trained model was then fine tuned using 89,904 training examples and 1,182 validation ex-

[3] github.com/JafferWilson/Process-Data-of-CNN-DailyMail

[4] https://huggingface.co/transformers/model doc/t5.html

amples. For each example, the source text was the CNN story, while the target was a short paragraph comprised of the story highlights. In addition to computing loss for a 1,081 story test set, ROUGE scores were calculated for each example to compare the multisentence summaries with the story highlights. A beam size of 4 was used during the summary generation process during testing. The model took 5 hours to train on Google Colab Pro.

While this T5 model does not attempt to identify or ignore any a priori knowledge, we developed a post-processing workflow to compare the T5 summary to a highlight deemed ”previously known” information, which is discussed below.

**Single Sentence Summaries**

As with our baseline model, the input for the single sentence summary model was the full story text. Rather than a multisentence target (as the rest of the models were trained), the single sentence summary model was trained on individual highlights. During training, every story was used as the source text for each associated highlight, increasing the size of the training set to 322,487. The model was otherwise produced following the same methodology as the multisentence baseline model. This model took 13 hours to train.

**Naïve Exclusion**

As with the baseline model, the Naïve Exclusion model takes a story as the model input, and the target consisted of a paragraph composed of the story highlights. For this model, however, one highlight (representing a priori knowledge) was excluded from training altogether. At evaluation, a highlight was simililarly not used to compare with the output summary sentences. This model took about 2 hours to train.

**Repatriated Highlight**

The final model represents a novel approach to downplaying the ’known’ highlight. attempted to identify and ignore previously known information by fine-tuning the T5 model itself. The input for this model was the story and a single highlight (this is deemed known information), and the target was a paragraph comprised of the remaining highlights. The input was fed to the model as a single string. In keeping with the format of the original data, the input string consisted of (1) the story, (2) a separator ”@highlight” (preceded and followed by two newline characters), and (3) the known highlight. This model, though trained on only two epochs, took over 15 hours to train.

The goal of this approach was to see if the model could learn that the story sentence following ”@highlight” contained information which should not need to included in the output summary.

**Raffel et al. (2020) Baseline Comparison**

This key reference paper thankfully uses a realistic-sized model to demonstrate the T5 Transformer in action. As shown in Table 3, it is still considerably bigger than the models we used, in almost every way. The model begins with a pre-trained model with almost four times as many parameters. Cleaning of the input data may have varied from our own, with single-highlight articles included in this model, but more importantly this model included a corpus of Daily Mail articles.

Hermann et al. (2015) noted the DailyMail dataset contains twice as many stories as the CNN dataset, and almost double the vocabulary of the CNN dataset. The author did not use these datasets for summarization. It was noted that neither dataset was consistently the better or the worse result considering the comprehension tasks they assigned. No beam size was used for the baseline model, thus it is ’greedy’. A beam size of 4 was added to a later, more powerful model.

**TF-IDF Post-processing**

A second novel approach was attempted, to take all summary sentences produced by the Multisentence Baseline model and f

Using TF-IDF, the highlight and each sentence in the T5 output summary were vectorized (these were sentence-level embeddings). The highlight was compared to each sentence from the output using cosine similarity, and the sentence from the output summary with the most similiarity to the known highlight was removed from the output. ROUGE scores comparing the ”trimmed” output and the concatenated unknown highlights (i.e. all highlights but the one deemed known).

## 4 Results and Discussion

Despite the simplicity of our models driven by resource constraints, our Multisentence baseline model demonstrated significant improvement over the baseline model by Raffel et al. (2020). This was in spite of inferior pre-training model size, corpus size and fine-tuning. Further work is required to isolate exactly why, but it could have been our data cleaning (less than 0.5% of articles cleaned up) or perhaps more likely relates to the lower suitability of the DailyMail dataset for the summarization task.

It may even have resulted from using a beam size of 4.

The significantly lower ROUGE score for the Single Sentence model can be easily explained as each ROUGE test is comparing only one reference sentence with a single predicted sentence. The odds of predicting overlap are smaller. This model helps us understand the impact of shrinking the summary.

We expect the results for the Naïve Exclusion model, with just one fewer sentences in each reference summary, to lie between the first two models. And it does! It remains questionable whether we might have improved our predictions by simply "pretending the known data never happened" as shorter reference summaries still cause an overall downgrade in ROUGE scores. If somehow this piqued someone's interest, models could be run with progressively fewer summaries and the results regressed. But the effects of 'shorter' and less 'misleading' training data may be inseparable.

Finally, we have our Repatriated Model. The improvement over the Naïve Exclusion model appears to support our hypothesis that the modeling could be made to forget about the 'known' data, so long as the process is aware of it. Further work can hopefully confirm this improvement is significant. The beauty of this solution is immense, as it only considers the sentence for exclusion only for that story. Additional work should definitely seek to understand why exactly this happens.

Lastly, the TF-IDF post-processing workflow developed to exclude known information yielded poor results from a ROUGE score perspective. This is much worse than we would predict to be the result of short summaries. Further work could revisit this mechanism to try to understand why it failed.

## 5 Conclusions

We have successfully improved our ROUGE scores by

Raffel et al. (2020) uses a larger pre-trained T5 model and a corpus three times as big, as well as running more fine-tuning steps, all of which ought to have produced a superior ROUGE-1-F score. Unfortunately the model did not set a beam size of 4, while ours did. Could that have made such a big difference? The value of a larger corpus to improve ROUGE scores especially has been very clear during our Background reading when applying the T5 Transformer to this dataset. Further work on this project would involve upgrading the pre-trained model, the corpus and the number of epochs, and cleaning up the DailyMail dataset.

While ROUGE scores remain a standard for evaluating summarization algorithms, they are limited in their ability to properly evaluate information content when summaries are abstractive rather than extractive. Further, since the highlights are made to contain only a portion of the information in the news story, poor ROUGE scores can result when an output summary contains different information found in the reference highlight (even if the information is implied by the source text and clear to a human reader). Alternative metrics should be explored. In addition, extending the architecture so that the T5 output is fed to a paraphrase or entailment classifier could be promising. T5 would produce a summary, and a second classifier model could trim sentences that paraphrase the known information from the output of the T5 model.

## Acknowledgments

We would like to acknowledge the support and inspiration given by our exceptional lecturers, Sid Reddy and Mark Butler.

## References

Jaime Carbinell and Jade Goldstein. 2017. The use of mmr, diversity-based reranking for reordering documents and producing summaries. *SIGIR Forum*, 51(2):209–210.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. *Computing Research Repository*, arXiv:1905.03197.

Aparna Elangovan, Jiayuan He, and Karin Verspoor. 2021. Memorization vs. generalization: Quantifying data leakage in nlp performance evaluation. *Computing Research Repository*, arXiv:2102.01818.

Kavita Ganesan. 2018. ROUGE 2.0: Updated and improved measures for evaluation of summarization tasks. *Computing Research Repository*, arXiv:1803.01937.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. *Computing Research Repository*, arXiv:1506.03340.

Ramesh Nallapati, Bowen Zhou, Cicero Nogueira dos santos, Caglar Gulcehre, and Bing Xiang. 2016.

Abstractive text summarization using sequence-to-sequence rnns and beyond. *Computing Research Repository*, arXiv:1602.06023.

Ghoziyah Haitan Rachman, Masaya Leylia Khodra, and Dwi Hendratmo Widyantoro. 2019. Towards guided summarization of scientific articles: Selection of important update sentences. *2019 International Conference on Electrical Engineering and Computer Science (ICECOS)*, pages 259–264.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Computing Research Repository*, arXiv:1910.10683.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. *Computing Research Repository*, arXiv:1704.04368.