ADS

Research Paper : **GenCompareSum** (a hybrid unsupervised summarization method using salience)

ABSTRACT : Pre-trained Language Models (PLMs) have been used to improve the performance of TS. However, **PLMs are limited by their need of labeled training data** and by their attention mechanism, which often makes them unsuitable for use on long documents. To this end, **we propose a hybrid, unsupervised, abstractive-extractive approach**, in which we walk through a document, generating salient textual fragments representing its key points. We then select the most important sentences of the document by choosing the most similar sentences to the generated texts, calculated using BERTScore. We evaluate the efficacy of generating and using salient textual fragments to guide extractive summarization on documents from the biomedical and general scientific domains. **We compare the performance between long and short documents using different generative text models**, which are fine tuned to generate relevant queries or document titles. We show that our hybrid approach outperforms existing unsupervised methods, as well as state-of-the-art supervised methods, despite not needing a vast amount of labeled training data.

INTRODUCTION : Whilst **transformers** have made great advancements in their ability at capturing semantic knowledge, they have also introduced new **limitations. Firstly, they are restricted by the number of tokens** that they can process at any one time. **Another issue is the computational cost of fine tuning** the attention mechanisms embedded in transformers. These constraints are challenging for recent text summarization methods, often resulting in analysis being done on a truncated version of a document. Since summarization should be able to succinctly capture the meaning of very long documents in a few sentences, **the requirement to truncate a document before summarization is a major disadvantage.**

These are mostly supervised methods, requiring large amounts of labeled training data, which are often unavailable or time-consuming and costly to produce. We address the challenges of supervised methods by adopting a hybrid unsupervised approach, where the PLMs are required only to act on short sections of the document at any time, meaning that our method can be extended to any document length. Furthermore, by nature of it being an unsupervised approach, it does not require manually labeled training data for the extractive summarization task. To-date, unsupervised methods for text summarization have generally used graph-based methods, the more recent of these using transformer-based embeddings to calculate weights between the nodes in the graph.

We differ from these previous approaches as we do not use a graph-based model and instead evaluate the effectiveness of a novel approach – generating and using salient textual fragments to guide the extractive summarization.

Text summarization methods are divided into extractive and abstractive groupings. Extractive methods select the most relevant sentences from a document and abstractive methods consider the most relevant

pieces of information to produce new textual fragments which convey the core message. Although abstractive summarization has the potential to be more succinct and readable, in its current state it cannot be trusted to be factually consistent, making it unsuitable in many practical applications, such as summarization of biomedical articles for use by clinicians.

Specifically, we use transformer-based models for the generation of salient points, but ultimately, we generate an extractive summarization to ensure factual consistency.

WORKING : Our method, GenCompareSum, is a two-step hybrid summarization approach. GenCompareSum first splits a document into sections of several sentences and walks through them, generating salient textual fragments which represent each section. We experiment with different generative models, which are fine tuned to predict either queries or document titles, that best represent a section of the document. Our method then uses these generated textual fragments to guide an unsupervised extractive summarization by calculating the BERTScore similarity between each of the generated texts and each of the sentences in the source document. We then select the sentences with the highest scores to form the extractive summarization.
We evaluate our approach on short and long versions of data sets from the biomedical and scientific domains. Furthermore, we compare the use of different PLMs for generating salient textual fragments.

Our main contributions are as follows:

1. A novel two-step unsupervised hybrid abstractive-extractive summarization method, which generates salient textual fragments - queries and document titles - which represent sections of a document, and then uses them to guide the extractive summarization step.
2. The fusion of state-of-the-art PLMs with unsupervised approaches, to achieve a summary which harnesses the semantic knowledge of transformer-based models, whilst being extendable to any length document, without requiring a large corpus of training data.
3. Evaluation results demonstrate our hybrid method outperforms both existing unsupervised methods and state-of-the-art supervised methods, both on long and short documents.

METHOD : We propose GenCompareSum, a hybrid abstractive-extractive model, which makes use of transformer-based architectures but is extendable to any document length, can represent multiple facts, and does not require vast amounts of training data. The method is comprised of two steps: first, using a generative model to produce salient textual fragments, i.e., queries or document titles, which represent key points from across a document, then a comparison between these salient fragments and each sentence, to select the most important sentences from across the document.

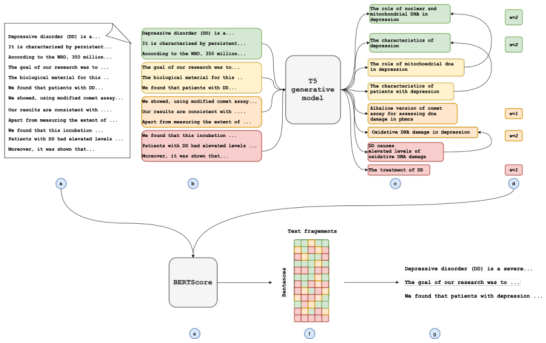

Figure 1: GenCompareSum pipeline. (a) We split the document into sentences. (b) We combine these sentences into sections of several sentences. (c) We feed each section into the generative text model and generate several text fragments per section. (d) We aggregate the questions, removing redundant questions by using n-gram blocking. Where aggregation occurs, we apply a count to represent the number of textual fragments which were combined and use this as a weighting going forwards. The highest weighted textual fragments are then selected to guide the summary. (e) The similarity between each sentence from the source document and each selected textual fragment is calculated using BERTScore. (f) We create a similarity matrix from the scores calculated in the previous step. These are then summed over the textual fragments, weighted by the values calculated in step (d), to give a score per sentence. (g) The highest scoring sentences are selected to form the summary.

Text Splitting :

T5 is a sequence-to-sequence model, pre-trained on a cleaned and pre-processed version of the Common Crawl2 dataset – a data set consisting of textual content scraped from the internet. T5-based models have been shown to be high performing sequence-to-sequence models across a range of generative tasks, from question generation, to graph-to-text generation, to generative common-sense reasoning, to abstractive text summarization. The T5 model uses an encoder-decoder architecture and is pre trained via an unsupervised task in which 15% of tokens are masked; the masked words can be individual words or a span of words; the target of the training objective is to predict these masked words, given the un-masked tokens and their respective positions.

For the generation of the textual fragments, we first experiment with a T5-based model fine tuned with a query generation task in the general domain. This model, provided textual input, aims to generate queries which ask the most relevant questions of it. We use docTTTTTquery, a question generation model trained on the MSMARCO data set, which is a question-answer data set generated from Bing's3 search query logs. It showed that this pre-trained model to be effective at generating questions for long texts.
Second, we finetune our own model on long-answer - query pairs from the biomedical domain. We refer to this model as 't5-med-query'.

Last, we experiment using an open-source T5- based model, fine tuned on abstract-title pairs from the scientific domain4 . This approach has shown to be effective at proxying highly abstractive summaries. We apply this model to our problem space, generating potential 'titles' for each document section. We refer to this model as 't5-s2orc-title'.

N-Gram Blocking - N-gram blocking is a technique which is applied to reduce redundancy and improve coverage in summarization models. We apply n-gram blocking to the generated textual fragments, Where we have removed generated texts by applying this technique, we keep a count of how many times a similar textual fragment was seen before n-gram blocking.

Text Vector Comparison - BERTScore uses BERTbased token embeddings, calculates the cosine similarity between them and uses greedy matching to match each token in the first text to its most similar

token in the second; these scores are averaged across the sentences to give precision, recall and F1 scores which quantify the similarity between two texts.

3. EXPERIMENTS

Datasets - The data sets included in our experiments are CORD-19, PubMed and ArXiv, and S2ORC. The CORD-19 data set used is the version released on 2020-06-28, containing 57,037 articles relating to COVID19. The S2ORC data set is a large corpus of scientific literature across several domains; we select a random subset of 63,709 articles tagged as being from the biological and biomedical domains. The PubMed and ArXiv data sets are from the biomedical and scientific domains respectively.
As training is not required for unsupervised models, for these methods only the test data sets are used. To train the supervised method, BERTExtSum, which we implement for comparison, we use the training data set to train the model and the validation data set to select the best performing epoch for evaluation on the test set.
Different methods for calculating text similarity were also compared, namely, BERTScore, SimCSE and Sentence Transformers, with BERTScore shown to be the highest performing against a ROUGE metric for the extractive summarization task.

Lastly, we implement GenCompareSum and compare the performance between using different generative text models: docTTTTTquery, t5-med-query, and t5-s2orctitle

4. EXPERIMENTAL RESULTS

Model	PubMed			S2ORC			CORD-19			ArXiv		
	R1	R2	RL	R1	R2	RL	R1	R2	RL	R1	R2	RL
Short Document												
ORACLE	47.27	22.85	43.20	49.29	25.42	45.52	43.47	17.75	39.28	47.29	18.49	41.90
RANDOM	34.98	10.82	31.37	34.69	11.06	31.30	31.64	7.91	28.10	34.53	8.88	30.26
LEAD	35.39	12.07	32.28	**40.50**	**16.72**	**37.68**	34.80	10.17	31.67	34.35	8.75	30.61
LexRank	38.48	13.05	34.92	39.44	14.57	36.13	35.65	10.17	32.11	38.98	11.44	34.64
TextRank	38.15	12.99	34.77	40.17	14.84	36.63	36.25	10.61	32.53	37.97	11.58	33.53
SumBasic	36.11	11.06	32.67	35.99	11.99	32.87	33.63	8.82	30.22	37.14	9.83	33.06
BERTExtSum	38.78	14.47	35.43	39.41	16.14	36.38	34.68	10.34	31.42	39.36	11.74	35.09
GenCompareSum (docTTTTTquery)	37.82	13.12	32.41	38.31	14.27	35.17	33.77	9.73	30.66	38.59	11.49	34.50
GenCompareSum (t5-med-query)	38.54	13.67	35.06	38.96	14.78	35.80	36.77	11.24	33.29	38.92	11.59	34.76
GenCompareSum (t5-s2orc-title)	**39.19**	14.35	**35.65**	40.16	15.84	36.91	**36.84**	**11.35**	**33.35**	**39.66**	**12.30**	**35.38**
Long Document												
ORACLE	61.76	36.78	57.61	64.11	39.21	60.16	59.10	32.09	54.63	60.16	32.17	54.97
RANDOM	37.26	11.19	33.66	37.12	10.23	33.73	33.37	7.70	29.98	34.20	8.70	30.64
LEAD	37.23	11.11	33.67	40.50	16.72	37.68	34.61	10.17	31.68	34.70	10.27	31.37
LexRank	41.02	15.83	37.18	42.60	15.84	38.97	39.50	12.65	35.68	33.94	12.09	30.62
TextRank	34.53	12.98	30.99	36.58	13.23	33.10	32.99	10.39	24.47	26.57	9.20	23.74
SumBasic	40.61	12.42	36.54	36.63	10.43	33.68	33.88	8.24	30.86	33.18	7.75	30.29
BERTExtSum*	41.87	16.01	**38.51**	**43.56**	**17.85**	**40.40**	38.95	12.17	35.48	**40.65**	14.01	**36.89**
GenCompareSum (docTTTTTquery)	40.54	14.77	36.83	40.78	14.24	37.43	36.84	11.19	33.51	38.19	12.76	34.55
GenCompareSum (t5-med-query)	41.60	15.67	37.79	41.84	15.10	38.35	39.33	12.31	35.74	37.17	11.97	33.95
GenCompareSum (t5-s2orc-title)	**42.10**	**16.51**	38.25	43.39	16.84	39.82	**41.02**	**13.79**	**37.25**	39.96	**15.15**	36.19

For the short documents, our method GenCompareSum (t5-s2orc-title) performs best across three out of four of the data sets, and second-best for the fourth data set. There is no clear ‘second-best’ model out of the methods compared for the short data sets.

For the long document data sets, GenCompareSum (t5-s2orc-title) outperforms all other unsupervised models. A strong unsupervised baseline, LexRank has been shown in prior literature to give competitive performance when compared to supervised approaches. In-line with these works, we show LexRank to be the best-performing unsupervised method after our own.

Future Work

Conclusion