SumPubMed: Summarization Dataset of PubMed Scientific Articles

Prerna Bharti
Microsoft Corporation
prernarbharti@gmail.com

Vivek Gupta, Pegah Nokhiz
University of Utah
{vgupta,pnokhiz}@cs.utah.edu

Harish Karnick
IIT Kanpur
hkarnick@cs.iitk.ac.in

ABSTRACT

Text summarization is one of the more challenging and open problems of Natural Language Processing. Most earlier work in this field has been carried out on news article datasets. However, news datasets are not suitable for summarization because the summary is usually placed in the first few lines of the text. Thus, we constructed a new dataset, SumPubMed , using scientific articles from the PubMed archive. The summaries in SumPubMed dataset are from the omnipresent information in the document (not merely the first few lines). The summary also contains scientific jargon making the summarization task more challenging. To verify the quality of summaries in SumPubMed, we conducted human annotation on several aspects, such as coverage of important content without repetition, readability, coherence, and informativeness. We observed that the existing *seq2seq*-based summarization methods struggle to perform well on SumPubMed, opening opportunities for further improvement in scientific summarization models. Furthermore, we observed that current summarization methods on news-based datasets yield acceptable results with ROUGE only because of the simplicity of summary placement and ROUGE's lexical matching-based evaluation. In contrast to the earlier news datasets, we found that ROUGE's scores do not correlate well with our human judgments on SumPubMed. Therefore, indicating the need for new evaluation metrics for scientific summarization.

CCS CONCEPTS

• **Computing methodologies** → **Lexical semantics**; **Language resources**.

KEYWORDS

summarization, resources, dataset, abstractive, extractive, evaluation metric

ACM Reference Format:
Prerna Bharti, Vivek Gupta, Pegah Nokhiz, and Harish Karnick. 2018. SumPubMed: Summarization Dataset of PubMed Scientific Articles. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/1122445.1122456

Woodstock '18, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Text Summarization is a standout amongst the more difficult and open problems of NLP. Summarization requires a compiled portrayal of the document that encapsulates the semantics of the original. Automatic summarization ought to have the capacity to produce summaries that can impersonate human-like summaries. Typical forms of summaries and their uses are: headlines, surveys for items and films, plots, digests, abstracts for researchers, and so on.

Text Summarization approaches can be broadly divided into two kinds: Extractive and Abstractive. Extractive methods discover the most significant sentences or words from the document and 'put them together' to create a summary. Abstractive approaches, conceptually, extract the 'meaning' of the text, compress it, and use language generation tools to produce summarized text. Abstractive summarization requires paraphrasing of sentences. We cannot reach standards of human-like summaries just by extractive methods. Therefore, a hybrid approach is needed in which one do some extraction first and then perform abstractive summarization on the extracted text. Thus, we can treat the extractive summarization as a highlighter and abstractive summarization as a final pen.

A majority of existing methods for summarization are extractive. This may be because the vast majority of datasets on which summarization experiments are currently done consist of news stories. News stories seem particularly suited for extractive summarization. Simply extracting the first few lines of a news story produces a good summary very often. News stories are also short in length compared with other kinds of texts that may need summarization. When documents are long and complex like in essays, research papers, long articles, books, etc., extractive summarization is unlikely to work sufficiently. Such documents are more likely to require abstractive methods that are closer to the way we believe humans generate summaries.

Most of the existing summarization dataset such as CNN, DUC, and Daily Mail are news datasets. These datasets have manually written highlights (i.e., headlines). In these datasets, a typical document is short – approximately 10-15 lines on average. Each document is a news article/news story, and a portion of the text/document contains highlights, which is considered as the summary of the document making them unsuitable for abstractive summarization. Also, the first few lines of the news article/story often produce decent summaries. This first-line-summary artifact makes the existing datasets considerably simple to solve, undermining the hard task of abstractive text summarization.

Thus, we believe it is time to create a dataset that avoids, or at least mitigates the two deficiencies mentioned above. Therefore, we tried to find publicly available documents that are of reasonable length and already have human-generated summaries that are

not localized to one part of the document. We also need to ensure that the overall size of the dataset in terms of the total number of documents that are available is sufficient, i.e., it is necessary to have enough articles. One obvious source that satisfies all the required constraints is a database of scientific articles in the biomedical domain – sourced from MEDLINE. These articles are publicly available, and for a significant fraction, the full text and abstract are freely accessible. They are often of reasonable length, the summary (or abstract) is neither localized nor does it consist of simple extracts from the main document.

Hence, we construct a new summarization dataset from preprocessed PubMed articles, named SumPubMed, where documents are longer, and summaries cannot be extracted by selecting some sentences from a particular location in the document.

Our contributions in this paper are:

- We created a new non-news dataset SumPubMed which has longer text documents and where the summaries are generated from the whole document. In contrast, earlier datasets with news stories (shorter text) appear to mostly have useful information localized in the first few lines.
- We conducted a human evaluation for 50 summaries of 250 words. Each summary is evaluated by three different individuals on the basis of four parameters: readability, coherence, non-repetition, and informativeness.
- We evaluated and analyzed the summaries attained from several extractive, abstractive (seq2seq) and hybrid baselines summarization models on our new datasets. We found that SumPubMed is more challenging compared to earlier news-based summarization datasets. Thus, SumPubMed, opens new opportunities for further improvement in scientific summarization models.
- We studied the effectiveness of ROUGE (Lin [13]) through Pearson's correlation analysis with human-evaluation scores on SumPubMed. We observed that many variants of ROUGE scores correlate poorly with human evaluation. Our results indicate that ROUGE is possibly not a proper metric for the PubMed text.

The dataset, along with associated scripts, are available at https://github.com/vgupta123/sumpubmed.

2 RELATED WORK

We divide the related work into three primary parts, namely datasets, models, and evaluation metrics:

Datasets. Most summarization datasets are based on news stories (where summaries are mostly hinged on the first few sentences). Recently, a few Social Media and Scientific summarization datasets are proposed. Below, we provide the details of these datasets:

News Summarization: CNN/Daily Mail, a well-known dataset, has manually written highlights (3 to 4 highlights for each document). These highlights were utilized to create short multi-line (maximum 30 to 50 words) summaries for each article. This dataset has, on average, 30 sentences per document and a collection of 92K documents with multiple topics.

DUC (Document Understanding Conference) dataset, is fundamentally a progression of summary evaluations that have been directed by the National Institute of Standards and Technology (years 2001 to 2007). From 2004 onward, DUC has been about multi-document summarization. DUC contains 500 documents (35.6 tokens on average) and summaries (10.4 tokens).

Gigaword, is a news-based summarization dataset (Rush et al. [27]), which is used for the sentence summarization and headline generation task. Here, the input document-summaries are very short, i.e., 31.4 input document tokens and 8.3 tokens for summarization.

X-Sum (Extreme Summarization) (Narayan et al. [20]) is another news summarization dataset that focuses explicitly on abstarctive summarization. The dataset contains online news articles from BBC and one-sentence news summaries. In this dataset, the documents (431 words which is around 20 sentences) and summaries (23 words) are also very short.

Social Media Summarization: Webis-TLDR-17 Corpus (Völske et al. [31]) is a large-scale dataset of 3 Million pairs of content and self-written summaries obtained from social media (Reddit). Webis-Snippet-20 Corpus (Chen et al. [4]) also contains approximately 10 million (webpage content and abstractive snippet) pairs and 3.5 million triples (query terms, abstractive snippets, etc.) for query-based abstractive snippet generation of web pages. However, both of these datasets lack the scientific aspect of the SumPubMed dataset.

Scientific Summarization: (Cohan et al. [5]) released a PubMed (ArXiv) based summarization dataset; however, unlike our dataset no extensive preprocessing pipeline was applied to clean the text. Moreover, we propose several versions of SumPubMed with distinct properties, especially varying summary lengths, article lengths, and vocabulary size. Additionally, the previous dataset only constrains to level-1 section headings as the discourse information, whereas we consider the whole PubMed document (including sub-sections level-2 and below) for summarization in SumPubMed . We performed human annotation on SumPubMed to assess the quality of our dataset, which was not done in the earlier dataset. Lastly, in the *raw text version* of the SumPubMed, our summary length and vocabulary size is larger compared to the previous dataset.

Summarization Models. Text Summarization by extractive models (Erkan and Radev [8], Mihalcea and Tarau [18]) assemble the most important sentences and words from the documents. Abstractive methods (Dohare et al. [7], Gu et al. [11], Liu et al. [14], Nallapati et al. [19], See et al. [28]), on the other hand, require understating the meaning of the text and then a generation process for creating summaries. Most of the abstractive summarization methods use seq2seq encoder-decoder models for the text generation (except AMR-based methods (Dohare et al. [7], Liu et al. [14])).

Evaluation Metrics. ROUGE, a well-known lexical matching metric, evaluates the quality of summaries obtained from a system. The pyramid method (Nenkova et al. [21]) compares the Summarization Count Units (SCUs) between the tuple of the candidate and the reference. Smatch (Cai and Knight [3]), matches the semantic structures, i.e., the Abstract Meaning Representation (AMR) of two

sentences. Recently, a new metric (Ng and Abrecht [22], ShafieiBavani et al. [30]) evaluates the summaries by relying on compositional attributes of corpus-based and lexical resource-based word embeddings. Louis and Nenkova [15], Peyrard et al. [24], Peyrard and Eckle-Kohler [25], Peyrard and Gurevych [26] propose an evaluation metric without reference summaries. Lastly, (Genest et al. [10]) employs a deep model using simple average aggregation.

3 SUMPUBMED CREATION

SumPubMed is created based on biomedical research papers, namely the PubMed database. PubMed is a central repository for 26 million citations, which has literature from MEDLINE, life science journals, and online books. We took a small subset of the research documents and preprocessed them to make them suitable for our summarization task. We downloaded 33,772 documents identified as BMC literature. BMC (BIO MED CENTRAL) literature incorporates BMC health services research papers related to medicine, pharmacy, nursing, dentistry, health care, and so on.

Preprocessing The average word count in the PubMed scientific articles is around 4,000 words for each document and 250 to 300 lines in every document. Therefore, to create SumPubMed, we performed extensive preprocessing so that non-textual content is removed and the overall text is reduced to a more manageable size.

We preprocessed the raw text to make it more appropriate for summarization. The research documents in their natural form contain two subsections: *Front* and *Body*. The front part of the document is basically the abstract, which has three subsections: background, results, and conclusion. The body part of the document is the text, which has three subsections: background, results, and conclusion containing figures, tables, citations, digits, acknowledgments, and references.

We carried out the following prepossessing steps to remove non-textual content from the scientific text.

- We replaced citations and digits in the content with *<cit>* and *<dig>* labels.
- We removed figures, tables, signature, subscript, superscript, and their associated text (e.g., captions).
- We removed acknowledgments and references from the text.

All the preprocessing mentioned above was done on a sentence as a unit utilizing regex library in Python [1].

We then converted all documents to *XML* formats and later utilized *XML* as the resource to create all versions of SumPubMed using a suitable parser. First, we cleaned the text and abstract to create document-summary pairs. Here, the abstracts are utilized as the gold summaries for SumPubMed. Then, we parse the text using the *SAX* parser.[2] In *SAX*, events are triggered when the *XML* is being parsed. When the parser is parsing the *XML* and encounters a tag starting (e.g., *<something>*), then it triggers the *tagStarted* event (actual name of the event might differ). Similarly, when the end of the tag is met while parsing (*</something >*), it triggers *tagEnded*. Using a *SAX* parser implies one needs to handle these events and make sense of the data returned with each event. One could also use the *DOM* parser,[3] where no events are triggered while parsing. In *DOM* the entire *XML* is parsed, and a *DOM* tree (of the nodes in the *XML*) is generated and returned. In general, *DOM* is easier to use but has a huge *overhead* of parsing the entire *XML* before one can start using it; therefore, we use *SAX* instead.

An example of the front part, body part, and the final *XML* file formed from the cleaned text is shown in the Appendix D.[4]

Versions of SumPubMed: We have created four versions of SumPubMed with varying degrees of preprocessing, a) XML, b) Raw Text, c) Non-phrases, and d) Hybrid. In the *XML* version we exported the whole dataset into a single *XML* file. Raw Text is the version just after basic preprocessing is completed (no *XML* parsing). Noun phrases version ensures that the summary and text have the same named entities. This produced a much shorter version of the text and summary than the original pair. We had to use a noun intersection operation on the summary and text since neither the standard Name Entity Recognition (*NER*) (Finkel et al. [9]) nor a Biomedical Named Entity Recognizer (*ABNER*) (Settles [29]) were able to pick out the named entities. The main reason behind *ABNER* insufficiency is due to the novel PubMed named entities that were not covered by any of the classes in the *ABNER* tool. In the Hybrid version, we kept the shorter abstract, i.e., the golden summary, the same but shortened the document (raw text). To shorten the text we used a hybrid (extractive + abstractive summarization) automatic summarization approach. That is, we first performed an extractive summarization utilizing TextRank on the preprocessed text and then carried out the abstractive summarization on the extracted text using a seq2seq (Mi et al. [17]) model. The statistics of all the versions of SumPubMed are available in Table 1.

Version	Statistic	Summary	Article
Raw Text version	Avg. Words	277	4227
	Avg. Sentences	14	203
Noun Phrases version	Avg. Words	223	1578
	Avg. Sentences	10	57
Hybrid version	Avg. Words	223	1891
	Avg. Sentences	10	71

Table 1: Average number of sentences and words in the abstract and text in the three main versions of SumPubMed

Figure 1 shows the complete pipeline with necessary steps needed for SumPubMed creation.

4 HUMAN ANNOTATION OF SUMPUBMED

We distributed 50 randomly chosen summaries to 10 expert annotators (graduate students in NLP) such that we have 3 different human results for each summary. We asked these human-annotators to rate the summaries on a scale of 1 to 10.

We created different document files, each having 10 pairs of summaries where we randomly shuffled between reference and generated summaries with respect to the placement on the page (left or right). Then, human evaluators were asked to evaluate summaries based on the following points:

[1] https://tinyurl.com/ydz3cejh
[2] https://tinyurl.com/y9k2f25b
[3] https://tinyurl.com/py6qxzc
[4] Appendix : https://vgupta123.github.io/docs/sumpubmed_appendix.pdf

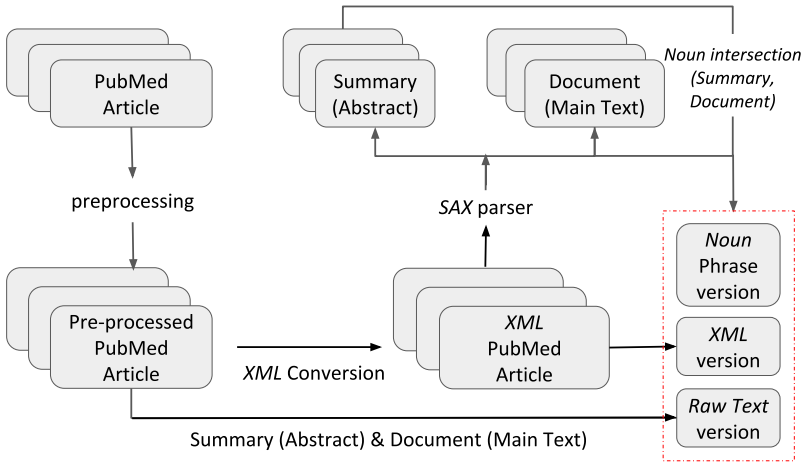

Figure 1: Flowchart highlighting the SumPubMed! creation pipeline.

- *Non-Repetition and no factual Redundancy (Non-Re)*: There should not be redundancy in the factual information and no repetition of sentences.
- *Coherence (Coh)*: Coherence means "continuity of sense". The arguments have to be connected sensibly so that the reader can see consecutive sentences as being about one (or a related) concept.
- *Readability (Read)*: Consideration of general readability criteria such as good spelling, correct grammar, understandability, etc. in the summaries.
- *Informativeness, Overlap and Focus (IOF)*: How much information in one summary is covered by the information in the other summary. That is, one finds the common pieces of information in both/matching the same keywords and key phrases, e.g., "Nematodes" is a keyword present in both summaries. Or noting the frequency of some keywords or key-phrases, and whether both summaries are about the same topic/idea.

The average scores and standard deviations are shown in Table 2.[5] We also received some feedback from the annotators regarding the technical nature of the summaries. Annotators agree that it is hard to compare the informativeness and overlap for most of the summaries presented to them. However, based on the parameters like readability, coherence, and non-repetitiveness the quality of summaries are satisfactory.

Criteria	Mean (μ)	S.D. (σ)
Non-Re	7.19	0.755
Coh	6.87	0.705
Read	6.82	0.821
IOF	6.31	0.879

Table 2: Mean and Standard Deviation (SD) scores of Human-annotation on 50 summaries

[5]Detailed scores: https://tinyurl.com/y9hfp4dd

Correlation between ROUGE and human scores: ROUGE-*N* is an *n*-gram similarity measure that computes unigram, bigram, trigram and higher order *n*-gram overlap. Here, *N* is the length of the *n*-gram, and the count of matching grams is the number of co-occurring *n*-grams between the candidate and reference summaries. We calculated the correlation between ROUGE scores (ROUGE-1 (R-1), ROUGE-2 (R-2) and ROUGE-L (R-L)) in terms of precision, recall and F1 score with the human-evaluated scores. In ROUGE-L (R-L)), L refers to the Longest Common Sub-sequence (LCS) overlap. LCS is a sub-sequence of matching words with maximal length, which is common in both texts with the order of words being preserved. We used Pearson's correlation coefficient (Pearson [23]), which yields a value between -1 and 1. The value demonstrates the degree to which quantitative and continuous variables are directly related.

ROUGE functions on the assumption that a high-quality summary generated by a model should have common words and phrases with a gold-standard summary. But in general that is not the case, since there can be synonymous words and paraphrases of information that is present in the reference summary text. Therefore, merely considering lexical overlaps to decide the quality of a summary is not sufficient to evaluate a summary. In other words, a high ROUGE score may indicate a good summary, but a low ROUGE score does not necessarily indicate a bad summary. While summarizing large documents, humans tend to utilize different paraphrasing/words to convey the same meaning. Lin [13] showed that the high Pearson's correlation between ROUGE scores and human-annotated scores for the DUC dataset denote that ROUGE is a highly reliable and stable summarization metric. However, later Cohan and Goharian [6], Dohare et al. [7] argued that ROUGE is not an accurate estimator of the quality of a summary for scientific input, e.g., bio-medical text. The correlations are shown in Table 3. We see that the correlation values are minimal in magnitude. Thus, we can conclude that ROUGE scores are weakly related with human ratings on the SumPubMed dataset.

Criteria	Prec			Recall			F1		
	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L
Non-Re	-0.09	-0.06	-0.11	+0.02	-0.07	+0.007	+0.008	-0.05	+0.03
Coh	+0.05	-0.14	+0.05	-0.04	-0.25	-0.01	+0.02	-0.19	+0.06
Read	+0.19	+0.09	+0.20	+0.006	-0.03	+0.03	+0.12	+0.01	+0.13
IOF	-0.15	-0.18	-0.16	+0.12	0.08	+0.09	+0.06	-0.007	+0.12

Table 3: Pearson's correlation between ROUGE scores and human ratings on SumPubMed

5 EXPERIMENTS

We have used the Noun phrase version of SumPubMed, in the abstractive summarization settings and the Hybrid version of the SumPubMed dataset in the hybrid setting, i.e., (extractive + abstractive) summarization. We split the dataset into train (93%), test (3%) and validation (4%) sets. Before training, we write a script which firstly, tokenizes all input files and then formed a vocabulary and chunked files for the train, test, and validation sets. This step forms the input in a format that is suitable to be fed to the *seq2seq* models.

Baseline Models: We use the following models on SumPubMed for evaluation: We use extractive, abstractive, and hybrid (extractive + abstractive) automatic summarization methods to evaluate SumPubMed.

(1) **Abstractive Methods** : We use several modifications of *seq2seq* with attention, as described below:

(a) *Seq2Seq with Attention (Nallapati et al. [19])*: The encoder is a single layer bidirectional LSTM, while the decoder is a single layer unidirectional LSTM. Both the encoder and decoder have same sized hidden states, with an attention mechanism over the source hidden states and a soft-max layer over the vocabulary to generate the words. We use the same vocabulary for both the encoding and the decoding phase.

(b) *Seq2Seq with Pointer Generation Networks (See et al. [28])*: The previous model has a computational decoder complexity because each time we have to apply the softmax over the entire vocabulary. The model also outputs an excessive number of UNK tokens (UNK is a special token utilized for out-of-vocabulary words) in the target summary. To address this issue, we use a pointer-generator network (See et al. [28]) which integrates the basic seq2seq model (with attention) with a copying mechanism (Gu et al. [11]). We call this model *seq2seq* for the rest of the paper.

(c) *The seq2Seq model with Pointer Generation Networks and Coverage Mechanism (+cov) (Mi et al. [17])*: The summaries generated by the model discussed before may show repetition, like generating the same arrangement of words multiple times (e.g., "this bioinformatic approach this bioinformatic approach..."). This repetition of phrases is prominent when generating multi-line summaries. The solution to the problem of redundancy in summaries in seq2seq models is the coverage mechanism of Mi et al. [17]. This model penalizes repeated word generations by keeping track of the hitherto covered parts using attention distribution.

(2) **Extractive Methods**: There are several existing approaches to extractive summarization, mostly derived from LexRank (Erkan and Radev [8]), and TextRank (Mihalcea and Tarau [18]). We use TextRank, which is an unsupervised approach for sentence extraction, and has been used successfully in many NLP applications (Hulth [12]).

(3) **Hybrid Methods (Extractive + Abstractive)**: We also experimented with the hybrid approach for summarization. Initially, we used extractive summarization using the TextRank ranking algorithm. We then applied abstractive summarization on the extracted text. We used the pointer-generator networks, followed by the coverage mechanism for the abstractive summarization. In this setting, we have not perfomed any preprocessing before extractive summarization to decrease the length of the documents. The extractive summarization step makes the text length sufficient to apply the abstractive summarization step on it quite easily.

Experimental Setting: We utilized tensorflow (Abadi et al. [1]) for writing our code. We used a single *GeForce GTX TITAN X* with 12GB *GPU* memory for all our training. Training on the full dataset takes on average 5 to 6 days per model. While decoding in seq2seq learning models (for abstractive and hybrid models), we do not take the first decoded sequence. Instead, we use a beam search (Medress et al. [16]) that expands over a greedy search and chooses the most likely word at each time-step to generate the target sequence. Beam search generates all next possible token predictions and keeps the best b sequences, where b is a parameter called the beam width. We used a beam width of four. The hyper-parameters we used for the seq2seq models is listed in Table 4

Exiting Datasets: In this section, we will list some existing datasets (news stories datasets) we utilized for evaluating results on abstractive summarization for comparison. We used the previously mentioned (§2) CNN/Daily Mail. In addition, we used the DUC 2001 dataset for testing our model. The DUC 2001 dataset is made of 50 sets of documents, where each document set contains 10 news articles about a given subject from the New York Times distributed around the 1998s and 2000s. We report the results of abstractive summarization with seq2seq models (and its variants) on these datasets in Table 5.

ROUGE Metrics: We considered ROUGE-1 (R-1) and ROUGE-2 (R-2), and ROUGE-*L* in our experiments. To perform an analysis of the quality of summaries, precision, recall, and F1 score are computed. We used the ROUGE evaluation package `pyrouge` for our studies.[6]

6 RESULTS AND ANALYSIS

ROUGE scores on SumPubMed are given in Table 6, as the length of the target summary increases (increase in the number of decoding

[6]https://pypi.org/project/pyrouge/

Hyper-parameter	Value
LSTM Hidden state size	256
Word embedding dimensions	128
Batch Size	16
encoder steps training	100-1000
encoder steps testing	100-4000
decoder steps length	100-250
beam size	4
learning rate for adagrad	0.15
maximum gradient norm	2.0

Table 4: Hyper-parameters for Sequence to Sequence models

steps), the performance improves. We examined both seq2seq models with and without coverage. The results of the extractive method, i.e., TextRank graph-based ranking Algorithm on SumPubMed is shown in Table 7. We also explored the hybrid approach on the SumPubMed dataset for two models, i.e., TextRank + seq2seq (with and without coverage), the results of which are shown in Table 8.

Analysis: In all three approaches, namely Abstractive in Table 6, Extractive in Table 7 and Extractive + Abstractive in Table 8, we notice that the Recall and F1-score increase with the number of words in the target summaries. We also obtain improved results for 250 words. In addition, we observe that ROUGE scores increase with the length of the generated summary. One possible reason could be that the chances of lexical overlap are more with larger generated summaries. However, precision yields better results for 100 to 150 words in summaries. This is because the fewer are the number of words in the output summary, the higher are the chances of its coverage in the reference summary.

We notice in both Tables 6 and 8 that by adding the coverage (+cov) mechanism, the problem of repetition in summaries is solved to a great extent. The ROUGE scores also show improvement after applying coverage to pointer-generator networks. In Table 9, we note that in terms of precision (Pr), the abstractive approach shows the best results. However, the Recall of the extractive summarization model is always better than abstractive and hybrid approaches. Also, the R-1 Re (ROUGE-1 Recall) and R-L Re (ROUGE-L Recall) for the hybrid models are approximately similar to the abstractive models. We observe that pointer generator networks are effective in handling named entities and out-of-vocabulary words. The coverage mechanism is useful to avoid repetitive generation, which is important for scientific summarization.

7 EXAMPLE OF SUMMARIZATION ON SUMPUBMED

In this section we provide some representative examples of actual summaries. We see factual redundancy and repetitiveness in the generated summaries with pointer-generation which is removed by applying coverage. We also observe that repetitiveness is removed by using the coverage mechanism. Repetitiveness is shown with highlighted text.

Reference Summary the origin of these genes has been attributed to horizontal gene transfer from bacteria, although there still is a lot of uncertainty about the origin and structure of the ancestral ghf <dig> ppn endoglucanase. our data confirm a close relationship between pratylenchus spp. furthermore, based on gene structure data, we inferred a model for the evolution of the ghf <dig> endoglucanase gene structure in plantparasitic nematodes. our evolutionary model for the gene structure in ppn ghf <dig> endoglucanases implies the occurrence of an early duplication event, and more recent gene duplications at genus or species level. the latter one is the first gene isolated from a ppn of a different superfamily -LRB- sphaerularioidea -RRB-; all previously known nematode endoglucanases belong to the superfamily tylenchoidea -LRB- order rhabditida -RRB-. no statistical incongruence between the phylogenetic trees deduced from the catalytic domain and the cbm <dig> was found, which could suggest that both domains have evolved together. and the root knot nematodes, while some radopholus similis endoglucanases are more similar to cyst nematode genes. two new endoglucanases from the migratory nematodes pratylenchus coffeae and ditylenchus africanus were included in this study. phylogenetic analyses were conducted with the ppn ghf <dig> endoglucanases and homologous endoglucanases from bacterial and other eukaryotic lineages such as beetles, fungi and plants. previous studies on the evolution of these genes have focused primarily on data of sedentary nematodes, while in this study, extra data from migratory nematodes were included. it is not clear whether this ancestral endoglucanase consisted of the whole gene cassette, containing a catalytic domain and a carbohydratebinding module -LRB- cbm, type <dig> in ppn and bacteria -RRB- or only of the catalytic domain while the cbm <dig> was retrieved by domain shuffling later in evolution. the ghf <dig> catalytic domain and the cbm <dig> rather than that it evolved by domain shuffling. endo <dig> betaglucanases or cellulases from the glycosyl hydrolase family <dig> -LRB- ghf5 -RRB- have been found in numerous bacteria and fungi, and recently also in higher eukaryotes, particularly in plantparasitic nematodes -LRB- ppn -RRB-.

Summary from seq2seq although the hox cell line with peculiar properties of horizontal gene transfer -LRB- hgt -RRB- has been proposed to delineate the function of ghf <dig> endoglucanases. the gene structure of the ppn ghf <dig> endoglucanases is a matter of debate, in this study no support for an early exon/domain shuffling event after the independent gain of a catalytic domain. the exon/intron gene structure of the ppn ghf <dig> endoglucanases was compared with the multiple protein sequence alignment. the gene structure of the ppn ghf <dig> endoglucanases was compared with the multiple protein sequence alignment. **the gene structure of the ppn ghf <dig> endoglucanases was compared with the multiple protein sequence alignment. the gene structure of the ppn ghf <dig> endoglucanases was compared with the multiple protein sequence alignment. the gene structure of the ppn ghf <dig> endoglucanases was compared with the multiple protein sequence alignment.** the evolution of the gene structure of the ppn ghf <dig> gene families: the number of members from an ancient or early eukaryotic ancestral gene is associated with the expansion of members from an ancient or early eukaryotic ancestral gene. in this study, we.

Summary from seq2seq with coverage the expansion of horizontal gene transfer -LRB- hgt -RRB- events in horizontal gene transfer -LRB- hgt -RRB- has been proposed to explain the origin of ghf <dig> endoglucanases in the nematode kingdom. while the

Data	Model	R-1			R-2			R-L		
		Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
CNN -DM	seq2seq	33.49	38.49	34.61	13.89	15.87	14.29	30.15	34.64	31.15
	+cov	**38.59**	**41.10**	**38.53**	**16.84**	**17.83**	**16.75**	**35.56**	**37.81**	**35.48**
DUC	seq2seq	41.34	21.33	27.63	14.28	7.30	9.49	32.95	16.93	21.93
	+cov	**43.86**	**21.92**	**28.57**	**15.04**	**7.41**	**9.68**	**34.96**	**17.29**	**22.60**

Table 5: ROUGE scores on existing datasets using seq2seq models: CNN-Dailymail (CNN-DM) and DUC 2001 dataset (DUC)

Steps	Model	R-1			R-2			R-L		
		Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
100	seq2seq	52.30	20.56	28.01	16.01	6.17	8.50	47.97	18.70	25.53
	+cov	**57.50**	22.66	31.04	**20.28**	7.74	10.73	**52.62**	20.56	28.23
150	seq2seq	48.88	27.10	32.81	15.18	8.35	10.18	44.64	24.56	29.81
	+cov	55.11	29.71	36.79	19.17	10.14	12.66	50.48	27.07	33.57
200	seq2seq	44.83	30.23	33.79	13.73	9.20	10.33	40.86	27.37	30.65
	+cov	52.86	33.84	39.21	18.25	11.52	13.43	48.47	30.88	35.84
250	seq2seq	41.18	31.84	33.00	12.80	9.79	10.22	37.68	28.89	30.03
	+cov	51.11	**36.24**	**40.13**	17.63	**12.39**	**13.77**	46.92	**33.13**	**36.73**

Table 6: ROUGE scores on *SumPubMed* using a seq2seq model with varying decoding steps

Steps	R-1			R-2			R-L		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
150	**45.91**	31.69	36.82	**16.97**	11.09	13.12	39.12	26.91	28.84
200	42.81	36.03	38.44	15.71	13.31	14.10	36.60	30.73	31.48
250	40.51	**39.59**	**39.33**	14.81	**15.30**	**14.72**	34.83	33.98	**34.83**

Table 7: Results for TextRank an Extractive Summarization approach on *SumPubMed*

Steps	Model	R-1			R-2			R-L		
		Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
100	seq2seq	50.32	21.09	28.45	12.66	5.14	7.04	46.58	19.40	26.23
	+cov	**56.07**	27.42	30.69	**16.65**	6.47	8.95	**51.87**	20.62	28.27
150	seq2seq	45.01	25.50	30.99	11.14	6.21	7.59	41.43	23.35	28.42
	+cov	52.23	29.11	35.62	15.44	8.45	10.42	48.35	26.81	32.86
200	seq2seq	40.55	28.46	31.56	9.93	6.93	7.70	37.21	25.98	28.86
	+cov	47.82	33.37	37.28	14.01	9.68	10.84	44.29	30.80	34.44
250	seq2seq	35.80	30.88	30.61	9.14	7.67	7.66	32.67	27.95	27.80
	+cov	43.82	**36.16**	**37.33**	12.77	**10.49**	**10.85**	40.55	**33.37**	**34.49**

Table 8: ROUGE scores on *SumPubMed* using Hybrid model: TextRank + seq2seq models

ppn ghf <dig> endoglucanases has a close relationship to the root knot nematodes. in order to have a broader overview of the endoglucanase evolution in the infraorder tylenchomorpha, the gene structure of six additional genes was incorporated in our study. the ppn ghf <dig> gene family is associated with the expansion of the ppn ghf <dig> gene family bordered by intron <dig> and intron <dig> although 1 - <dig> symmetrical domains are suggested to be frequently associated with domain shuffling events in the evolution of paralogous gene families: the evolution of the ppn indicate a history of recent duplication events for which little information is available. our model implies that the divergence of the gene structure of the ppn ghf <dig> gene family is notably dynamic, and this evolution involves more intron gains than losses in the order rhabditida -LRB- infraorder tylenchomorpha -RRB-, which is part of one of the three evolutionary independent plantparasitic nematode clades. our results demonstrate that the conserved gene structure of the ppn ghf <dig> endoglucanases and the observation of some sequence conservation in the evolution of the plantparasitic bacteria and nematodes. our results suggest that the evolution of the ghf <dig> gene family is a major consequence of the evolution of.

For examples with attention visualization in abstractive setting, extractive sumamrization examples on SumPubMed and, abstractive summarization examples on CNN/DailyMail refer to the Appendix sections A,B, and C, respectively.[7]

8 CONCLUSION

We created a non-news, SumPubMed, from the PubMed archive to study how various summarization techniques perform on longer scientific texts which have essential information scattered throughout the whole text. In contrast, earlier datasets with news stories appear to mostly have useful information in the first few lines of the document text. Due to the unavailability of any state-of-the-art

[7] Appendix : https://vgupta123.github.io/docs/sumpubmed_appendix.pdf

Model	R-1			R-2			R-L		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
Abstractive	**51.11**	36.24	**40.13**	**17.63**	12.39	**13.77**	**46.92**	33.13	**36.73**
Extractive	40.51	**39.59**	39.33	14.81	**15.30**	14.72	34.83	**33.98**	32.82
Hybrid	43.82	36.16	37.33	12.77	10.49	10.85	40.55	33.37	34.49

Table 9: Comparison of ROUGE scores between various methods on SumPubMed. Here, seq2seq abstractive methods have a target summary length of 250 words

results on this new dataset, we built several baseline models for . We also conducted a human evaluation on aspects such as repetition, readability, coherence, and Informativeness for 50 summaries of 250 words. Each summary is evaluated by 3 different individuals on the basis of four parameters: readability, coherence, non-repetition, and informativeness. To check the significance of our results, we studied the effectiveness of ROUGE through Pearson's correlation analysis with human-evaluation and observed that many variants of ROUGE scores correlate poorly with human evaluation. Our results indicate that ROUGE is possibly not a proper metric for SumPubMed. We also employ extractive, abstractive, and hybrid model baselines to evaluate SumPubMed.

9 ACKNOWLEDGMENT

This project was supported by the Research-I foundation, Computer Science and Engineering Department, IIT Kanpur [8] under the master thesis project (Bharti [2]). Vivek Gupta and Pegah Nokhiz would like to thanks School of Computing, Univeristy of Utah for the needed support and guidance.

REFERENCES

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467* (2016).

[2] Prerna Bharti. 2018. Master Thesis, Text Summarization using Sequence to Sequence Models. , 67 pages.

[3] Shu Cai and Kevin Knight. 2013. Smatch: an Evaluation Metric for Semantic Feature Structures. In *ACL*.

[4] Wei-Fan Chen, Shahbaz Syed, Benno Stein, Matthias Hagen, and Martin Potthast. 2020. Abstractive Snippet Generation. In *Web Conference (WWW 2020)*, Yennung Huang, Irwin King, Tie-Yan Liu, and Maarten van Steen (Eds.). ACM, 1309–1319. https://doi.org/10.1145/3366423.3380206

[5] Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A Discourse-Aware Attention Model for Abstractive Summarization of Long Documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, Vol. 2. 615–621.

[6] Arman Cohan and Nazli Goharian. 2016. Revisiting Summarization Evaluation for Scientific Articles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. 806–813.

[7] Shibhansh Dohare, Harish Karnick, and Vivek Gupta. 2017. Text summarization using abstract meaning representation. *arXiv preprint arXiv:1706.01678* (2017).

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

[9] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd annual meeting on association for computational linguistics*. Association for Computational Linguistics, 363–370.

[10] Pierre-Etienne Genest, Fabrizio Gotti, and Yoshua Bengio. 2011. Deep learning for automatic summary scoring.

[11] Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating Copying Mechanism in Sequence-to-Sequence Learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

[12] Anette Hulth. 2003. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*. Association for Computational Linguistics, 216–223.

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

[14] Fei Liu, Jeffrey Flanigan, Sam Thomson, Norman Sadeh, and Noah A Smith. 2018. Toward abstractive summarization using semantic representations. *arXiv preprint arXiv:1805.10399* (2018).

[15] Annie Louis and Ani Nenkova. 2013. Automatically assessing machine summary content without a gold standard. *Computational Linguistics* 39, 2 (2013), 267–300.

[16] Mark F. Medress, Franklin S Cooper, Jim W. Forgie, CC Green, Dennis H. Klatt, Michael H. O'Malley, Edward P Neuburg, Allen Newell, DR Reddy, B Ritea, et al. 1977. Speech understanding systems: Report of a steering committee. *Artificial Intelligence* 9, 3 (1977), 307–316.

[17] Haitao Mi, Baskaran Sankaran, Zhiguo Wang, and Abe Ittycheriah. 2016. Coverage Embedding Models for Neural Machine Translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. 955–960.

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

[19] Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. 2016. Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*. 280–290.

[20] Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Don't Give Me the Details, Just the Summary! Topic-Aware Convolutional Neural Networks for Extreme Summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 1797–1807.

[21] Ani Nenkova, Rebecca Passonneau, and Kathleen McKeown. 2007. The Pyramid Method: Incorporating Human Content Selection Variation in Summarization Evaluation. *ACM Trans. Speech Lang. Process.* 4, 2, Article 4 (May 2007). https://doi.org/10.1145/1233912.1233913

[22] Jun-Ping Ng and Viktoria Abrecht. 2015. Better summarization evaluation with word embeddings for rouge. *arXiv preprint arXiv:1508.06034* (2015).

[23] Karl Pearson. 1895. VII. Note on regression and inheritance in the case of two parents. *proceedings of the royal society of London* 58, 347-352 (1895), 240–242.

[24] Maxime Peyrard, Teresa Botschen, and Iryna Gurevych. 2017. Learning to Score System Summaries for Better Content Selection Evaluation.. In *Proceedings of the Workshop on New Frontiers in Summarization*. 74–84.

[25] Maxime Peyrard and Judith Eckle-Kohler. 2017. A principled framework for evaluating summarizers: Comparing models of summary quality against human judgments. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Vol. 2. 26–31.

[26] Maxime Peyrard and Iryna Gurevych. 2018. Objective Function Learning to Match Human Judgements for Optimization-Based Summarization. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, Vol. 2. 654–660.

[27] Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A Neural Attention Model for Abstractive Sentence Summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. 379–389.

[28] Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get To The Point: Summarization with Pointer-Generator Networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1073–1083.

[29] Burr Settles. 2005. ABNER: an open source tool for automatically tagging genes, proteins and other entity names in text. *Bioinformatics* 21, 14 (2005), 3191–3192.

[30] Elaheh ShafieiBavani, Mohammad Ebrahimi, Raymond Wong, and Fang Chen. 2018. Summarization Evaluation in the Absence of Human Model Summaries Using the Compositionality of Word Embeddings. In *Proceedings of the 27th International Conference on Computational Linguistics*. 905–914.

[31] Michael Völske, Martin Potthast, Shahbaz Syed, and Benno Stein. 2017. Tl; dr: Mining reddit to learn automatic summarization. In *Proceedings of the Workshop on New Frontiers in Summarization*. 59–63.

[8] https://www.cse.iitk.ac.in/users/rif/