SumSurvey: An Abstractive Dataset of Scientific Survey Papers for Long Document Summarization

Ran Liu[1,2], Ming Liu[3], Min Yu[1,2*], He Zhang[4], Jianguo Jiang[1,2], Gang Li[3], Weiqing Huang[1,2]

[1]Institute of Information Engineering, Chinese Academy of Sciences
[2]School of Cyber Security, University of Chinese Academy of Sciences
[3]School of Information Technology, Deakin University
[4]Zhongtukexin Co., Ltd.
{liuran, yumin, jiangjianguo, huangweiqing}@iie.ac.cn
{m.liu, gang.li}@deakin.edu.au
zhanghe@kxsz.net

Abstract

With the popularity of large language models (LLMs) and their ability to handle longer input documents, there is a growing need for high-quality long document summarization datasets. Although many models already support 16k input, current lengths of summarization datasets are inadequate, and salient information is not evenly distributed. To bridge these gaps, we collect a new summarization dataset called SumSurvey, consisting of more than 18k scientific survey papers. With an average document length exceeding 12k and a quarter exceeding 16k, as well as the uniformity metric outperforming current mainstream long document summarization datasets, SumSurvey brings new challenges and expectations to both fine-tuned models and LLMs. The informativeness of summaries and the models supporting the evaluation of long document summarization warrant further attention. Automatic and human evaluation results on this abstractive dataset confirm this view. Our dataset and code are available at https://github.com/Oswald1997/SumSurvey.

1 Introduction

As one of the major problems in natural language processing, summarization is about processing input documents and generating short texts that contain key information. Its main applications include extracting main events of news, mining opinions on social media, and analyzing comments on shopping websites (Nallapati et al., 2016; Bilal et al., 2022; Angelidis and Lapata, 2018). These tasks use extractive or abstractive models for summarization. The former refers to the extraction of key words or sentences from input to form a summary (Erkan and Radev, 2004; Rossiello et al., 2017), while the latter is to rewrite important information on the basis of understanding (Cho et al., 2022; Liu et al., 2022). Intuitively, abstractive tasks are more challenging, which require models to *read* the input document and present the most salient information within output length limits. In the process of improving and evaluating this capability of summarization models, abstractive datasets play an indispensable role.

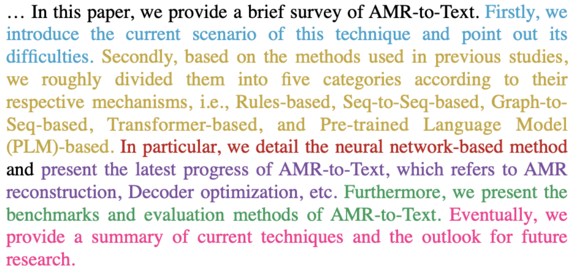

… In this paper, we provide a brief survey of AMR-to-Text. Firstly, we introduce the current scenario of this technique and point out its difficulties. Secondly, based on the methods used in previous studies, we roughly divided them into five categories according to their respective mechanisms, i.e., Rules-based, Seq-to-Seq-based, Graph-to-Seq-based, Transformer-based, and Pre-trained Language Model (PLM)-based. In particular, we detail the neural network-based method and present the latest progress of AMR-to-Text, which refers to AMR reconstruction, Decoder optimization, etc. Furthermore, we present the benchmarks and evaluation methods of AMR-to-Text. Eventually, we provide a summary of current techniques and the outlook for future research.

Figure 1: Sample summary from SumSurvey, each color represent the summary of one section.

In addition, the distribution form of salient information is also an indicator of quality for summarization datasets, because a more uniform distribution means that obvious layout bias cannot be exploited (Koh et al., 2022a). For some tasks, extracting only the first few sentences directly as the final summary can achieve good results, which is common in news domain (Ishikawa et al., 2001). Summarization models need to be able to extract key information distributed throughout the document and deal with relationship among different positions.

With the explosion of data volume and increase of computing resources, language models can gradually deal with longer documents (Beltagy et al., 2020; Guo et al., 2022). However, even with the ability to handle long text, it is still difficult to evaluate the performance of these models due to the different lengths of documents in public datasets.

[*]Corresponding author.

Findings of the Association for Computational Linguistics ACL 2024, pages 9632–9651
August 11-16, 2024 ©2024 Association for Computational Linguistics

The above problems bring a requirement to summarization task: *To construct a new summarization dataset with high abstractiveness and uniformity of salient information distribution, while lengths of documents are in line with the processing capacity of current language models.*

We believe that collecting survey papers is a feasible approach, because such documents are usually long and enumerate many studies in a certain field in detail. Besides, the abstract section naturally summarizes the full paper. Compared with general papers, information in survey papers tend to be more evenly distributed. Thus, we propose a long document summarization dataset called SumSurvey consisting of scientific survey papers. Figure 1 shows a random summary example[1] from SumSurvey, different colored parts indicate that they are generated from different sections, corresponding to contents from chapter I to chapter VI. It can be seen that each section is reflected in this summary, and is presented in a highly general discourse, which is exactly what we need.

We conduct a series of experiments using fine-tuned models (Beltagy et al., 2020; Phang et al., 2023; Guo et al., 2022) and large language models(Du et al., 2022; Zheng et al., 2023). Both automatic and human evaluation are conducted to explore models' ability to handle long documents. Our main contributions are as follows:

- We propose a new long document summarization dataset called SumSurvey, consisting of scientific survey papers. To our knowledge, it is an English summarization dataset with the longest average document length compared with publicly available datasets, and has advantages in abstractiveness, distribution uniformity, summary non-redundancy and compression rate.

- We benckmark various fine-tuned models and LLMs. Both automatic and human evaluation are conducted to comprehensively evaluate the quality of summaries. We also investigate limitations of automatic evaluation on long document summarization.

- We discuss the impact and expectations of SumSurvey on the community. We hope SumSurvey will contribute to evaluating and further improving fine-tuned models on longer inputs, and also expect LLMs to pay more attention to informativeness and factuality. In addition, we hope to see more models that support automatic evaluation of long document summarization.

2 Related Work

Summarization Datasets Datasets are critical to improving and evaluating model performance. For summarization task, there exists many publicly available datasets in various domains. According to the number of source documents in each sample, they can be categorized into single-document (Grusky et al., 2018; Kornilova and Eidelman, 2019) and multi-document (Fabbri et al., 2019; Lu et al., 2020) summarization datasets. From the perspective of source document length, they can be classified into short and long document summarization dataset. Koh et al. (2022b) define datasets that cannot be directly read by pre-trained Transformers as long document datasets. Phang et al. (2023) extend the text length to 4096 tokens and considers it as long input. As many models are being extended to support longer input, currently widely used long document datasets (Cohan et al., 2018; Sharma et al., 2019; Kornilova and Eidelman, 2019) can not meet the demand, and many of them filtered out particularly long documents. Although average length of documents in some datasets is long (Huang et al., 2021), abstractiveness of their summaries is not high (See §3 for more details). In conclusion, there is a lack of long document summarization datasets with high quality, such as high abstractiveness and uniformity of salient information.

Summarization Models Pre-trained language models are mainstream methods for summarization. Some generation models are task-generic and can be fine-tuned to fit summarization scenario (Lewis et al., 2020; Raffel et al., 2020). Other models focus on summarization, researchers design objective functions during pre-training to make models generate higher quality summaries (Zhang et al., 2020a; Xiao et al., 2022). Since these models are based on Transformer (Vaswani et al., 2017), they have quadratic time complexity so cannot adapt well to long document summarization, which has longer input and more topic coverage. The use of efficient attention mechanism can improve the efficiency of processing long documents. For example, Longformer utilizes sparse attention where win-

[1] The random sample is from here

dow has gaps and adds global attention to adapt to different tasks (Beltagy et al., 2020). BigBird combines global, sliding and random attention to reduce quadratic dependency to linear (Zaheer et al., 2020). Zhang et al. (2021) explored three different strategies for long dialogue summarization, and more recent studies investigated the impact of different Transformer modules (Phang et al., 2023) and the use of large language models for long document summarization (Syed et al., 2023; Ravaut et al., 2023) or other tasks (Liu et al., 2023a). In many cases, models can only be evaluated with at most 8k input length limited by current datasets (Manakul and Gales, 2021; Koh et al., 2022b). Therefore, the proposal of our SumSurvey will alleviate this situation.

3 SumSurvey

3.1 Dataset Construction

We crawled all searchable papers on *arxiv.org* which have the word *survey* in their titles[2]. Abstract of each paper is extracted precisely because it can be matched directly from web page. As for main body, we downloaded PDF files and extracted the plain text, then we removed all contents including abstract section and before, and removed contents including references section and after. After cleaning and filtering, we obtain a total of 18,884 samples. The main body of each sample is used as input, and abstract is target. We split our dataset into train (15,108, 80%), validation (1,888, 10%), and test (1,888, 10%) subsets. More details about dataset construction and discussions on data quality are in Appendix A.1 and Appendix A.2.

The average input document length of SumSurvey is 12k, which exceeds existing summarization datasets. More than half of the documents are over 10k in length, in addition, a quarter of documents are longer than 16k, which is the maximum input length supported by many extended models, meaning that our dataset is well suited to evaluating their performance. Papers in SumSurvey spans from 1991 to 2023. Generally, there tends to be more papers in recent years than in the past. These papers come from a variety of subjects, top subjects are *astrophysics*, *machine learning*, *computer vision and pattern recognition*, *artificial intelligence*, *cryptography and security*. The subject distribution conforms to a long-tail distribution. More details about distributions of lengths, years and fields are in Appendix A.3

3.2 Dataset Properties

In this section, we use four indicators to evaluate intrinsic characteristics of datasets. We choose five commonly used English long document datasets for comparison. PubMed and arXiv (Cohan et al., 2018) are from scientific papers. BigPatent (Sharma et al., 2019) consists of records of U.S. patent documents. BillSum (Kornilova and Eidelman, 2019) is a summarization dataset from U.S. Congressional and California state bills. GovReport (Huang et al., 2021) is a collection of reports published by U.S. Government Accountability Office and Congressional Research Service.

Coverage measures the percentage of words in a summary that are part of an extractive fragment from the document. It was proposed by Grusky et al. (2018) and the calculation method is expressed as:

$$Coverage(A, B) = \frac{1}{|B|} \sum_{f \in F(A,B)} |f| \quad (1)$$

where A and B represent the document and its summary respectively. $F(A, B)$ is the set that includes all extractive fragments. $|\cdot|$ is the length of a token sequence. Larger coverage value means more contents are copied directly from document when generating summary.

Density is similar to coverage, where the sum of fragment lengths is changed to the sum of squares of lengths (Grusky et al., 2018):

$$Density(A, B) = \frac{1}{|B|} \sum_{f \in F(A,B)} |f|^2 \quad (2)$$

If length of each fragment is short, the density value will be low, which means that if two summaries have the same coverage value, the one with lower density might have more variability, because its fragments are short rather than long and continuous.

Redundancy is used to evaluate whether sentences in a summary are similar to each other (Bommasani and Cardie, 2020):

$$Redundancy(B) = \underset{(x,y)\in S\times S, x\neq y}{mean} R_l(x, y) \quad (3)$$

In this formula S is sentence set of summary B, (x, y) is a sentence pair. R_l is ROUGE-L F1-score (Lin, 2004). Larger R_l means higher degree of

[2] Deadline is 19th May, 2023.

overlap in a sentence pair. Therefore, redundancy can be used to indicate the extent to which sentences in a summary are redundant. In general, a high-quality summary needs to be as concise as possible.

Uniformity measures the degree to which salient information in a summary is evenly distributed throughout the document:

$$Uniformity(A, B) = -\frac{1}{n}\sum_{i=1}^{m} p_i \log(p_i) \quad (4)$$

We divide source document A equally into m parts and calculate the probability of the salient word[3] in the summary B falling into each of these parts. If different p_i values are close, the final uniformity value will be high.

Table 1 shows coverage, density, redundancy and uniformity scores of several datasets, SumSurvey has the highest overall ranking. To be specific, BigPatent achieves highest scores on coverage and density, closely followed by SumSurvey, which means that less summary contents in these two datasets are extracted from documents, and each single extracted fragment is relatively short in length, thus the possibility of rearranging these fragments is high. We plot heatmaps in Appendix B.1 to show coverage and density more intuitively. Apart from coverage and density, we also calculate proportions of novel n-grams in summaries to further evaluate the abstractiveness of datasets, see Appendix B.2 for results.

SumSurvey has good redundancy score, this is consistent with the characteristics of abstracts in survey papers. In addition, SumSurvey achieves the highest uniformity score, which also benefits from the fact that these documents are derived from survey papers, because typically large amounts of contents scattered throughout the survey paper are about describing previous research and methods, besides, these contents are equally important, so they will be all reflected in the abstract.

4 Experiments

We conduct a series of experiments to evaluate performance of baseline models on SumSurvey.

4.1 Baselines

We use **LED** (Beltagy et al., 2020), **PEGASUS-X** (Phang et al., 2023) and **LongT5** (Guo et al., 2022) as baselines. LED is based on Longformer, it combines a local windowed attention and a task motivated global attention. PEGASUS-X uses staggered block-local Transformer with global encoder tokens. LongT5 integrates attention ideas from ETC and adopts pre-training strategies from PEGASUS. These models support long input at most 16k tokens. More details about these baselines are described in Appendix C.1.

In addition, we evaluate large language models under zero-shot settings. We choose **ChatGPT**, **ChatGLM3** (Du et al., 2022) and **Vicuna** (Zheng et al., 2023), which all have versions that support long inputs. As these models are all instruction-tuned models, they are often capable of generalizing to unseen tasks (Longpre et al., 2023; Chung et al., 2024; Iyer et al., 2022).

4.2 Settings

For the pre-trained summarization models, we use `led-large-16384` [4] and `long-t5-tglobal-base` [5] for summarization of 16k input tokens, while `pegasus-x-large`[6] is adopted for 10k only, because fine-tuning PEGASUS-X under 16k tokens with batch size of 1 requires more than 80GB of GPU memory, which is beyond the computing resources we have. Nevertheless, our length settings still exceed experiments in previous research. For zero-shot LLMs, we use `gpt-3.5-turbo-16k`[7], `chatglm3-6b-32k`[8] and `vicuna-13b-v1.5-16k`[9] for implementation. We evaluate ChatGPT by OpenAI API, while the remaining LLMs are evaluated locally.

We use an NVIDIA A100 80GB PCIe GPU for experiments. Models are fine-tuned for 10 epochs, other parameters vary depending on different experiment settings. Beam search is adopted in inference phase and we set beam size to be 5. Implementation details are in Appendix C.2.

[3] Salient words are extracted by NLTK.

[4] https://huggingface.co/allenai/led-large-16384

[5] https://huggingface.co/google/long-t5-tglobal-base

[6] https://huggingface.co/google/pegasus-x-large

[7] https://platform.openai.com/docs/models/gpt-3-5-turbo

[8] https://huggingface.co/THUDM/chatglm3-6b-32k

[9] https://huggingface.co/lmsys/vicuna-13b-v1.5-16k

dataset	Coverage #rank	Density #rank	Redundancy #rank	Uniformity #rank	avg. #
PubMed	0.893 #2	5.6 #4	0.146 #4	0.896 #5	#3.75
arXiv	0.920 #5	3.7 #3	0.144 #2	0.894 #6	#4.00
BigPatent	**0.861 #1**	**2.1 #1**	0.223 #6	0.922 #3	#2.75
BillSum	0.913 #4	6.6 #5	0.163 #5	0.903 #4	#4.50
GovReport	0.942 #6	7.7 #6	**0.124 #1**	0.932 #2	#3.75
SumSurvey	0.898 #3	3.3 #2	0.144 #2	**0.968 #1**	**#2.00**

Table 1: Intrinsic evaluations of different long document summarization datasets, including values and rankings, calculated on test sets only. Smaller coverage, density and redundancy values are deemed preferable, while larger uniformity values are considered ideal.

5 Results

5.1 Automatic Evaluation

5.1.1 Reference-Based Evaluation

Typically, reference-based methods evaluate summarization models by measuring the similarity between generated summaries and the references. Several evaluation methods are used in this section:

ROUGE We use F1-score of ROUGE-1, ROUGE-2 and ROUGE-L[10], taking into account the completeness, readability and order of summary.

BERTScore BERTScore (Zhang et al., 2020b) computes a similarity score for each token in the candidate summary with each token in the reference summary. The similarity score is calculated using contextual embeddings rather than exact matches like ROUGE. In implementation, roberta-large is used to represent embeddings (Liu et al., 2019).

UniEval-Relevance UniEval (Zhong et al., 2022) reconstructs generation evaluation into a Boolean Question Answering (QA) task. By providing a summary and a reference, it can calculate a relevance score, indicating whether the summary contains only the important information.

ChatGPT and Vicuna We use powerful LLMs to evaluate summary quality. We prompt these models (Appendix C.3) to rate the similarity between the summary and the reference on a scale from 1 to 5.

See Table 2 for results. Large language models overall scored lower on ROUGE compared to fine-tuned models, which can be expected as summaries generated by LLMs tend to be more abstractive. All BERTScore results are quite similar, indicating that this method might not be well-suited for evaluating summarization performance in SumSurvey. In contrast, scores generated by UniEval are more discriminative. Unlike ROUGE, the scores of large language models are consistently higher than fine-tuned models. Among them, ChatGLM3 performs the best, while PEGASUS-X scores notably lower than other models. Evaluation results by both LLMs indicate that the quality of summaries generated by Vicuna is significantly lower than those generated by ChatGPT and ChatGLM, and even lower than fine-tuned models. It seems that language models tend to favor summaries generated by themselves, for example, ChatGPT indicates its summaries are similar in score to those generated by ChatGLM3, while Vicuna suggests there is a significant difference in scores between them.

5.1.2 Reference-Free Evaluation

In this section, we use reference-free metrics to evaluate linguistic quality and abstractiveness of summaries:

UniEval-Fluency By providing a question and a paragraph, the model can evaluate whether the summary is fluent.

ChatGPT and Vicuna LLMs will assign three scores ranging from 1 to 5 for each summary, corresponding to its grammatical accuracy, coherence, and referential clarity.

Novel N-Gram This metric is used to measure the abstractiveness of summaries.

See Table 3 for results of linguistic quality. Text generated by large language models exhibits higher linguistic quality, which is one of the main advantages of LLMs. UniEval considers LLMs to be superior overall to fine-tuned models but fails to distinguish between different models within each category. Evaluation results based on LLMs indicate that summaries generated by Vicuna are lower in all three metrics compared to the other two large

[10] ROUGE-1.5.5 is used for evaluation.

	ROUGE			BERTScore			UniEval	LLM	
model	R-1	R-2	R-l	P	R	F1	relevance	ChatGPT	Vicuna
LED	**43.47**	14.66	23.03	85.85	**86.68**	**86.25**	83.38	3.30	3.02
PEGASUS-X	39.20	13.01	22.28	85.66	84.73	85.17	71.18	-	-
LongT5	42.89	**15.05**	**23.87**	**86.42**	86.11	86.24	82.32	3.12	2.69
ChatGPT	39.16	10.94	20.80	86.31	84.69	85.46	92.63	3.84	3.14
ChatGLM3	36.99	10.24	10.09	84.78	83.97	84.34	**95.10**	**4.02**	**3.64**
Vicuna	33.37	8.35	18.83	86.01	83.05	84.47	90.84	2.92	2.74

Table 2: Results of reference-based evaluation. Best results are **bolded** (statistical significance with p-value < 0.05)

language models as well as LED.

Table 4 shows the proportion of novel n-grams in summaries. Since all three LLMs have not been fine-tuned on SumSurvey, the style of their generated text comes from previous data, therefore they reach a high abstractiveness. Among them, ChatGLM3 exhibits the highest abstractiveness, with its generated summaries being comparable to human-written summaries. PEGASUS-X has lowest proportion of novel n-grams, indicating that it tends to generate summaries on SumSurvey in an extractive way.

5.1.3 Summary

The advantages of automatic evaluation are convenience and speed. However, the results presented by different evaluation methods can sometimes vary greatly, especially adopting reference-based methods. Some reference-free methods use the source document as input to evaluate factuality of generated summaries, for example, UniEval-Consistency, FactCC (Kryscinski et al., 2020) and QuestEval (Scialom et al., 2021). However, most of these methods are based on BERT or T5, and are inefficient in handling long inputs, making it challenging to automatically evaluate factuality in long document summarization. We also attempted to use LLMs that support long inputs to evaluate the informativeness and factuality of summaries, but instability issues often arose. Exploring suitable evaluation methods for long document summarization is a future research direction.

5.2 Human Evaluation

The summaries generated by fine-tuned models depend on learning the mapping relationship between source documents and references during fine-tuning, while the summaries generated by large language models are based on the knowledge learned in previous phases and the designed prompts. Due to the different paradigms, automated evaluation methods may not accurately assess summary quality. For example, ROUGE scores focus on the overlap between generated summaries and references. Fine-tuned models often generate summaries with lower abstractiveness compared to LLMs, which may result in higher scores for fine-tuned models. We aim to alleviate this bias through human evaluation and gain a comprehensive understanding of summarization performance of different models on SumSurvey.

Three graduate students serving as annotators rated summaries generated by different models on four indicators: fluency, coherence, non-redundancy, and informativeness. Considering the workload, we randomly selected 50 samples for evaluation. We formulated human evaluation guidelines to clarify the meanings represented by different indicators (Appendix D).

See Table 5 for human evaluation results. We computed Kendall's coefficient of concordance (Kendall-W) for inter-annotator agreement[11], annotators reached a moderate agreement on fluency and coherence, and reached a substantial agreement on non-redundancy and informativeness. Three large language models scored higher overall in fluency compared to fine-tuned models, consistent with the results in Table 3. However, in terms of coherence, ChatGLM3 performed poorly. Upon inspection, we discovered that summaries generated by ChatGLM3 often lack a cohesive paragraph structure and are instead divided into numerous points, resulting in annotators perceiving them as incoherent. An important point to note is that automatic evaluation methods cannot distinguish fine-grained linguistic quality indicators, whereas human evaluation can differentiate fluency and coherence. In

[11] Kendall-W for fluency, coherence, non-redundancy, informativeness are 0.54, 0.58, 0.68, 0.62 respectively.

model	UniEval Fluency	ChatGPT Gram	 Coherence	 Ref	Vicuna Gram	 Coherence	 Ref
LED	85.69	3.74	3.66	3.48	3.72	3.74	3.74
LongT5	85.93	3.64	3.58	3.32	3.46	3.46	3.46
ChatGPT	92.84	**4.48**	**4.42**	**4.10**	**4.14**	**4.14**	**4.14**
ChatGLM3	92.23	4.20	4.22	4.08	4.08	4.10	4.10
Vicuna	**92.87**	3.50	3.40	3.24	3.62	3.62	3.62

Table 3: Results of linguistic quality.

	% of novel n-grams			
model	uni-	bi-	tri-	4-
LED	7.71	24.83	44.46	57.54
PEG-X	3.55	10.58	19.88	27.58
LongT5	4.67	15.38	29.59	40.54
ChatGPT	7.55	37.69	67.01	82.07
ChatGLM3	**10.88**	**44.71**	**73.70**	**86.46**
vicuna	8.81	39.36	66.49	79.63
Reference	12.68	45.84	73.67	86.16

Table 4: Percentages of novel n-grams in candidate and reference summaries.

model	flu	co	non	in
LED	3.52	2.90	3.76	3.04
LongT5	3.68	2.64	2.52	2.78
ChatGPT	**4.34**	**4.28**	3.42	**3.16**
ChatGLM3	4.30	2.98	**4.10**	3.02
Vicuna	4.18	4.04	3.36	2.66

Table 5: Results of human evaluation. Four indicators are fluency, coherence, non-redundancy, and informativeness. The maximum score is 5.

terms of non-redundancy, LED outperformed ChatGPT and Vicuna, we find that LED-generated summaries are very concise, leading to high score in non-redundancy. As for informativeness, there is no significant distinction between models, as large language models do not demonstrate superiority. Vicuna performs the lowest on this indicator, we later find that summaries generated by Vicuna are relatively short, hence unable to contain much information.

Comparing the results of automatic evaluation and human evaluation, we find that automatic evaluation methods perform relatively well in linguistic quality metrics, closely aligning with human preference, but show significant differences in content-related metrics. This is because many evaluation methods are based on language models, thus they can more accurately evaluate linguistic quality.

6 Discussion

6.1 Is SumSurvey a challenging dataset?

The average length of samples in SumSurvey exceeds 12k, with a quarter of samples exceeding 16k in length. Due to high uniformity of SumSurvey, models find it challenging to utilize positional bias for summarization. Additionally, given the specificity of scientific papers, many domain-specific terms need to be retained when generating summaries. However, our dataset still demonstrates high abstractiveness, necessitating summarization models to possess not only high abstractive capabilities but also the ability to identify and preserve domain-specific terms. This significantly increases the challenges associated with SumSurvey.

Appendix E shows ROUGE scores of some baselines on arXiv and GovReport. Results on SumSurvey are significantly worse than on the other two datasets, especially than on GovReport, which has the lowest coverage and density scores according to Table 1. However, ROUGE scores in Table 2 are still within a reasonable range, documents in SumSurvey are not unsummarizable, they just require models to have better abilities to understand long and complicated text.

We conduct another set of experiments. We use two selection strategies of tokens in training, in addition to the normal way of selecting tokens in a natural order, there is another option called oracle. The oracle selector is implemented by greedily searching sentences that achieve maximum ROUGE-2 Recall till input length limit is reached (Manakul and Gales, 2021). See Table 6 for results. We observe that whether oracle selector is used has little effect on results. When sentences are selected by oracle selector, the mapping between input and output is more natural during training, and it is easier for model to learn how to extract and utilize key features, thereby improving its inference ability. However, the difference between the oracle and normal way on SumSurvey is not obvious, indicating mapping abilities learned by model under these two settings are similar, which means distributions of two types of data are also similar and proves high uniformity and high challenging of SumSurvey.

6.2 What is the impact of SumSurvey on fine-tuned models?

Although many summarization models already support 16k long inputs, they have been limited by previous datasets to only test the performance with inputs of up to 8k in length. With SumSurvey, it is now possible to expand on previous experiments. We use BART (Lewis et al., 2020) and PEGASUS (Zhang et al., 2020a) as base models of LED and PEGASUS-X, and obtained ROUGE scores of different models under varying input lengths.

See Table 6 for results, scores increase with length in most cases, indicating that if only basic models are used by truncating the input, it will lead to information loss. Therefore, it is necessary to use summarization models supporting long document in many scenarios. Besides, when the token length increases from 8k, there is still a significant improvement in results, especially for LED which improves greatly from 8k to 16k. It means that the latter part of SumSurvey still contains a lot of information. More than a quarter of documents in SumSurvey have more than 16k tokens, while current models can not process these documents well, so there is still room for research in the field of long document summarization models.

We hope that our proposed SumSurvey dataset will further evaluate the performance of summarization models on longer inputs and enhance summarization capability through fine-tuning. Additionally, we hope to see more models supporting longer inputs to handle samples in SumSurvey that exceed 16k in length.

6.3 What are the expectations for LLMs?

Summaries generated by LLMs tend to be preferred by humans, for they have fewer grammatical errors and are more fluent and coherent. But these models have a tendency to focus on linguistic aspects but struggle to ensure fidelity to the factual information and alignment with the original source, so it may overfit unconstrained human evaluation, which is affected by annotators' prior, input-agnostic preferences (Atri et al., 2023; Liu et al., 2023b). These findings are consistent with the results in Table 5. In terms of informativeness, LLMs lack an advantage over fine-tuned models. We hope that long survey papers containing rich information can be used to improve information extraction and summarization capabilities of LLMs. After all, while generating summaries that align with human preferences is important, it is also necessary for summaries to contain rich and factual information.

Another expectation for LLMs is their ability to evaluate informativeness and factuality of generated summaries, which requires inputting both the original document and the summary for evaluation. Due to the length of SumSurvey, some models we have tested are unable to stably evaluate informativeness and factuality. We hope to see more models that support automatic evaluation of long document summarization.

model	oracle	R-1 / R-2 / R-L
BART (1k)	✗	36.06 / 10.58 / 20.65
LED (4k)	✗	40.19 / 12.43 / 21.57
LED (8k)	✗	40.89 / 12.78 / 21.70
LED (16k)	✗	**43.47 / 14.66 / 23.03**
PEG (1k)	✗	33.68 / 9.47 / 19.30
PEG-X (4k)	✗	37.05 / 11.59 / 21.38
PEG-X (8k)	✗	38.79 / 12.84 / 22.27
PEG-X (10k)	✗	39.20 / 13.01 / 22.28
BART (1k)	✓	36.59 / 10.27 / 19.50
LED (4k)	✓	39.83 / 11.47 / 20.69
LED (8k)	✓	38.33 / 11.19 / 20.33
LED (16k)	✓	42.49 / 13.75 / 22.42
PEG (1k)	✓	33.66 / 9.52 / 19.28
PEG-X (4k)	✓	36.31 / 10.80 / 20.11
PEG-X (8k)	✓	38.27 / 12.22 / 21.39
PEG-X (10k)	✓	38.31 / 12.40 / 21.52

Table 6: ROUGE scores of baselines on SumSurvey. The oracle column refers to whether to use oracle selector. The lengths used for training and inference are stated in parentheses. PEGASUS and PEGASUS-X are abbreviated to PEG and PEG-X respectively. Best results are **bolded** (statistical significance with p-value $<$ 0.05).

6.4 How does SumSurvey help in creating models supporting inputs exceeding 16k?

For models incapable of handling inputs exceeding 16k tokens, SumSurvey encourages further research into "extract then abstract" and "divide and conquer" approaches. Due to the high uniformity of our dataset, the "extract then abstract" approach tends to lose significant information, resulting in an insignificant performance improvement (as shown in Table 6). Concerning the "divide and conquer" approaches, the even distribution of information makes such methods feasible. We anticipate the emergence of better hierarchical models and feature fusion models.

For the creation of models supporting inputs exceeding 16k tokens, SumSurvey encourages the development of more efficient attention mechanisms. Since these models can access the entire document, better summarization performance can be achieved theoretically. However, this necessitates models with both long-context modeling capabilities and efficient processing techniques, which poses greater challenges.

7 Conclusion

We propose a new long document summarization dataset SumSurvey consisting of scientific survey papers. The average length of documents in SumSurvey is longer than publicly available summarization dataset, and it has higher abstractiveness. In addition, salient information is more evenly distributed throughout documents. By benchmarking baseline models using automatic and human evaluation, we have a comprehensive view of how these models perform on SumSurvey. We hope SumSurvey will contribute to evaluating and further improving fine-tuned models on longer inputs, and also expect LLMs to pay more attention to informativeness and factuality. Moreover, we look forward to more comprehensive automatic evaluation models supporting long documents.

Limitations

Upon examining the metadata, it was confirmed that over two-thirds of the documents from SumSurvey contain publication information. For those lacking such information, a manual inspection of a subset revealed that the vast majority of samples actually originate from published or peer-reviewed papers, with only a negligible fraction being unpublished, which is insufficient to significantly affect the overall quality. However, due to the dataset's diverse range of fields, including many that are unfamiliar to us, manually checking the text quality of all documents presents a considerable challenge.

We did not benchmark all baselines because: 1) limited by computation power, some models cannot be fine-tuned on a single NVIDIA A100 with 80GB GPU memory, such as pegasus-x-large and long-t5-tglobal-large with 16k tokens; 2) some models are not publicly available at the moment (Rohde et al., 2021; Pang et al., 2023); 3) some large language models like LongAlpaca (Chen et al., 2023) exhibited instability on SumSurvey, often failing to generate outputs properly.

Documents in SumSurvey lack language diversity. The fact that English is the dominant language in academia makes it a single-language dataset.

Ethics Statement

Data collection approval was received from an ethics review board. Remuneration we paid to the annotators is above the average salary level in the area where the annotators are located. Since documents in our dataset are scientific papers, it is unlikely that offensive contents are included. However, not all papers in arXiv have been published, so they may contain unacceptable conclusions or non-original ideas. All codes and data used in this paper comply with the license for use.

Acknowledgments

This work is supported by Youth Innovation Promotion Association CAS (No.2021155).

References

Stefanos Angelidis and Mirella Lapata. 2018. Summarizing opinions: Aspect extraction meets sentiment prediction and they are both weakly supervised. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3675–3686, Brussels, Belgium. Association for Computational Linguistics.

Yash Atri, Arun Iyer, Tanmoy Chakraborty, and Vikram Goyal. 2023. Promoting topic coherence and inter-document consorts in multi-document summarization via simplicial complex and sheaf graph. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2154–2166, Singapore. Association for Computational Linguistics.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv:2004.05150*.

Iman Munire Bilal, Bo Wang, Adam Tsakalidis, Dong Nguyen, Rob Procter, and Maria Liakata. 2022. Template-based abstractive microblog opinion summarization. *Transactions of the Association for Computational Linguistics*, 10:1229–1248.

Rishi Bommasani and Claire Cardie. 2020. Intrinsic evaluation of summarization datasets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8075–8096, Online. Association for Computational Linguistics.

Mingda Chen, Zewei Chu, Sam Wiseman, and Kevin Gimpel. 2022. SummScreen: A dataset for abstractive screenplay summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8602–8615, Dublin, Ireland. Association for Computational Linguistics.

Yukang Chen, Shaozuo Yu, Shengju Qian, Haotian Tang, Xin Lai, Zhijian Liu, Song Han, and Jiaya Jia. 2023. Long alpaca: Long-context instruction-following models. `https://github.com/dvlab-research/LongLoRA`.

Sangwoo Cho, Kaiqiang Song, Xiaoyang Wang, Fei Liu, and Dong Yu. 2022. Toward unifying text segmentation and long document summarization. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 106–118, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

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)*, pages 615–621, New Orleans, Louisiana. Association for Computational Linguistics.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. GLM: General language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 320–335, Dublin, Ireland. Association for Computational Linguistics.

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

Alexander Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. 2019. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1074–1084, Florence, Italy. Association for Computational Linguistics.

Max Grusky, Mor Naaman, and Yoav Artzi. 2018. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 708–719, New Orleans, Louisiana. Association for Computational Linguistics.

Mandy Guo, Joshua Ainslie, David Uthus, Santiago Ontanon, Jianmo Ni, Yun-Hsuan Sung, and Yinfei Yang. 2022. LongT5: Efficient text-to-text transformer for long sequences. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 724–736, Seattle, United States. Association for Computational Linguistics.

Luyang Huang, Shuyang Cao, Nikolaus Parulian, Heng Ji, and Lu Wang. 2021. Efficient attentions for long document summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1419–1436, Online. Association for Computational Linguistics.

Kai Ishikawa, Shinichi Ando, and Akitoshi Okumura. 2001. Hybrid text summarization method based on the tf method and the lead method. In *Proceedings of the 2nd National Institute of Informatics Test Collection Information Retrieval (NTCIR) Workshop*.

Srinivasan Iyer, Xi Victoria Lin, Ramakanth Pasunuru, Todor Mihaylov, Daniel Simig, Ping Yu, Kurt Shuster, Tianlu Wang, Qing Liu, Punit Singh Koura, et al. 2022. Opt-iml: Scaling language model instruction meta learning through the lens of generalization. *arXiv preprint arXiv:2212.12017*.

Huan Yee Koh, Jiaxin Ju, Ming Liu, and Shirui Pan. 2022a. An empirical survey on long document summarization: Datasets, models, and metrics. *ACM computing surveys*, 55(8):1–35.

Huan Yee Koh, Jiaxin Ju, He Zhang, Ming Liu, and Shirui Pan. 2022b. How far are we from robust long abstractive summarization? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2682–2698, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Anastassia Kornilova and Vladimir Eidelman. 2019. BillSum: A corpus for automatic summarization of US legislation. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 48–56, Hong Kong, China. Association for Computational Linguistics.

Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346, Online. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023a. Lost in the middle: How language models use long contexts. *arXiv preprint arXiv:2307.03172*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Yixin Liu, Alex Fabbri, Pengfei Liu, Yilun Zhao, Linyong Nan, Ruilin Han, Simeng Han, Shafiq Joty, Chien-Sheng Wu, Caiming Xiong, and Dragomir Radev. 2023b. Revisiting the gold standard: Grounding summarization evaluation with robust human evaluation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4140–4170, Toronto, Canada. Association for Computational Linguistics.

Yixin Liu, Ansong Ni, Linyong Nan, Budhaditya Deb, Chenguang Zhu, Ahmed Hassan Awadallah, and Dragomir Radev. 2022. Leveraging locality in abstractive text summarization. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6081–6093, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V. Le, Barret Zoph, Jason Wei, and Adam Roberts. 2023. The flan collection: designing data and methods for effective instruction tuning. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Yao Lu, Yue Dong, and Laurent Charlin. 2020. Multi-XScience: A large-scale dataset for extreme multi-document summarization of scientific articles. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8068–8074, Online. Association for Computational Linguistics.

Potsawee Manakul and Mark Gales. 2021. Long-span summarization via local attention and content selection. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6026–6041, Online. Association for Computational Linguistics.

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*, pages 280–290, Berlin, Germany. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Bo Pang, Erik Nijkamp, Wojciech Kryscinski, Silvio Savarese, Yingbo Zhou, and Caiming Xiong. 2023. Long document summarization with top-down and bottom-up inference. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1267–1284, Dubrovnik, Croatia. Association for Computational Linguistics.

Jason Phang, Yao Zhao, and Peter Liu. 2023. Investigating efficiently extending transformers for long input summarization. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3946–3961, Singapore. Association for Computational Linguistics.

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. *Journal of Machine Learning Research*, 21(140):1–67.

Mathieu Ravaut, Shafiq Joty, Aixin Sun, and Nancy F Chen. 2023. On position bias in summarization with large language models. *arXiv preprint arXiv:2310.10570*.

Tobias Rohde, Xiaoxia Wu, and Yinhan Liu. 2021. Hierarchical learning for generation with long source sequences. *arXiv preprint arXiv:2104.07545*.

Gaetano Rossiello, Pierpaolo Basile, and Giovanni Semeraro. 2017. Centroid-based text summarization through compositionality of word embeddings. In *Proceedings of the MultiLing 2017 Workshop on Summarization and Summary Evaluation Across Source Types and Genres*, pages 12–21, Valencia, Spain. Association for Computational Linguistics.

Thomas Scialom, Paul-Alexis Dray, Sylvain Lamprier, Benjamin Piwowarski, Jacopo Staiano, Alex Wang, and Patrick Gallinari. 2021. QuestEval: Summarization asks for fact-based evaluation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6594–6604, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Eva Sharma, Chen Li, and Lu Wang. 2019. BIGPATENT: A large-scale dataset for abstractive and coherent summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2204–2213, Florence, Italy. Association for Computational Linguistics.

Shahbaz Syed, Dominik Schwabe, Khalid Al-Khatib, and Martin Potthast. 2023. Indicative summarization of long discussions. *arXiv preprint arXiv:2311.01882*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Alex Wang, Richard Yuanzhe Pang, Angelica Chen, Jason Phang, and Samuel R. Bowman. 2022. SQuALITY: Building a long-document summarization dataset the hard way. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1139–1156, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Wen Xiao, Iz Beltagy, Giuseppe Carenini, and Arman Cohan. 2022. PRIMERA: Pyramid-based masked sentence pre-training for multi-document summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5245–5263, Dublin, Ireland. Association for Computational Linguistics.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. Big bird: Transformers for longer sequences. In *Advances in Neural Information Processing Systems*, volume 33, pages 17283–17297. Curran Associates, Inc.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020a. PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 11328–11339. PMLR.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2020b. Bertscore: Evaluating text generation with bert. In *Proceedings of the 8th International Conference on Learning Representations*.

Yusen Zhang, Ansong Ni, Tao Yu, Rui Zhang, Chenguang Zhu, Budhaditya Deb, Asli Celikyilmaz, Ahmed Hassan Awadallah, and Dragomir Radev. 2021. An exploratory study on long dialogue summarization: What works and what's next. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4426–4433, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*.

Ming Zhong, Yang Liu, Da Yin, Yuning Mao, Yizhu Jiao, Pengfei Liu, Chenguang Zhu, Heng Ji, and Jiawei Han. 2022. Towards a unified multidimensional evaluator for text generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2023–2038, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

A Dataset details

A.1 Construction Details

The data we have collected falls into two categories: metadata extracted directly from websites and downloaded PDF files. Regarding metadata, it contains a wealth of information, including authors, abstracts, subjects, initial and latest submission times, acceptance and publication information, and more. For PDF files, pdfminer[12] and RE rules were utilized to extract the plain text and split it into a list of tuples, where each tuple contains a section header and the corresponding text. We excluded the abstract section and content before it, as well as the reference section and content after it. To enhance the accuracy of abstract matching and removal, we employed BERT-based Sentence Transformers[13]. This allowed us to calculate the similarity between the abstract identified using RE rules and the actual abstract from metadata. Subsequently, we filtered out certain outlier data based on this similarity measure. In this above process, plain text of tables was retained while figures were removed, but figure captions were preserved for summarization models to utilize this information. We kept the original citation format of the source document, but the reference section was removed.

A.2 Dataset Quality

In this section, we discuss whether papers in SumSurvey are of high quality and whether SumSurvey is a different dataset.

Regarding the former, both the "Journal ref" field and the "Comments" field in the extracted metadata contain information about the paper's acceptance and publication information. By examining the information, we observed that more than two-thirds of papers contain information regarding publication, and most papers without such information were, in fact, published. Therefore, we confirm that the vast majority of papers in the dataset are high-quality, published survey papers.

As for the latter, we compared SumSurvey with the existing arXiv dataset for they are both scientific paper datasets. We examined 6440 samples from the test set of arXiv dataset, and used arXiv API[14] to obtain additional information not provided by the arXiv dataset. We found that papers with "survey" in their titles account for approximately one percent. Based on this proportion, we estimate that there are no more than 3000 survey papers in the entire arXiv dataset, equivalent to approximately 15% of SumSurvey in terms of sample size. It is worth noting that papers in the old arXiv dataset are all dated before 2018. Additionally, we calculated that the average length of survey papers in arXiv dataset is only 6k, approximately half of SumSurvey, as the authors of the arXiv dataset have filtered out excessively long papers.

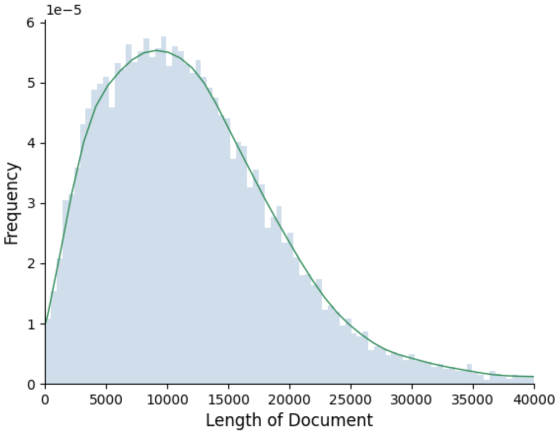


Figure 2: Histogram and density curve of document length in SumSurvey.

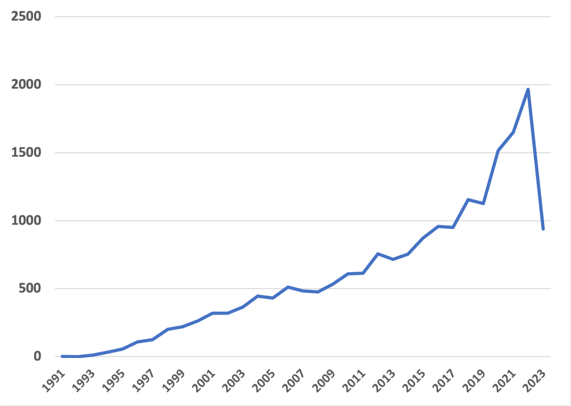


Figure 3: Year distribution of papers in SumSurvey. We use latest submit times for statistics.

A.3 Dataset Distributions

Length Distribution Figure 2 is length distribution of documents in SumSurvey. Table 7 shows length statistics in different datasets. The input document length of our SumSurvey exceeds other datasets, and summary length is in a reasonable range.

Year Distribution See Figure 3 for year distribution of papers in SumSurvey. The more recent the year, the greater the number of papers. This indi-

[12] https://github.com/euske/pdfminer

[13] https://github.com/UKPLab/sentence-transformers

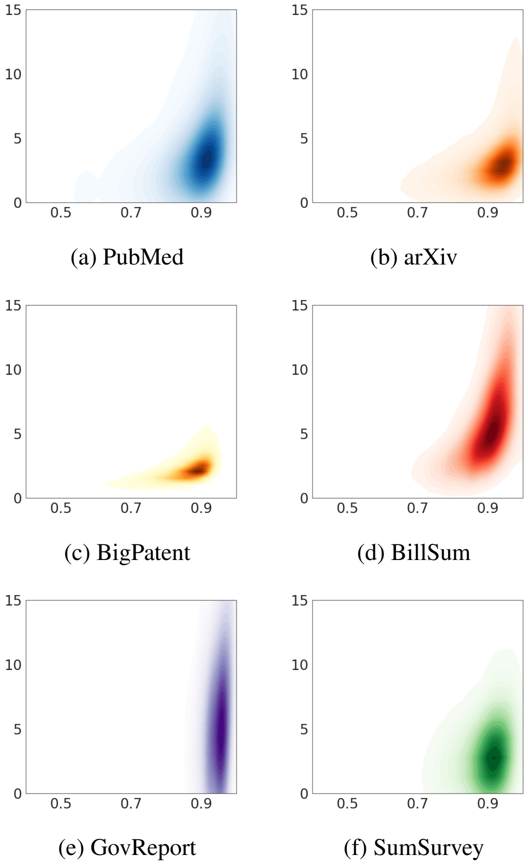
[14] https://info.arxiv.org/help/api/basics.html

dataset	summ sentences	doc sentences	summ tokens	doc tokens
PubMed	7.1	102	208	3143
arXiv	6.3	251	242	6446
BigPatent	3.6	143	117	3573
BillSum	7.1	42	243	1686
GovReport	21.4	300	607	9409
SumSurvey	8.1	413	236	**12532**

Table 7: Length statistics of different long document summarization datasets.

cates that the data in our dataset is relatively recent.

Field Distribution There are over 100 types of tags in total, and each sample typically has multiple tags. Representative fields include Astrophysics, Machine Learning, Computer Vision and Pattern Recognition, Artificial Intelligence, Cryptography and Security. In addition , there are other fields such as Signal Processing, Methodology, Algebraic Geometry, General Finance, and more. The field distribution conforms to a long-tail distribution.



(a) PubMed

(b) arXiv

(c) BigPatent

(d) BillSum

(e) GovReport

(f) SumSurvey

Figure 4: Heatmaps of different datasets, x-axis represents coverage and y-axis represents density.

B Intrinsic Characteristics

B.1 Heatmaps

Heatmaps in Figure 4 show coverage and density of different datasets. BillSum and GovReport perform the worst, while BigPatent is the best. Note that the average document length of BigPatent is much smaller than SumSurvey, making it less necessary for humans to extract many contents directly from the document when generating the summary, because of the reduced workload and the fact that the summary and corresponding short document will be too similar if lots of contents are copied (see scores of GovReport for example). In addition, scope of its heatmap is too narrow, indicating that writing styles of these summaries are highly consistent. PubMed, arXiv and SumSurvey are all consists of scientific papers, while the heatmap of SumSurvey is more regular in shape, proving that papers we collected are indeed in a same category, which is survey. As for PubMed and arXiv, particularly long documents have been filtered out (Cohan et al., 2018), so it is likely that many survey papers are lost.

B.2 Novel N-grams

The proportion of novel n-grams in a summary reflects the degree of abstractiveness of this sample. We calculate percentages of uni-, bi-, tri-, and 4-grams in summaries for long document summarization datasets, we also introduce two more long-input datasets SummScreen (Chen et al., 2022) and SQuALITY (Wang et al., 2022), see Table 8 for results.

As shown in Table 8, SumSurvey has higher abstractiveness than arXiv, PubMed, BillSum and GovReport, especially on tri-grams and 4-grams, this proves again that SumSurvey has good density scores (see Table 1). SummScreen and SQuALITY have an advantage in abstractiveness as their data is from TV programs or stories, making the text filled with colloquialism and lack technical

	% of novel n-grams			
dataset	uni-	bi-	tri-	4-
PubMed	12.4	44.0	65.3	76.0
arXiv	9.5	41.0	66.4	79.6
BigPatent	13.5	52.6	78.3	89.5
BillSum	10.4	38.2	55.2	65.6
GovReport	5.7	32.7	56.3	68.9
SumSurvey	12.1	45.3	72.5	85.3
SummScreen-FD	18.4	70.1	94.4	98.7
SumSurvey-TMS	13.5	65.9	93.1	97.9
SQuALITY	17.5	65.9	92.7	98.0

Table 8: Percentages of novel n-grams in summaries of different datasets, results of SummScreen are from Chen et al. (2022).

terms. So, there is a broader range of word choices when writing abstracts. Regarding SumSurvey, the abundance of technical terms naturally puts it at a disadvantage in abstractiveness. Small data size of SQuALITY makes it difficult to use for fine-tuning models to improve long document summarization performance. Besides, the average length of both datasets is only around 5k-6k and they cannot be used to evaluate the ability of summarization models to handle longer inputs.

C Experiment Details

C.1 Baselines Description

BART (Lewis et al., 2020) is a denoising autoencoder for pre-training sequence-to-sequence models. By destroying and then reconstructing text, BART has the flexibility to process raw text and learn to reconstruct it effectively. The fine-tuned BART performs well on summarization task.

LED (Beltagy et al., 2020) is based on Longformer, which replaces the standard self-attention in Transformer with combination of a local windowed attention and a task motivated global attention. A token with a local attention attends to its context tokens while a global attention token attends to all tokens across the sequence. LED is a Longformer variant supporting long input up to 16k tokens, and its parameters are initialized from BART.

PEGASUS (Zhang et al., 2020a) uses a new objective called Gap Sentences Generation (GSG) during pre-training designed specifically for summarization task. The authors select sentences that are important to document and mask them, then train PEGASUS to generate these sentences.

PEGASUS-X (Phang et al., 2023) is an extension of the PEGASUS model supporting long input at most 16k tokens. The authors design staggered block-local Transformer with global encoder tokens, where a block-local attention token can only attend to other tokens within the same block and the block allocation across alternating layers is staggered. Additionally, long documents are included during pre-training to improve downstream summarization performance.

LongT5 (Guo et al., 2022) integrates attention ideas from ETC, and adopt pre-training strategies from PEGASUS into the scalable T5 architecture. It uses a new attention mechanism called Transient Global (TGlobal), which mimics ETC's local/global attention mechanism, but without requiring additional side-inputs.

C.2 Implementation Details

BART We use fairseq (Ott et al., 2019) for implementation with Python version of 3.8 and PyTorch version of 2.0.1. Learning rate is set to 1e-4. Batch size is set to 2 with update frequency of 8. Other parameters like label smoothing, dropout, attention dropout, weight decay, clip norm are consistent with the official example[15].

LED We use Transformers (Wolf et al., 2020) as framework with version of 4.31.0. We follow this fine-tuning notebook[16] with learning rate of 1e-4.

PEGASUS Transformers is used as framework for implementation and pytorch-lightning[17] version is 1.0.4. We set learning rate to 1e-5 and batch size to 6.

PEGASUS-X It is implemented on Flax[18] with version of 0.6.11 and JAX[19] version of 0.4.13. Learning rate is set to 1e-4. Batch size are 2, 1, 1 when maximum input length are 4k, 8k, 10k respectively.

LongT5 We use Transformers for implementation. Learning rate is set to 5e-4, and batch size is 1 with gradient accumulation step of 32.

ChatGPT We use OpenAI API[20] for implementation with prompt of *"Please summarize the following contents with no more than 256 words: "*, and

[15] https://github.com/facebookresearch/fairseq/blob/main/examples/bart/README.summarization.md

[16] https://colab.research.google.com/drive/12LjJazBl7Gam0XBPy_y0CTOJZeZ34c2v

[17] https://github.com/Lightning-AI/pytorch-lightning

[18] https://github.com/google/flax

[19] https://github.com/google/jax

[20] https://platform.openai.com

refer to official use case[21] for parameter settings.
ChatGLM3 We use Transformer for implementation. The prompt is the same as used by ChatGPT.
Vicuna We use FastChat[22] as evaluation framework. The prompt is the same as used by ChatGPT.

For LLMs, experiments are under zero-shot settings. While for other baselines, the fine-tuning time varies from a few hours to two days.

C.3 LLMs for Evaluation

We use ChatGPT and Vicuna to evaluate summary quality. Prompt used in §5.1.1 is *"I will provide you with five different summaries of the same document generated by different models, as well as a manually generated standard summary. Please rate each of the five model-generated summaries on a scale of one to five in terms of similarity to the standard summary."*. And prompt used in §5.1.2 is *"I will provide you with five different summaries of the same document generated by different models. Please rate each of the five model-generated summaries on a scale of one to five in terms of grammatical accuracy, coherence, and clarity of reference. "*

D Human Evaluation Guidelines

Three annotators[23] scores summaries independently, they need to complete 50 subtasks, each of which consists of source document and five summaries generated by LED, LongT5, ChatGPT, ChatGLM3 and Vicuna, respectively. All summaries are lowercased and tokenized, making it impossible to use these features to determine which summaries are likely generated by the same model. We have developed a guideline for annotators, see Fig 5.

E Results on other datasets

We show some results of baselines on other datasets in Table 9. On the one hand, limited by these datasets, experiments cannot be conducted with longer input length; on the other hand, scores on these datasets are higher than those on SumSurvey, indicating strong challenges of our dataset. Among them, scores of baselines on GovReport are particularly high, because GovReport has the lowest abstractiveness (see Table 1 and Table 8), therefore, models tend to generate summaries in an extractive way.

F Case Study

After checking the summaries generated by baselines, we find that even as abstractive fine-tuned models, they tend to generate summaries in an extractive way when facing long document summarization task. This may be because documents in SumSurvey are too long for models to understand-then-reword, so summaries are generated by identifying salient sentences.

See Figure 6 and Figure 7 for examples. The summary generated by LED is more precise, while PEGASUS-X and LongT5 generates some examples, resulting in a bit of verbosity. The summary generated by ChatGPT has higher abstractiveness and contains some information that could easily be ignored. The summary generated by ChatGLM3 is not a complete paragraphin, thus achieving a low coherence score in human evaluation. Typically, even when given length constraint in the prompt, the summary generated by Vicuna is still relatively short, resulting in low informativeness.

G Software and Licenses

Data and codes used in this paper are:

- arXiv, Misc[24]
- BERTScore, MIT[25]
- Datasets, Apache-2.0[26]
- fariseq, MIT[27]
- Flax, Apache-2.0[28]
- Jax, Apache-2.0[29]
- matplotlib, Misc[30]

[21] `https://platform.openai.com/docs/api-reference/chat/create`

[22] `https://github.com/lm-sys/FastChat`

[23] Graduate students, two of whom are non-native English speakers and one is a native English speaker.

[24] `https://info.arxiv.org/help/license/index.html`

[25] `https://github.com/Tiiiger/bert_score/blob/master/LICENSE`

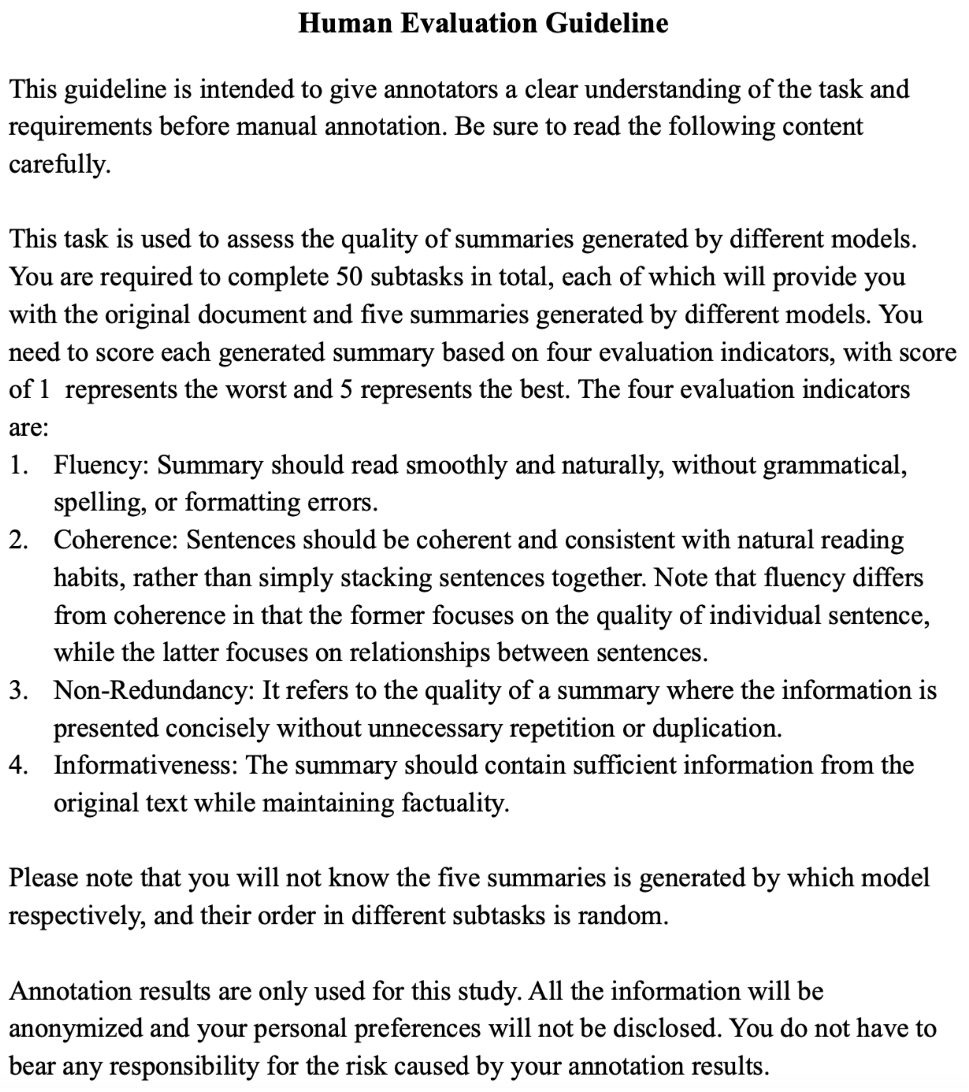
[26] `https://github.com/huggingface/datasets/blob/main/LICENSE`

[27] `https://github.com/facebookresearch/fairseq/blob/main/LICENSE`

[28] `https://github.com/google/flax/blob/main/LICENSE`

[29] `https://github.com/google/jax/blob/main/LICENSE`

[30] `https://github.com/matplotlib/matplotlib/blob/main/LICENSE/LICENSE`



Human Evaluation Guideline

This guideline is intended to give annotators a clear understanding of the task and requirements before manual annotation. Be sure to read the following content carefully.

This task is used to assess the quality of summaries generated by different models. You are required to complete 50 subtasks in total, each of which will provide you with the original document and five summaries generated by different models. You need to score each generated summary based on four evaluation indicators, with score of 1 represents the worst and 5 represents the best. The four evaluation indicators are:

1. Fluency: Summary should read smoothly and naturally, without grammatical, spelling, or formatting errors.
2. Coherence: Sentences should be coherent and consistent with natural reading habits, rather than simply stacking sentences together. Note that fluency differs from coherence in that the former focuses on the quality of individual sentence, while the latter focuses on relationships between sentences.
3. Non-Redundancy: It refers to the quality of a summary where the information is presented concisely without unnecessary repetition or duplication.
4. Informativeness: The summary should contain sufficient information from the original text while maintaining factuality.

Please note that you will not know the five summaries is generated by which model respectively, and their order in different subtasks is random.

Annotation results are only used for this study. All the information will be anonymized and your personal preferences will not be disclosed. You do not have to bear any responsibility for the risk caused by your annotation results.

Figure 5: Human evaluation guideline

		arXiv			GovReport		
model	length	R-1	R-2	R-L	R-1	R-2	R-L
BART	1K	43.84	16.55	39.86	56.55	26.7	54.46
BART + Longformer (LED)	4K	45.72	18.48	41.82	57.45	28.14	55.40
BART + Longformer (LED)	8K	46.60	19.05	42.21	58.35	28.78	56.35
PEGASUS	1K	44.17	17.16	40.18	57.19	27.87	55.17
PEGASUS + Longformer	4K	46.02	18.33	42.28	58.35	28.78	56.35
PEGASUS + Longformer	8K	46.87	19.73	42.36	58.59	29.02	56.29

Table 9: ROUGE results on arXiv and GovReport. Baselines include base models (BART, PEGASUS) and their long input versions extended by Longformer. Results are from Koh et al. (2022b).

- NLTK, Apache-2.0[31]
- NumPy, BSD-3-Clause[32]
- PDFMiner, MIT[33]
- PubMed, arXiv dataset, Apache-2.0[34]
- Pytorch Lightning, Apache-2.0[35]
- PyTorch, Misc[36]
- ROUGE Metric, MIT[37]
- scikit-learn, BSD-3-Clause[38]
- SciPy, BSD-3-Clause[39]
- Scrapy, BSD-3-Clause[40]
- seaborn, BSD-3-Clause[41]
- Sentence Transformers, Apache-2.0[42]
- spaCy, MIT[43]
- Stanza, Apache-2.0[44]
- TensorFlow, Apache-2.0[45]
- Transformers, Apache-2.0[46]
- UniEval, MIT[47]

[31] https://github.com/nltk/nltk/blob/develop/LICENSE.txt

[32] https://github.com/numpy/numpy/blob/main/LICENSE.txt

[33] https://github.com/euske/pdfminer/blob/master/LICENSE

[34] https://github.com/armancohan/long-summarization/blob/master/LICENSE

[35] https://github.com/Lightning-AI/pytorch-lightning/blob/master/LICENSE

[36] https://github.com/pytorch/pytorch/blob/main/LICENSE

[37] https://github.com/li-plus/rouge-metric/blob/master/LICENSE

[38] https://github.com/scikit-learn/scikit-learn/blob/main/COPYING

[39] https://github.com/scipy/scipy/blob/main/LICENSE.txt

[40] https://github.com/scrapy/scrapy/blob/master/LICENSE

[41] https://github.com/mwaskom/seaborn/blob/master/LICENSE.md

[42] https://github.com/UKPLab/sentence-transformers/blob/master/LICENSE

[43] https://github.com/explosion/spaCy/blob/master/LICENSE

[44] https://github.com/stanfordnlp/stanza/blob/main/LICENSE

[45] https://github.com/tensorflow/tensorflow/blob/master/LICENSE

[46] https://github.com/huggingface/transformers/blob/main/LICENSE

[47] https://github.com/maszhongming/UniEval/blob/main/LICENSE

Reference

Recent years have witnessed a surge of publications aimed at tracing temporal changes in lexical semantics using distributional methods, particularly prediction-based word embedding models. However, this vein of research lacks the cohesion, common terminology and shared practices of more established areas of natural language processing. In this paper, we survey the current state of academic research related to diachronic word embeddings and semantic shifts detection. We start with discussing the notion of semantic shifts, and then continue with an overview of the existing methods for tracing such time-related shifts with word embedding models. We propose several axes along which these methods can be compared, and outline the main challenges before this emerging subfield of NLP, as well as prospects and possible applications.

LED

The study of semantic shifts in human language reflects complex processes in language and society. The availability of large corpora and the development of computational semantics have given rise to a number of research initiatives trying to capture such shifts in a data-driven way. Among them, word embeddings have become a widely used input representation for this task. There are dozens of papers on the topic, mostly published after 2011, but this emerging field is highly heterogenous. This survey paper attempts to describe this diversity, introduce some axes of comparison and outline main challenges which the practitioners face. We hope that this survey will be useful to those who want to understand how this field has developed, and gain an overview of what defines the current state-of-the-art and what challenges lie ahead.

PEGASUS-X

The meanings of words continuously change over time, reflecting complicated processes in language and society. Examples include both changes to the core meaning of words (like the word gay shifting from meaning ’carefree’ to ’homosexual’ during the 20th century) and subtle shifts of cultural associations (like Iraq or Syria being associated with the concept of ’war’ after armed conflicts had started in these countries). Studying these types of changes in meaning enables researchers to learn more about human language and to extract temporal-dependent data from texts. The availability of large corpora and the development of computational semantics have given rise to a number of research initiatives trying to capture diachronic semantic shifts in a data-driven way. This emerging field is highly heterogenous. There are at least three different research communities interested in it: natural language processing (and computational linguistics), information retrieval (and computer science in general), and political science. The present survey paper attempts to describe this diversity, introduce some axes of comparison and outline main challenges which the practitioners face. We hope that this survey will be useful to those who want to understand how this field has developed, and gain an overview of what defines the current state-of-the-art and what challenges lie ahead.

LongT5

Human languages change over time, reflecting complicated processes in language and society. Examples include both changes to the core meaning of words (like the word gay shifting from meaning carefree’ to homosexual’ during the 20th century) and subtle shifts of cultural associations (like Iraq or Syria being associated with the concept of war’ after armed conflicts had started in these countries). Studying these types of changes in meaning enables researchers to learn more about human language and to extract temporal-dependent data from texts. The availability of large corpora and the development of computational semantics have given rise to a number of research initiatives trying to capture diachronic semantic shifts in a data-driven way. Word embeddings have become a widely used input representation for this task. There are dozens of papers on the topic, mostly published after 2011 (we survey them in Section 3 and further below). However, this emerging field is highly heterogenous, with at least three different research communities interested in it: natural language processing (and computational linguistics), information retrieval (and computer science in general), and political science. This is reflected in the terminology, which is far from being standardized. The present survey paper attempts to describe this diversity, introduce some axes of comparison and outline main challenges which the practitioners face.

Figure 6: Reference summary and candidate summaries generated by fine-tuned models.

ChatGPT

This survey paper provides an overview of research on the detection of semantic shifts using diachronic word embeddings. Semantic shifts refer to changes in the meaning of words over time, reflecting processes in language and society. The availability of large corpora and the development of computational semantics have led to research initiatives that aim to capture diachronic semantic shifts in a data-driven way. Word embeddings have become a widely used input representation for this task. The paper describes the diversity of the field, introduces axes of comparison, and outlines main challenges faced by researchers. The paper covers different approaches to the task of automatic detection of semantic shifts, including the use of various sources of diachronic data for training and testing, and the methodology of extracting semantic shifts from data. It also discusses the development of "laws" of semantic change based on empirical observations, such as the law of prototypicality, the law of conformity, the law of innovation, and the law of differentiation. The paper concludes by discussing applications of diachronic word embeddings in linguistic studies and event detection, and by highlighting open challenges in the field, such as expanding research to other languages and developing mathematical models for diachronic embeddings. Overall, the paper provides a comprehensive overview of the current state of research in this area and identifies important directions for future work.

ChatGLM3

Here is a summary of the key points from the text: - The text surveys research on using diachronic (temporal) word embeddings to detect semantic shifts in language over time. - Semantic shifts are defined as changes to the meaning of words, either in their core meaning or cultural associations. They can be studied by analyzing large corpora and comparing word vectors over time. - The text covers the sources of diachronic data used, such as Google Ngrams, Amazon reviews, and social media. Time granularity can range from decades to monthly slices. - Distributional models like word2vec and prediction-based models like CBOW and skip-gram are commonly used. Methods for comparing word vectors across time include aligning models, using ’second-order embeddings’, and incremental updating. - Some proposed ’laws’ of semantic change, like the law of prototypicality, have been questioned due to issues with model assumptions. - Applications of diachronic word embeddings include linguistic studies of semantic shifts and event detection for cultural shifts. - Key open challenges include expanding to more languages, developing algorithms for small datasets, creating gold standard test sets, developing formal models, and analyzing the nature and causes of semantic shifts. - The text provides an overview of the current state-of-the-art and challenges for this emerging field.

Vicuna

This survey paper discusses the field of tracing semantic shifts using distributional word embedding models, which involves studying changes in the meaning of words over time using large corpora and computational semantics. The paper provides an overview of the different approaches and methods used in this field, including word embeddings, temporal embeddings, and dynamic embeddings. The paper also discusses the challenges and open problems in this field, such as the need for more data and better evaluation methods. Overall, the paper provides a comprehensive introduction to the field of tracing semantic shifts using distributional word embedding models.

Figure 7: Candidate summaries generated by large language models.