# TLDR: Extreme Summarization of Scientific Documents

**Isabel Cachola**[†] **Kyle Lo**[†] **Arman Cohan**[†] **Daniel S. Weld**[†‡]

[†]Allen Institute for AI

[‡]Paul G. Allen School of Computer Science & Engineering, University of Washington

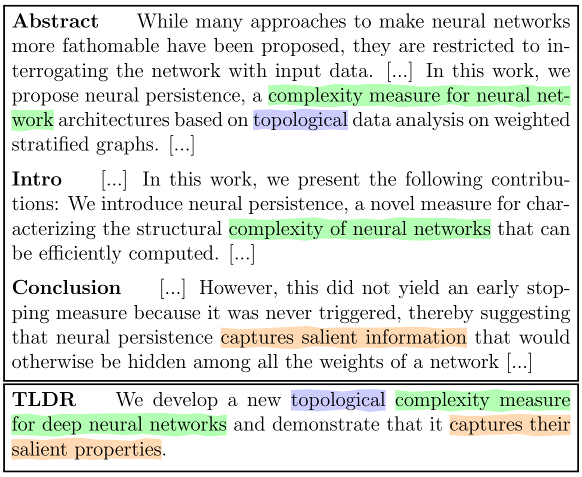
{isabelc,kylel,armanc,danw}@allenai.org

## Abstract

We introduce TLDR generation, a new form of extreme summarization, for scientific papers. TLDR generation involves high source compression and requires expert background knowledge and understanding of complex domain-specific language. To facilitate study on this task, we introduce SciTLDR, a new multi-target dataset of 5.4K TLDRs over 3.2K papers. SciTLDR contains both author-written and expert-derived TLDRs, where the latter are collected using a novel annotation protocol that produces high-quality summaries while minimizing annotation burden. We propose CATTS, a simple yet effective learning strategy for generating TLDRs that exploits titles as an auxiliary training signal. CATTS improves upon strong baselines under both automated metrics and human evaluations. Data and code are publicly available at https://github.com/allenai/scitldr.

## 1 Introduction

We introduce TLDR[1] generation for scientific papers. An alternative to abstracts, TLDRs focus on the key aspects of the paper, such as its main contributions, eschewing nonessential background or methodological details. Given the increasing pace of publication (Van Noorden, 2014) and resulting difficulty in keeping up with the literature, TLDRs can enable readers to quickly discern a paper's key points and decide whether it's worth reading. The goal of existing work in summarization of scientific documents is to generate abstracts or provide complimentary summaries to abstracts. (Collins et al., 2017; Cohan et al., 2018; Chandrasekaran et al., 2019; Yasunaga et al., 2019). In contrast, TLDR generation seeks to produce an extreme (single sentence) summary (Narayan et al., 2018) given the entire paper. Further, TLDR generation is a challenging natural language generation task. Writing a TLDR of a scientific paper requires expert background knowledge and understanding of complex domain-specific language to identify the salient aspects of the paper, while maintaining faithfulness to the source and correctness of the written summary. An example TLDR is provided in Figure 1.



**Abstract** While many approaches to make neural networks more fathomable have been proposed, they are restricted to interrogating the network with input data. [...] In this work, we propose neural persistence, a complexity measure for neural network architectures based on topological data analysis on weighted stratified graphs. [...]

**Intro** [...] In this work, we present the following contributions: We introduce neural persistence, a novel measure for characterizing the structural complexity of neural networks that can be efficiently computed. [...]

**Conclusion** [...] However, this did not yield an early stopping measure because it was never triggered, thereby suggesting that neural persistence captures salient information that would otherwise be hidden among all the weights of a network [...]

**TLDR** We develop a new topological complexity measure for deep neural networks and demonstrate that it captures their salient properties.

Figure 1: An example TLDR of a scientific paper. A TLDR is typically composed of salient information (indicated by colored spans) found in the abstract, intro, and conclusion sections of a paper.

To facilitate the study of TLDR generation, we introduce SciTLDR, a new dataset of 5,411 TLDRs of computer science papers. SciTLDR is built from a combination of TLDRs written by authors of submissions on OpenReview[2] and TLDRs derived by a novel annotation protocol that asks domain experts to rewrite peer review comments for that submission. Having multiple gold summaries per paper is especially important for evaluation when there is

[1] TLDR is an acronym that stands for "too long; didn't read," which is often used in online informal discussion (e.g., Twitter or Reddit) about scientific papers. For visual clarity, we omit the semi-colon.

[2] https://openreview.net/

variability in human-written gold summaries (Zechner, 1996; Harman and Over, 2004).

In addition to establishing strong extractive and abstractive summarization baselines using Transformer-based (Vaswani et al., 2017) models, we present CATTS (Controlled Abstraction for TLDRs with Title Scaffolding), a simple yet effective learning strategy for TLDR generation. CATTS incorporates ideas from scaffold tasks for multitask learning (Swayamdipta et al., 2018a; Cohan et al., 2019) and control codes in conditional language generation (Keskar et al., 2019) to address the problem of data scarcity in the highly-specialized scientific domain. In particular, CATTS exploits titles as an auxiliary, naturally-occurring training signal by training the model to generate both titles and TLDRs indicated by control codes. We show that CATTS applied to BART (Lewis et al., 2020), a state-of-the-art summarization model, results in performance improvement in both automated metrics and human evaluation.

Our contributions are summarized below:

1. We introduce TLDR generation, a new form of extreme summarization, for scientific papers. With extensive analysis of properties of TLDRs, we provide insight into the types of information and amount of variability in human-written TLDRs.
2. We release SciTLDR, a new multi-target dataset of 5,411 TLDRs over 3,229 scientific papers. SciTLDR contains both author-written and expert-derived TLDRs, where the latter are collected using a novel annotation protocol that produces high-quality summaries while avoiding the burden of reading the full paper.
3. We establish strong baselines on SciTLDR and improve them with CATTS, a simple yet effective learning strategy for generating TLDRs that uses titles as an auxiliary training signal.
4. We perform extensive analysis and human evaluation of system-generated TLDRs, focusing on informativeness and factual correctness.

**Peer review** The paper proposes variance regularizing adversarial learning (VRAL), a new method for training GANs. The motivation is to ensure that the gradient for the generator does not vanish. [...] The discriminator itself is trained through two additional meta-discriminators Are the meta-discriminators really necessary? Have you tried matching moments or using other methods [...]

**Derived TLDR** The paper proposes variance regularizing adversarial learning for training gans to ensure that the gradient for the generator does not vanish.

Figure 2: Example of a reviewer comment rewritten as a TLDR (best viewed in color). A peer review comment often begins with a summary of the paper which annotators use to compose a TLDR. Annotators are trained to preserve the original reviewer's wording when possible (indicated by colored spans), and to avoid using any *excess details* or *criticism*.

## 2 Dataset construction

**Overview** We introduce SciTLDR, a new multi-target dataset of 5,411 TLDRs over 3,229 scientific papers in the computer science domain.[3] The training set contains 1,992 papers, each with a single gold TLDR. The dev and test sets contain 619 and 618 papers each, with 1,452 and 1,967 TLDRs, respectively. This is unlike the majority of existing summarization datasets that assume only one gold summary for a given document.

As evidenced by earlier work in summarization evaluation (Cohan and Goharian, 2016), variability in human-written summaries (Zechner, 1996; Harman and Over, 2004) can negatively impact the reliability of automated summarization metrics like Rouge (Lin, 2004).[4] Considering only one gold TLDR for each paper as a basis of automated evaluation might result in inaccurate system quality assessment because content that might appear in a TLDR can have large variability. In addition, having multiple gold summaries for each document enables performing more in-depth analysis and thorough evaluation (Nenkova and Passonneau, 2004).

To address this, SciTLDR contains TLDRs written from the perspective of the author ("TLDR-Auth") and TLDRs written from the perspective of the peer reviewer("TLDR-PR"). We describe these two types of TLDRs in the following paragraphs.

**Collecting TLDR-Auth pairs** Scholar-written TLDRs of scientific papers are available on various online platforms. On OpenReview.org, a publicly available scientific reviewing platform, authors submit TLDRs of their papers that summarize the main content for both reviewers and other interested scholars. Scholars also share TLDRs social media platforms, such as Twitter and Reddit.

We use the OpenReview API[5] to collect pairs of papers and author-written TLDRs, along with the

[3] See Appendix Table 9 for full venue breakdown.

[4] While Rouge is capable of handling multiple targets for a given document, most summarization datasets are single target. See Table 1.

[5] https://github.com/openreview/openreview-py

| Dataset | Number of documents | Avg. words in document | Avg. words in summary | Compression ratio | % novel words | Multi-target |
|---|---|---|---|---|---|---|
| *Non-scientific documents* | | | | | | |
| DUC (Over, 2003) | 624 | 441 | 11 | 40.1 | 30.0 | yes |
| NYTimes (Sandhaus, 2008) | 655K | 549 | 40 | 13.7 | 20.1 | no |
| DailyMail (Hermann et al., 2015) | 220K | 653 | 55 | 11.9 | 17.0 | no |
| CNN (Hermann et al., 2015) | 93K | 760 | 46 | 16.5 | 16.8 | no |
| XSUM (Narayan et al., 2018) | 226K | 431 | 23 | 18.7 | 35.8 | no |
| Newsroom (Grusky et al.) | 1.32M | 659 | 27 | 24.4 | 26.0 | no |
| BigPatent (Sharma et al., 2019) | 1.34M | 3.6K | 117 | 30.5 | 13.6 | no |
| *Scientific documents* | | | | | | |
| CLPubSum (Collins et al., 2017) | 10.3K | 8.2K | 226 | 36.5 | 7.7 | no |
| PubMed (Cohan et al., 2018) | 133K | 3K | 203 | 14.9 | 10.5 | no |
| ArXiv (Cohan et al., 2018) | 215K | 4.9K | 220 | 22.5 | 8.3 | no |
| SciSummNet[†] (Yasunaga et al., 2019) | 1.0K | 4.7K | 150 | 31.2 | 7.4 | no |
| TalkSumm[‡] (Lev et al., 2019) | 1.7K | 4.8K | 965 | 5.0 | 16.5 | no |
| **SciTLDR** (ours) | 3.2K | 5K | 21 | 238.1 | 15.2 | yes |

Table 1: Comparison of SciTLDR to existing summarization datasets. *(i)* SciTLDR provides multiple summary targets unlike other recent summarization datasets. *(ii)* SciTLDR requires both extreme compression and abstraction, as evidenced by the compression ratio and novelty (% of summary words not in the source document), especially when compared with other scientific summarization datasets.

[†] SciScummNet data was later included in the CL-SciSumm shared task and dataset (Jaidka et al., 2018; Chandrasekaran et al., 2019), which has an additional 40 manually annotated documents and its statistics are similar to SciSummNet.

[‡] Unlike the other summarization datasets presented here, TalkSumm is an automatically-constructed dataset for training; the TalkSumm-supervised model in Lev et al. (2019) was evaluated using CL-SciSumm (Jaidka et al., 2018).

full-text PDFs[6] of those papers. We use the S2ORC pipeline (Lo et al., 2020) to convert PDFs to structured, machine-readable full text. We then split the papers randomly into the previously-mentioned train, dev, and test sets; each paper at this point has an associated author-written gold TLDR.

**Rewriting peer reviews into TLDR-PR pairs** Scaling up data collection in a specialized scientific domain is costly and challenging. To sidestep this problem, we use a novel annotation protocol that exploits natural summaries in peer review comments. Assuming the typical peer reviewer has carefully scrutinized the source paper and provided a faithful summary in their comment (often in the first paragraph), domain experts can rewrite these comments into TLDRs.

For this task, we recruit 28 undergraduate computer science students from the University of Washington with self-reported experience in reading scientific papers. Each recruited student received one hour of one-on-one writing training and then was asked to work independently. Annotators were only shown the first 128 words of a sampled[7] peer review comment. They were instructed to keep their TLDRs between 15-25 words (similar to the length of an author written TLDR) and to skip reviews that do not contain a summary or if they did not understand the content. They were also instructed to use the original language in the review, when possible. We manually assessed every written summary, discarding TLDRs that did not adhere to the guidelines, and allowed 20/28 students who performed well to continue work beyond the first hour. Students were compensated at the local median hourly wage of $20 USD per hour. Refer to Appendix §F for full annotation instructions. Figure 2 contains an example of a peer review and its corresponding TLDR-PR. We discuss differences between TLDR-PR and TLDR-Auth throughout Section 3.

## 3 Dataset analysis

### 3.1 Compression and abstractiveness

Table 1 compares SciTLDR with other summarization datasets in both scientific and non-scientific domains. We observe that SciTLDR has short summaries, like XSUM and NewsRoom, with long

[6] A small fraction of those papers ($< 5\%$) did not have an available PDF file, so we could not parse their full body text. This are still included the dataset as it is possible to generate a TLDR from an abstract alone.

[7] Multiple peer review comments can be available for each paper on OpenReview. We focused on ensuring that each paper in dev and test had at least one TLDR-PR.

source documents, like BigPatent and the other scientific-domain datasets. This results in a much higher compression ratio compared with existing datasets. Summarization in higher compression settings is challenging as it requires capturing more precisely the salient aspects of the document (Grusky et al.).

Following Narayan et al. (2018); Grusky et al., we measure abstractiveness (or novelty) by percentage of words in the summary that do not appear in the source document. We observe that SCITLDR is more abstractive compared with other scientific domain datasets but less abstractive compared with non-scientific domain datasets. We also observe that SCITLDR is smaller in comparison to automatically collected datasets, such as XSUM and ArXiv, but is larger in comparison to other manually collected datasets, such as SciSummNet.

### 3.2 Information content

We analyze the information content of TLDRS using an approach motivated by the nugget-based summarization evaluation framework of Nenkova and Passonneau (2004). In a similar manner, we asked two computer science researchers to read through a collection of TLDRS to both define a comprehensive set of categories of types of information present in TLDRS, which we refer to as nuggets.[8] We also label each TLDR with all represented nuggets. Table 2 presents this categorization, along with example phrases and nugget occurrence frequencies of SCITLDR. For simplicity, we use the category codes defined in the table (with brackets) to reference specific categories.

Most TLDRS contain between two to four nuggets (never all six), and will provide some indication of their subject area (**A**) and the paper's contributions (**C**). In fact, they are the most frequently *co-occurring* nuggets, appearing in 63% of TLDR-Auth and 71% of TLDR-PR. TLDR-Auth tend to include results or scientific/theoretical findings (**R**) and often signal the value of their work (**V**) by describing their contributions as *novel* or their results as *strong* or *state-of-the-art*. In contrast, TLDR-PR focus more on articulating problems the paper addresses (**P**). Interestingly, TLDR-PR place less emphasis on **R** and **V** in favor of further methodological details in the paper **D**. More details about nuggets in Appendix §A.

| Category | Example phrase | % of TLDRS<br>AUTH / PR |
|---|---|---|
| **[A]**rea, field or topic of study | *reinforcement learning, dependency parsing* | 85.6 / 90.8 |
| **[P]**roblem or motivation | *mode collapse, catastrophic forgetting* | 29.0 / 32.9 |
| Mode of **[C]**ontribution | *method, dataset, proof, theorem* | 68.4 / 76.3 |
| **[D]**etails or description | *graph convolution operations with dynamically computed graphs* | 43.4 / 57.9 |
| **[R]**esults or findings | *improved performance on ImageNet,*<br>*simple defenses work on MNIST but not CIFAR* | 29.0 / 17.1 |
| **[V]**alue or significance | *novel, state-of-the-art,*<br>*simple yet effective,*<br>*easily applicable* | 23.7 / 7.9 |

Table 2: Example categories (or nuggets) of information a TLDR might contain. Proportion of TLDRS containing each nugget estimated on 76 randomly sampled gold papers (each with its TLDR-Auth and a sampled TLDR-PR). Percentages do not sum to one because each TLDR can contain multiple nuggets.

### 3.3 Variability in TLDRS

To explore variability in our human-written summaries, we examine differences between TLDRS written by authors (TLDR-Auth) and TLDRS derived from the perspective of a peer reviewer (TLDR-PR).

**Lexical variation** First, we note that TLDR-Auth are on average 18.9 words long, while TLDR-PR are slightly longer on average at 22.9 words. Despite similarities in length, the 1-, 2-, and 3-gram mean Jaccard indices between TLDR-Auth and TLDR-PR are 15.0%, 2.5%, and 0.7%, respectively, indicating extremely little lexical overlap between the two sources of TLDRS. We can also observe through qualitative examples in Figure 3 how TLDR-Auth and TLDR-PR can differ greatly, even when they contain the same information content.

**Abstractiveness** TLDR-PR is more abstractive with a novelty score of 20.2% compared with TLDR-Auth with a novelty score of 9.6%, where novelty is computed as the percentage of words in the TLDR *not* in the source paper. This is not unexpected because TLDR-PR are derived from peer review comments which themselves have already gone through one stage of abstraction.

[8] While we adopt the term 'nugget' for convenience, we recognize that that they traditionally correspond to factoids, while here they correspond to discourse roles Teufel (1999).

**TLDR-Auth** The authors propose a framework to learn a good policy through imitation learning from a noisy demonstration set via meta-training a demonstration suitability assessor.
**TLDR-PR** Contributes a maml based algorithm for imitation learning which automatically determines if provided demonstrations are "suitable".

**TLDR-Auth** The authors evaluate the effectiveness of having auxiliary discriminative tasks performed on top of statistics of the posterior distribution learned by variational autoencoders to enforce speaker dependency.
**TLDR-PR** Propose an autoencoder model to learn a representation for speaker verification using short-duration analysis windows.

Figure 3: Two example TLDR-Auth and TLDR-PR pairs with colored spans corresponding to nuggets in Table 3 – **A**, **P**, **C**, **D**. On **top**, we see TLDRs can have substantial lexical variation despite covering similar information content. On **bottom**, we naturally see even more variation when the information content differs.

## 4 CATTS

We introduce CATTS (Controlled Abstraction for TLDRs with Title Scaffolding), a simple yet effective method for learning to generate TLDRs. Our approach addresses two main challenges: (1) the limited size of the training data and (2) the need for domain knowledge in order to write high-quality gold TLDRs. To address these challenges, we propose using *titles* of scientific papers as additional generation targets. As titles often contain key information about a paper, we hypothesize that training a model to generate titles will allow it to learn how to locate salient information in the paper that will be also useful for generating TLDRs. In addition, all papers have a title, and thus we have an abundant supply of paper-title pairs for training.

Incorporating auxiliary **scaffold** tasks via multitask learning has been studied before for improving span-labeling and text classification (Swayamdipta et al., 2018b; Cohan et al., 2019). Similar to multitask learning, training on heterogenous data annotated with **control codes** has been shown to improve controlled generation in autoregressive language models (Keskar et al., 2019; ElSahar et al., 2020; Sudhakar et al., 2019; Li et al., 2020). In fact, it has been shown effective for generating biomedical abstracts (Sybrandt and Safro, 2020). We demonstrate that control codes can be used to effectively incorporate scaffold tasks (e.g. title generation) for denoising autoencoders like BART (Lewis et al., 2020).

In order to use title generation as a scaffold task for TLDR generation, we propose shuffling SciTLDR with a title generation dataset, then appending each source with control codes ⟨|TLDR|⟩ and ⟨|TITLE|⟩, respectively. This allows the parameters of the model to learn to generate both TLDRs and titles. This process is visualized in Frigure 4. At generation time, the appropriate control code is appended to the source. Additionally, upsampling particular tasks can be viewed as applying task-specific weights, similar to weighting losses in multitask learning setups.

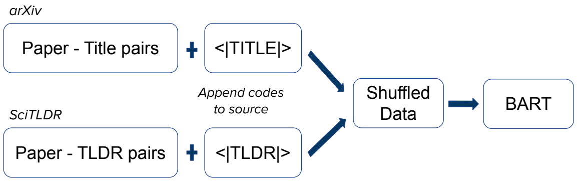


Figure 4: Training regimen for CATTS.

## 5 Experiments

### 5.1 Baselines

We establish baselines for TLDR generation on SciTLDR using state-of-the-art extractive and abstractive summarization models.

**Extractive methods** We consider both unsupervised and supervised extractive methods. For our unsupervised baseline, we use PACSUM (Zheng and Lapata, 2019), an extension of TextRank (Mihalcea and Tarau, 2004) that uses BERT (Devlin et al., 2019) as a sentence encoder. For our supervised baselines, we use BertSumExt (Liu and Lapata, 2019), which uses BERT as a sentence encoder augmented with inter-sentence Transformer layers to capture interactions, and MatchSum (Zhong et al., 2020), which uses a BERT Siamese network to score whole summaries.

**Abstractive methods** Since TLDRs often contain information spread across multiple sentences, we expect abstractive summarization methods to produce strong results for this task. We focus on BART (Lewis et al., 2020), a Transformer-based denoising autoencoder for pretraining sequence-to-sequence models. We use BART-large, which achieves state-of-the-art results in summarization on XSUM. We additionally use BART-large finetuned on XSUM, hypothesizing that the task of extreme summarization of news articles might transfer to TLDR generation on SciTLDR.

**Oracle** We define a sentence-level extractive oracle: Given a paper and its multiple gold TLDRs, it selects the single sentence in the document with the highest Rouge overlap for each gold TLDR. Then it returns the single sentence that yields the maximum Rouge across all gold TLDRs. This sets an upper-bound on the performance of the sentence-level extractive methods under our multi-target evaluation (Section 5.4). Our full text oracle achieves 54.5 Rouge-1, 30.6 Rouge-2, and 45.0 Rouge-L on the test set.

## 5.2 Input space

The **input space** is the context provided to the model when generating TLDRs.

**Abstract-only** Since the vast majority of scientific papers do not have open-access full text (Lo et al., 2020), it is worth considering the setting in which we generate TLDRs for papers given only their abstracts as input. The average length of an abstract is 159 words and resulting compression ratio is 7.6.

**AIC** Previous studies have found that the most salient information in a paper for writing a summary is often found in the abstract, introduction, and conclusion (AIC) sections (Sharma et al., 2019). An important consequence of this is the ability to substantially reduce computational costs[9] (Schwartz et al., 2019) by supplying only these sections as context. The average combined length of these contexts is 993 words and resulting compression ratio is 47.3, which is still higher than other datasets surveyed in Table 1.

Comparing oracle results in Table 3, we see that increasing the input space from abstract-only to AIC improves Rouge-1 by +4.7. Yet, this is only 2.1 Rouge-1 lower than the full text oracle performance, despite requiring five times more text.

## 5.3 Training and implementation details

All experiments use Titan V or V100 GPUs. We experiment on abstract-only and AIC input spaces. Best hyperparameters for the models are selected based on dev set Rouge-1. Supervised models like BERTSUMEXT and BART are trained on SCITLDR and the best model checkpoint chosen using dev set loss. See Appendix§D for additional parameter tuning details of all models.

**Extractive Methods** For PACSUM, BERTSUMEXT and MatchSum we use original code released by the authors. The first two use BERT-base and the last one uses RoBERTa-base (Liu et al., 2019). For MatchSum in AIC input space, following the authors, we use BERTSUMEXT to first extract 7 highly scoring sentences as the input to MatchSum.[10] Sentence segmentation is performed using ScispaCy (Neumann et al., 2019), and models select a single sentence as their predictions. We use the default hyperparameters for PACSUM.

**Abstractive Methods** We experiment with BART-large and BART-large finetuned on XSUM, using the Fairseq (Ott et al., 2019) implementation and the released XSUM weights. We apply the CATTS training method to these two models, using an additional 20K paper-title pairs from arXiv for title generation.[11] We up-sample TLDR instances to match the size of the title scaffold data.[12] For simplicity, we refer to these as BART, $\text{BART}_{\text{XSUM}}$, CATTS and $\text{CATTS}_{\text{XSUM}}$, respectively. For all models, we use a learning rate of 3e-5, update frequency of 1, and max tokens per batch of 1024[13] chosen through manual tuning. We tune decoder for all models via grid search over five length penalties between 0.2 and 1.0 and 7 beam sizes 2 to 8.

## 5.4 Evaluation

**Automated evaluation** Following recent work on extreme summarization (Narayan et al., 2018; Lewis et al., 2020), we use Rouge-1, Rouge-2, and Rouge-L (Lin, 2004) as our automated metrics. As discussed in Section 2, we have multiple target summaries available per paper. To exploit this during evaluation, we calculate the Rouge score of the system-generated TLDR with respect to each of the gold TLDRs for the corresponding paper (including its TLDR-Auth and all of its TLDRs-PR) individually. We take the **maximum** Rouge score over these gold TLDRs as the final Rouge score for that paper. An alternative approach to aggregating scores would be to take the mean, but due to the

[9] Especially for methods that rely on $O(n^2)$ inter-sentence comparisons or wrappers around Transformer-based methods to long contexts.

[10] In abstract-only setting, MatchSum takes the full context.

[11] Includes all papers on arXiv with at least one of the following tags CS.CL, CS.CV, CS.LG, CS.AI, CS.NE, and STAT.ML *and* have identified introduction and conclusion sections by S2ORC (Lo et al., 2020).

[12] While this up-sampling may indicate that CATTS is training on more TLDRs than BART, we allow BART training up to 20 epochs and it quickly overfits within a few epochs.

[13] Fairseq reports an "average batch size" of 36, which is a consequence of adaptive batching of examples based on the update frequency and max tokens per batch.

| Method | Abstract-only |  |  | AIC |  |  |
|---|---|---|---|---|---|---|
|  | R1 | R2 | RL | R1 | R2 | RL |
| *Oracle* | 47.7 | 24.7 | 38.5 | 52.4 | 29.0 | 42.9 |
| PACSUM (Zheng and Lapata, 2019) | 19.3 | 4.0 | 15.1 | 28.7 | 9.8 | 21.9 |
| BERTSUMEXT (Liu and Lapata, 2019) | 38.5 | 16.6 | 30.5 | 36.2 | 14.7 | 28.5 |
| MatchSum (Zhong et al., 2020) | 42.7 | 20.0 | 34.0 | 38.6 | 16.4 | 30.1 |
| BART (Lewis et al., 2020) | 43.3 | 20.8 | 35.0 | 42.9 | 20.8 | 35.1 |
| BART$_{\text{XSUM}}$ (Lewis et al., 2020) | 42.5 | 21.1 | 34.9 | 43.7 | 21.4 | 36.0 |
| CATTS (Ours) | **43.8** | 20.9 | 35.5 | $^\dagger$**44.9** | $^\dagger$**22.6** | $^\dagger$**37.3** |
| CATTS$_{\text{XSUM}}$ (Ours) | $^\dagger$44.3 | **21.3** | **35.9** | 44.6 | 21.7 | 36.5 |

Table 3: Test set max Rouge scores of extractive and abstractive baselines and CATTS. We use $^\dagger$ to indicate CATTS variants that significantly ($p<0.05$) outperform their corresponding BART baseline.

variability in TLDRs shown in Section 3.3, we argue the maximum operation is more appropriate – That is, matching *any* of the gold TLDRs is rewarded.[14]

**Human evaluation** While our multi-target setting allows us to mitigate some of the limitations of Rouge (Conroy et al., 2011; Cohan and Goharian, 2016), we acknowledge that relying only on automated metrics is insufficient for evaluating the quality of the models. In addition to automated metrics, we also have human experts in computer science assess system-generated TLDRs under two criteria – informativeness and correctness.

For **informativeness**, we perform the nugget-based analysis for information content over system-generated TLDRs for the same 76 gold papers from Section 3.2. We use the presence (or lack) of different nuggets in predicted and gold TLDRs to quantify differences in information content. Specifically, we score each gold and system-generated TLDR by *the number of unique nuggets divided by the number of tokens*. This length normalization handles cases where systems returning the source document are trivially more informative. For each paper, we rank the predicted and gold TLDRs. Then, we compute overall metrics for each gold or system variant by aggregating their ranks across papers using mean reciprocal rank (MRR).

Evaluating **correctness** requires careful reading and understanding the source paper. To minimize this burden and have reliable evaluation, we ask the original authors of papers to assess the correctness of our system-generated TLDRs. We manually email (first or second) authors of arXiv papers and ask them to score each system-generated TLDR with *1 - false or misleading*, *2 - partially accurate* or *3 - mostly correct*, regardless of comprehensiveness. We compare the mean correctness (across papers) for each system variant. We received responses from 29 unique authors with annotations covering 64 arXiv papers.

|  | MRR | Avg. # nuggets | Avg. # words |
|---|---|---|---|
| TLDR-Auth (Gold) | 0.53 | 2.5 | 20.5 |
| TLDR-PR (Gold) | 0.60 | 2.4 | 18.7 |
| BART$_{\text{XSUM}}$ | 0.42 | 2.2 | 19.4 |
| CATTS$_{\text{XSUM}}$ | 0.54 | 2.6 | 20.8 |

Table 4: Human evaluation on informativeness of gold and system-generated TLDRs. Higher MRR corresponds to variants that, on average, rank higher than others by length-normalized number of nuggets.

## 6 Results

### 6.1 Quantitative results

We present our main results in Table 3.

**Extractive results** We establish baseline results for extractive methods on our new dataset SCITLDR. We observe that MatchSum has the highest extractive performance, followed by BERTSUMEXT. We observe that increasing input space from abstract-only to AIC greatly improves PACSUM[15] performance but decreases performance of both BERTSUMEXT and MatchSum. We suspect that increasing the input space makes it more difficult for these models to learn optimal parameters including new position embeddings in low-resource training. Compared to the extractive oracle scores, we see there is plenty of room for improvement.

[14] For completeness we provide mean Rouge scores in Appendix Table 10 to supplement our main max Rouge results in Table 3.

[15] PACSUM using the full text yields a Rouge-1 of 12.7, significantly worse than abstract-only.

| Method | Abstract-only | | AIC | |
|---|---|---|---|---|
| | % novel words | Avg. # words | % novel words | Avg. # words |
| BART | 2.9% | 20.9 | 1.3% | 20.4 |
| BART$_{\text{XSUM}}$ | 3.7% | 18.4 | 1.1% | 18.9 |
| CATTS | 5.5% | 19.1 | 5.3% | 18.4 |
| CATTS$_{\text{XSUM}}$ | 5.8% | 19.7 | 4.5% | 19.7 |

Table 5: Lexical features of system-generated TLDRs.

| Method | R1 | Δ | R2 | Δ | RL | Δ |
|---|---|---|---|---|---|---|
| BART | 44.9 | +1.6 | 22.6 | +1.8 | 37.1 | +2.1 |
| BART$_{\text{XSUM}}$ | 44.8 | +1.1 | 21.8 | +0.4 | 36.4 | +0.4 |
| CATTS | 44.9 | +0.0 | 21.9 | -0.7 | 36.6 | -0.7 |
| CATTS$_{\text{XSUM}}$ | 45.7 | +1.1 | 23.0 | +1.7 | 37.1 | +1.2 |

Table 6: Oracle input space experiments. Δ are differences between oracle result and model's best performance (across abstract-only and AIC) from Table 3.

**Abstractive results** Abstractive methods are not limited to choosing exact sentences. For a given abstractive baseline BART or BART$_{\text{XSUM}}$, our CATTS learning strategy results in improvements in both abstract-only and AIC settings. Comparing CATTS variants with their corresponding BART baselines, we observe that in the abstract-only setting, CATTS and CATTS$_{\text{XSUM}}$ achieve +0.5 and +1.8 Rouge-1, respectively. In the AIC setting, CATTS and CATTS$_{\text{XSUM}}$ achieve +2.0 and +0.9 Rouge-1, respectively. We use the two-sided paired t-test against a null hypothesis of no difference to assess these differences. To address the issue of multiple hypothesis testing over Rouge scores, we perform a Holm-Bonferroni (Holm, 1979)[16] correction for determining significant $p$-values in Table 3.

### 6.2 Human evaluation

We perform our human evaluation on BART$_{\text{XSUM}}$ and CATTS$_{\text{XSUM}}$ using the AIC input space on 51 sampled papers. In this setting, we have both chosen the strongest baseline and controlled for XSUM pretraining. From Table 4, we see that CATTS$_{\text{XSUM}}$ is more informative than BART$_{\text{XSUM}}$ and is comparable to gold TLDR-Auth, though still less informative than TLDR-PR.

In addition to informativeness, we also evaluate content accuracy of generated tldrs as explained in Section 5.4. We report no difference in correctness between BART$_{\text{XSUM}}$ and CATTS$_{\text{XSUM}}$. We observe 42 ties, 10 cases where BART$_{\text{XSUM}}$ is more correct, and 12 cases where CATTS$_{\text{XSUM}}$ is more correct. Both models average a rating of 2.5 (scoring between partially accurate and mostly correct).

### 6.3 Analysis

**How abstractive are the generations?** From Table 5, we observe: (1) BART variants are less abstractive than CATTS variants. (2) Initial training on XSUM might influence models to be slightly less abstractive. (3) BART variants are more abstractive in the abstract-only setting than the longer AIC settings, while CATTS seems to have the same level of abstractiveness regardless of input space.

**How long are the generations?** From Table 5, we see the systems all generate TLDRs of similar length to the average length reported in Table 1.

**How important is using the full text?** To analyze whether one can improve abstractive model performance by improving the input space selection (compared to just using AIC), we define an *oracle input space*. That is, for each TLDR, we select sentences from the full text that maximize Rouge-1 with the gold TLDRs-Auth[17] and select the top sentences to match the length of AIC. Repeating the experiments in Section 5 with this input source, we observe some performance improvement across models (Table 6).

**Qualitative example** Table 7 contains system generations on the same paper (alongside the gold TLDRs). Curiously, despite both achieving the same Rouge-1, the generated TLDRs are quite different. BART$_{\text{XSUM}}$ focuses on the methodological contribution while CATTS$_{\text{XSUM}}$ focuses on a scientific finding. The "two hidden layer" detail by BART$_{\text{XSUM}}$ is from the paper introduction and the "defining the appropriate sampling distributions" from CATTS$_{\text{XSUM}}$ is from the conclusion.[18]

## 7 Related work

**Transformers for summarization** Transformer-based models have achieved strong results in extractive and abstractive summarization. PACSUM (Zheng and Lapata, 2019) combines BERT sentence representation with unsupervised text ranking; MatchSum (Zhong et al., 2020) uses a Siamese BERT model to score the entire summary instead of a single extraction; and Liu and Lapata (2019)

[16] Using the P.ADJUST library in R (R Core Team, 2018)

[17] Only TLDRs-Auth is exists for all papers. TLDRs-PR are only in dev and test.

[18] See original paper: https://openreview.net/pdf?id=SkGT6sRcFX

| | |
|---|---|
| **TLDR-Auth** | We propose a method for the construction of arbitrarily deep infinite-width networks, based on which we derive a novel weight initialisation scheme for finite-width networks and demonstrate its competitive performance. |
| **TLDR-PR** | Proposes a weight initialization approach to enable infinitely deep and infinite-width networks with experimental results on small datasets. |
| **BART**$_{\text{XSUM}}$ | We propose a principled approach to weight initialisation that allows the construction of infinite-width networks with more than two hidden layers. |
| **CATTS**$_{\text{XSUM}}$ | We study the initialisation requirements of infinite-width networks and show that the main challenge for constructing them is defining the appropriate sampling distributions for the weights. |

Table 7: Examples of system generations. BART$_{\text{XSUM}}$ and CATTS$_{\text{XSUM}}$ both achieve Rouge-1 of 40.7 on this paper. Colored spans indicate text overlap.

show that BERT is effective for both extractive and abstractive summarization. Zhang et al. (2019); Bi et al. (2020) introduce new pretraining objectives that improve generation. Sequence-to-sequence models (Raffel et al., 2020; Lewis et al., 2020; Bao et al., 2020) have state-of-the-art performance on XSUM (Narayan et al., 2018), a dataset for extreme summarization dataset of news articles. SciTLDR is a new form of extreme summarization focused on scientific papers.

**Scientific document summarization** Most work in summarization of scientific papers have focused on longer summaries (i.e. 150-200 words). Existing datasets include CSPubSum for extractive summarization (Collins et al., 2017), ArXiv and PubMed for abstract generation (Cohan et al., 2018), and SciSummNet (Yasunaga et al., 2019) and CL-SciSumm (Jaidka et al., 2018; Chandrasekaran et al., 2019) datasets, which incorporate citation contexts into human-written summaries. TalkSumm (Lev et al., 2019) uses recordings of conference talks to create a distantly-supervised training set for the CL-SciSumm task.

Modeling approaches in scientific document summarization include models that exploit citation contexts (Qazvinian et al., 2013; Cohan and Goharian, 2015, 2017; Zerva et al., 2020), automated survey generation (Mohammad et al., 2009; Jha et al., 2015; Fabbri et al., 2018; Wang et al., 2018), and other techniques focusing on exploiting the unique properties of scientific documents such as long length and structure (Conroy and Davis, 2017; Nikolov et al., 2018; Cohan et al., 2018; Xiao and Carenini, 2019). Yet, such methods have not been studied in the setting of extreme summarization (i.e. short target summaries, high compression, high abstraction), and SciTLDR is the first dataset to facilitate such research.

## 8 Conclusion

We introduce TLDR generation for scientific papers, and release SciTLDR, a multi-target dataset of TLDR-paper pairs. We also present CATTS, a simple yet effective learning strategy for improving TLDR generation that exploits auxiliary training signal from paper titles. We show that our approach improves over strong modeling baselines.

Existing methods for scientific document summarization often make use of properties unique to those papers, like sections, citation contexts or scientific discourse roles. Future work can examine how best to incorporate these properties to improve TLDR generation models. Additionally, while our experiments are limited to abstract-only and AIC input spaces, we provide the full text of the source papers to support research into using longer input contexts. Furthermore, the multiple target summaries in SciTLDR reflect diverse perspectives and can be used to support summarization research into training and evaluation techniques previously unavailable with existing datasets. Finally, the idea of a TLDR can differ between academic disciplines, and we leave such exploration open for future work.

## Acknowledgments

We thank the Semantic Scholar Research team and John Bohannon and Oleg Vasilyev from Primer for helpful feedback and discussions. This work was supported in part by NSF Convergence Accelerator award 1936940, NSF RAPID award 2040196, ONR grant N00014-18-1-2193, and the University of Washington WRF/Cable Professorship.

## References

Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, Xiulei Liu, Yu Wang, Songhao Piao, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2020. Unilmv2: Pseudo-masked language models for unified language model pre-training. *ArXiv*, abs/2002.12804.

Bin Bi, Chenliang Li, Chen Wu, Ming Yan, and Wei Wang. 2020. Palm: Pre-training an autoencoding and autoregressive language model for context-conditioned generation. *ArXiv*, abs/2004.07159.

Muthu Kumar Chandrasekaran, Michihiro Yasunaga, Dragomir Radev, Dayne Freitag, and Min-Yen Kan. 2019. Overview and results: Cl-scisumm shared task 2019. In *Workshop on Bibliometric-enhanced Information Retrieval and NLP for Digital Libraries (BIRNDL)*.

Arman Cohan, Waleed Ammar, Madeleine van Zuylen, and Field Cady. 2019. Structural scaffolds for citation intent classification in scientific publications. In *NAACL-HLT*.

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 *NAACL-HLT*.

Arman Cohan and Nazli Goharian. 2015. Scientific article summarization using citation-context and article's discourse structure. In *EMNLP*.

Arman Cohan and Nazli Goharian. 2016. Revisiting summarization evaluation for scientific articles. *ArXiv*, abs/1604.00400.

Arman Cohan and Nazli Goharian. 2017. Scientific document summarization via citation contextualization and scientific discourse. *International Journal on Digital Libraries*, 19:287–303.

Ed Collins, Isabelle Augenstein, and Sebastian Riedel. 2017. A supervised approach to extractive summarisation of scientific papers. *CoNLL*, abs/1706.03946.

John M. Conroy and Sashka Davis. 2017. Section mixture models for scientific document summarization. *IJDL*, 19:305–322.

John M Conroy, Judith D Schlesinger, and Dianne P O'Leary. 2011. Nouveau-rouge: A novelty metric for update summarization. *Computational Linguistics*, 37(1):1–8.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv*, abs/1810.04805.

Hady ElSahar, Maximin Coavoux, Matthias Gallé, and Jos Rozen. 2020. Self-supervised and controlled multi-document opinion summarization. *ArXiv*, abs/2004.14754.

Alexander Fabbri, Irene Li, Prawat Trairatvorakul, Yijiao He, Weitai Ting, Robert Tung, Caitlin Westerfield, and Dragomir Radev. 2018. TutorialBank: A manually-collected corpus for prerequisite chains, survey extraction and resource recommendation. In *ACL*.

Max Grusky, Mor Naaman, and Yoav Artzi. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In *NAACL-HLT*.

Donna Harman and Paul Over. 2004. The effects of human variation in DUC summarization evaluation. In *Text Summarization Branches Out*, pages 10–17, Barcelona, Spain. Association for Computational Linguistics.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in neural information processing systems*, pages 1693–1701.

Sture Holm. 1979. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70.

Kokil Jaidka, Muthu Kumar Chandrasekaran, Sajal Rustagi, and Min-Yen Kan. 2018. Insights from cl-scisumm 2016: the faceted scientific document summarization shared task. *IJDL*, 19(2-3):163–171.

Rahul Jha, Reed Coke, and Dragomir R. Radev. 2015. Surveyor: A system for generating coherent survey articles for scientific topics. In *AAAI*.

Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. 2019. CTRL: A Conditional Transformer Language Model for Controllable Generation. *ArXiv*, abs/1909.05858.

Guy Lev, Michal Shmueli-Scheuer, Jonathan Herzig, Achiya Jerbi, and David Konopnicki. 2019. Talksumm: A dataset and scalable annotation method for scientific paper summarization based on conference talks. In *ACL*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *ACL*.

Kun Li, Chengbo Chen, Xiaojun Quan, Qing Ling, and Yan Song. 2020. Conditional augmentation for aspect term extraction via masked sequence-to-sequence generation. *ArXiv*, abs/2004.14769.

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.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. In *EMNLP/IJCNLP*.

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*.

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel S. Weld. 2020. S2orc: The semantic scholar open research corpus. In *Proceedings of ACL*.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into texts. In *EMNLP*.

Saif M. Mohammad, Bonnie J. Dorr, Melissa Egan, Ahmed Hassan Awadallah, Pradeep Muthukrishnan, Vahed Qazvinian, Dragomir R. Radev, and David M. Zajic. 2009. Using citations to generate surveys of scientific paradigms. In *HLT-NAACL*.

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. pages 1797–1807.

Ani Nenkova and Rebecca J Passonneau. 2004. Evaluating content selection in summarization: The pyramid method. In *NAACL*.

Mark Neumann, Daniel King, Iz Beltagy, and Waleed Ammar. 2019. Scispacy: Fast and robust models for biomedical natural language processing.

Nikola I. Nikolov, Michael Pfeiffer, and Richard H. R. Hahnloser. 2018. Data-driven summarization of scientific articles. *ArXiv*, abs/1804.08875.

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 *NAACL-HLT, Demonstrations*.

Paul Over. 2003. An introduction to duc 2003: Intrinsic evaluation of generic news text summarization systems. In *Proceedings of Document Understanding Conference 2003*.

Vahed Qazvinian, Dragomir R. Radev, Saif M. Mohammad, Bonnie J. Dorr, David M. Zajic, Michael Whidby, and Taesun Moon. 2013. Generating extractive summaries of scientific paradigms. *J. Artif. Intell. Res.*, 46:165–201.

R Core Team. 2018. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

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. *JMLR*, 21(140):1–67.

Evan Sandhaus. 2008. The new york times annotated corpus.(october 2008). ldc catalog no.: Ldc2008t19.

Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. 2019. Green ai.

Eva Sharma, Chen Li, and Lu Wang. 2019. Bigpatent: A large-scale dataset for abstractive and coherent summarization. In *ACL*.

Akhilesh Sudhakar, Bhargav Upadhyay, and Arjun Maheswaran. 2019. "transforming" delete, retrieve, generate approach for controlled text style transfer. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3269–3279, Hong Kong, China. Association for Computational Linguistics.

Swabha Swayamdipta, Sam Thomson, Kenton Lee, Luke Zettlemoyer, Chris Dyer, and Noah A. Smith. 2018a. Syntactic scaffolds for semantic structures. In *EMNLP*.

Swabha Swayamdipta, Sam Thomson, Kenton Lee, Luke Zettlemoyer, Chris Dyer, and Noah A. Smith. 2018b. Syntactic scaffolds for semantic structures. In *EMNLP*.

Justin Sybrandt and Ilya Safro. 2020. Cbag: Conditional biomedical abstract generation. *ArXiv*, abs/2002.05637.

S. Teufel. 1999. Argumentative zoning information extraction from scientific text.

Richard Van Noorden. 2014. Global scientific output doubles every nine years. *Nature news blog*.

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 I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *NeurIPS*.

Jie Wang, Chengzhi Zhang, Mengying Zhang, and Sanhong Deng. 2018. Citationas: A tool of automatic survey generation based on citation content. *Journal of Data and Information Science*, 3(2):20–37.

Wen Xiao and Giuseppe Carenini. 2019. Extractive summarization of long documents by combining global and local context. In *EMNLP/IJCNLP*.

Michihiro Yasunaga, Jungo Kasai, Rui Zhang, Alexander Richard Fabbri, Irene Li, Dan Friedman, and Dragomir R. Radev. 2019. Scisummnet: A large annotated corpus and content-impact models for scientific paper summarization with citation networks. In *AAAI*.

Klaus Zechner. 1996. Fast generation of abstracts from general domain text corpora by extracting relevant sentences. In *Proceedings of the 16th conference on Computational linguistics-Volume 2*, pages 986–989.

Chrysoula Zerva, Minh-Quoc Nghiem, Nhung T. H. Nguyen, and S. Ananiadou. 2020. Cited text span identification for scientific summarisation using pretrained encoders. *Scientometrics*, pages 1 – 29.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2019. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. *ArXiv*, abs/1912.08777.

Hao Zheng and Mirella Lapata. 2019. Sentence centrality revisited for unsupervised summarization. In *ACL*.

Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2020. Extractive summarization as text matching. *ACL*.

## A How many nuggets in TLDRs?

| # categories | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| TLDR-Auth | 2.6% | 10.5% | 26.3% | 34.2% |
| TLDR-PR | 0.0% | 9.2% | 30.3% | 31.6% |
| **# categories** | **4** | **5** | **6** | |
| TLDR-Auth | 18.4% | 7.9% | 0.0% | |
| TLDR-PR | 26.3% | 2.6% | 0.0% | |

Table 8: Number of categories represented in a TLDR

## B Breakdown of venues in SciTldr?

| Venue | Proportion |
|---|---|
| ICLR | 85.2% |
| NeurIPS/NIPS | 5.8% |
| OpenReview | 2.1% |
| ICML | 2.0% |
| ICAPS | 1.8% |
| other | 3.1% |

Table 9: Breakdown of venues represented by papers in SciTldr

## C Background knowledge for TLDRs

What a paper's TLDR looks like or what information it should include is subjective and follows (community-specific) commonsense rather than any formally-defined procedure. Since TLDRs are inherently ultra-short, they are not necessarily self-contained statements, and understanding them requires background expertise within their respective scientific domain. Therefore, when designing SciTldr, we assume readers have sufficient background knowledge to follow a general research topic in a given domain. This eliminates the need for TLDRs to include explanations or clarifications of common domain-specific terms (e.g., "bounds," "LSTM," or "teacher").

## D Additional model training details

**PACSUM** The default hyperparameters are beta and lambda1 set to 0. We did some initial tuning of the hyperparameters using the provided tuning code, which performs a search over 10 beta values and 10 lambda1 values. This did not result in a significant difference in performance. PACSUM had a total runtime of 12 minutes on abstracts and 6.5 hours on AIC. We used the released code by authors.[19]

**BertSumExt** We trained with a batch size of 1 sentence per batch and for 5,000 total steps for a total training time of 30 min. We use a learning rate of 2e-3 and a dropout rate of 0.1, which are the reported parameters used for XSUM. BERTSumExt also requires a max token length for initializing position embeddings. For the abstract-only setting, we use the default number of max tokens 512, which fits the full length of all of abstracts in SciTldr. For AIC, we first attempted 3 different truncation lengths – 1024 (double the max tokens for abstracts), 1500 (90th percentile length), and 1800 (95th percentile length) tokens. We found that truncation at 1500 performs best on AIC. We used the released code by authors.[20]

**MatchSum** We trained MatchSum with a batch size of 32, learning rate of 2e-5 with a linear warmup and decay scheduler, and trained the model for 15 epochs. We chose the best checkpoint based on linear combination of Rouge-1, Rouge-2 and Rouge-L. We manually tuned hyperparameters – For learning rate, we tried 2e-5 and 3e-5 and for number of epochs, we tried 5, 15, and 20. For AIC, as MatchSum requires few salient sentences as input for candidate generation, we used BertSumExt to score sentences and chose the top 7 ones as input to MatchSum. This is according to instructions by authors[21]. Instead of training the model from scratch we used the authors released checkpoint based on the CNN/DM dataset. This resulted in about 1 Rouge-1 point improvement.

**BART** For BART and BART$_{\text{XSUM}}$ finetuning experiments, we train all the models for 500 steps with 20% warm-up for an approximate training time of 45 minutes. This is equivalent to 5 epochs, though we initially allowed BART to train for up to 20 epochs and found that the model quickly overfits to the training set (as evidenced by poor performance on the dev set).

Through manual tuning, we achieved the best results by reducing the training time. Also in manual tuning, we first ran the experiments on four learning rates, 2e-5, 3e-5, 4e-5, and 5e-5 and controlled for all other hyperparameters. We then tested three different seeds, again controlling for all other parameters. Finally, we tested two batch sizes, 2048 tokens per batch and 1024 tokens per batch.

[19] https://github.com/mswellhao/PacSum

[20] https://github.com/nlpyang/PreSumm

[21] https://github.com/maszhongming/MatchSum

**CATTS** In the abstract-only setting, we train CATTS for 11,000 total steps for a total training time of 2.5 hours. For AIC, we train CATTS for 45,000 total steps for a total training time of 10 hours. This also equivalent to 5 epochs of training. We do not perform tuning on the training hyperparameters for CATTS, instead opting to use the same parameters as the baseline BART models.

## E Mean ROUGE test results

| | **Abstract-only** | | | **AIC** | | |
|---|---|---|---|---|---|---|
| **Method** | R1 | R2 | RL | R1 | R2 | RL |
| BART | 31.1 | 10.7 | 24.4 | 30.7 | 10.6 | 24.4 |
| BART$_{\text{XSUM}}$ | 30.1 | 10.7 | 24.1 | 31.0 | 10.9 | 24.7 |
| CATTS | 31.5 | 11.0 | 24.9 | †31.9 | †**11.8** | †**25.6** |
| CATTS$_{\text{XSUM}}$ | †**31.7** | **11.1** | †**25.0** | †**32.1** | †11.6 | †25.4 |

Table 10: Test set results using mean Rouge scores instead of max for abstractive methods. We use † to indicate CATTS variants that significantly ($p < 0.05$) outperform their corresponding BART baseline.

## F TLDR-PR annotation instructions

Below are the instructions provided to annotators rewriting peer-review comments.

**Task:** We want to collect a dataset of short summaries of CS papers, but it's hard to get people to read and write summaries about entire papers. Instead, we collected a dataset of peer reviewer comments, in which many CS researchers have read and written reviews of papers. Often, a reviewer's comments will also include a summary of the paper they've read. Our task is given the title and first 128 words of a reviewer comment about a paper, re-write the summary (if it exists) into a single sentence or an incomplete phrase. Summaries must be no more than one sentence. Most summaries are between 15 and 25 words. The average rewritten summary is 20 words long.

**What might be included in your re-write?**

1. What subfield is their work in?
2. What problem are they trying to solve?
3. What did the paper do?
4. Why should you care/how is it novel?

**What to exclude when re-writing a comment:** Not everything in the reviewer comment belongs in the summary. We purposefully leave out:

- Reviewer decisions/opinions (accept, reject, suggestions, etc.)
  - "The paper is well-written and it is quite easy to follow along with the discussion."
- Background information/ previous work
  - "The authors propose a method for learning node representations which, like previous work (e.g. node2vec, DeepWalk), is based on the skip-gram model."
  - "In particular, when node2vec has its restart probability set pretty high, the random walks tend to stay within the local neighborhood (near the starting node)."
- Excessive details about methodology
  - "Whereas node2vec may sample walks that have context windows containing the same node, the proposed method does not as it uses a random permutation of..."

**Enter "None" for the summary for the following conditions:**

- The comment is entirely the reviewer's opinions about the paper
- The reviewer's summary carries heavy sentiment about the paper
  - "This paper presents a method that is not novel or interesting"
  - This applies when the sentiment is so heavy that you are unable to write a summary.
- If the comment is about a paper that is out of your domain of expertise.