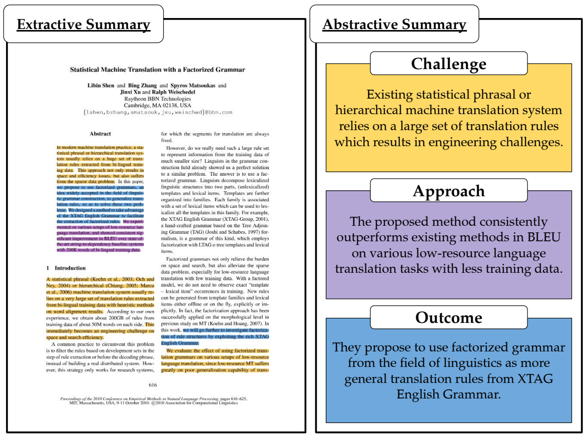
ACLSum: A New Dataset for Aspect-based Summarization of Scientific Publications

Sotaro Takeshita[1], Tommaso Green[1], Ines Reinig[1],
Kai Eckert[2], Simone Paolo Ponzetto[1]
[1]Data and Web Science Group, University of Mannheim, Germany
[2]Mannheim University of Applied Sciences, Mannheim, Germany
{sotaro.takeshita, tommaso.green, ines.reinig, ponzetto}@uni-mannheim.de
k.eckert@hs-mannheim.de

Abstract

Extensive efforts in the past have been directed toward the development of summarization datasets. However, a predominant number of these resources have been (semi)-automatically generated, typically through web data crawling. This resulted in subpar resources for training and evaluating summarization systems, a quality compromise that is arguably due to the substantial costs associated with generating ground-truth summaries, particularly for diverse languages and specialized domains. To address this issue, we present **ACLSum**, a novel summarization dataset carefully crafted and evaluated by domain experts. In contrast to previous datasets, **ACLSum** facilitates multi-aspect summarization of scientific papers, covering challenges, approaches, and outcomes in depth. Through extensive experiments, we evaluate the quality of our resource and the performance of models based on pretrained language models (PLMs) and state-of-the-art large language models (LLMs). Additionally, we explore the effectiveness of extract-then-abstract versus abstractive end-to-end summarization within the scholarly domain on the basis of automatically discovered aspects. While the former performs comparably well to the end-to-end approach with pretrained language models regardless of the potential error propagation issue, the prompting-based approach with LLMs shows a limitation in extracting sentences from source documents.[1]



Statistical Machine Translation with a Factorized Grammar

Libin Shen and **Bing Zhang** and **Spyros Matsoukas** and
Jinxi Xu and **Ralph Weischedel**
Raytheon BBN Technologies
Cambridge, MA 02138, USA
{lshen,bzhang,smatsouk,jxu,weisched}@bbn.com

Abstract

In modern machine translation practice, a statistical phrasal or hierarchical translation system usually relies on a huge set of translation rules extracted from bi-lingual training data. This approach not only results in space and efficiency issues, but also suffers from the sparse data problem. In this paper, we propose to use factorized grammars, an idea widely accepted in the field of linguistic grammar construction, to generalize translation rules, so as to solve these two problems. We designed a method to take advantage of the XTAG English Grammar to facilitate the extraction of factorized rules. We experimented on various setups of low-resource language translation, and showed consistent significant improvement in BLEU over state-of-the-art string-to-dependency baseline systems with 200K words of bi-lingual training data.

1 Introduction

A statistical phrasal (Koehn et al., 2003; Och and Ney, 2004) or hierarchical (Chiang, 2005; Marcu et al., 2006) machine translation system usually relies on a very large set of translation rules extracted from bi-lingual training data with heuristic methods on word alignment results. According to our own experience, we obtain about 200GB of rules from training data of about 50M words on each side. This immediately becomes an engineering challenge on space and search efficiency.

A common practice to circumvent this problem is to filter the rules based on development sets in the step of rule extraction or before the decoding phrase, instead of building a real distributed system. However, this strategy only works for research systems, for which the segments for translation are always fixed.

However, do we really need such a large rule set to represent information from the training data of much smaller size? Linguists in the grammar construction field already showed us a perfect solution to a similar problem. The answer is to use a factorized grammar. Linguists decompose lexicalized linguistic structures into two parts, (unlexicalized) templates and lexical items. Templates are further organized into families. Each family is associated with a set of lexical items which can be used to lexicalize all the templates in this family. For example, the XTAG English Grammar (XTAG-Group, 2001), a hand-crafted grammar based on the Tree Adjoining Grammar (TAG) (Joshi and Schabes, 1997) formalism, is a grammar of this kind, which employs factorization with LTAG e-tree templates and lexical items.

Factorized grammars not only relieve the burden on space and search, but also alleviate the sparse data problem, especially for low-resource language translation with few training data. With a factored model, we do not need to observe exact "template – lexical item" occurrences in training. New rules can be generated from template families and lexical items either offline or on the fly, explicitly or implicitly. In fact, the factorization approach has been successfully applied on the morphological level in previous study on MT (Koehn and Hoang, 2007). In this work, we will go further to investigate factorization of rule structures by exploiting the rich XTAG English Grammar.

We evaluate the effect of using factorized translation grammars on various setups of low-resource language translation, since low-resource MT suffers greatly on poor generalization capability of trans-

616

Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, pages 616–625,
MIT, Massachusetts, USA, 9-11 October 2010. ©2010 Association for Computational Linguistics

Existing statistical phrasal or hierarchical machine translation system relies on a large set of translation rules which results in engineering challenges.

The proposed method consistently outperforms existing methods in BLEU on various low-resource language translation tasks with less training data.

They propose to use factorized grammar from the field of linguistics as more general translation rules from XTAG English Grammar.

Figure 1: A data sample from **ACLSum**. Each document is complemented with manually-crafted and validated summaries for both extractive and abstractive setups on three different aspects. We annotate salient sentences to be used as extractive summaries and then write abstractive summaries by merging annotated sentences.

1 Introduction

The availability of high-quality datasets annotated with ground-truth human judgements has been a staple of the elements required to advance research in NLP for a very long time, dating back to the very dawn of statistical NLP (Marcus et al., 1993). Unfortunately, the availability of such resources has been quite scarce in the domain of text summarization of scientific papers (Koh et al., 2022). A prevalent approach in summarization from the last few years is to semi-automatically collect text snippets from the Internet that would serve as pseudo-summaries, with only partial quality control (Kryscinski et al., 2019; Tejaswin et al., 2021). While this enabled the creation of large datasets for data-hungry learning methods, it also makes it challenging to truly capture the summarization capabilities of models. While new metrics have been proposed to improve the automatic evaluation of summarization systems (Zhang et al., 2020b; Deutsch et al., 2021), only a few works approach this challenge from the standpoint of dataset quality. In machine translation, Freitag et al. (2020) shows that the low correlation between human and automatic evaluation is not only caused by the nature of the evaluation metrics but also by the lack of proper reference translations. Zhang et al. (2023b)

[1] Our data is released at https://github.com/sobamchan/aclsum.

Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 6660–6675
June 16-21, 2024 ©2024 Association for Computational Linguistics

showed that the agreement between an evaluation metric and human judgements can be increased by improving reference summaries with no modification to the metric itself. This is also true for the domain on which we focus in this paper, namely the automatic multi-aspect summarization of scientific papers, where we want to preserve output truthfulness to produce a correct summary along three different aspects of scientific work: the main challenge addressed in the paper, the approach developed to overcome it and the overall outcome of the work.

In this paper, we aim to foster reliable development of summarization systems from the dataset perspective. Concretely, we present **ACLSum**, a new multi-aspect extreme summarization dataset manually annotated and validated by domain experts. We propose a two-stage summary annotation approach where, for each of the proposed aspects, the annotators first select aspect-relevant sentences in the source documents and then use these to produce an abstractive summary. The process is represented in Figure 1: the left-hand side contains the paper with the color-coded relevant sentences for each of the aspects, while the right-hand side contains summaries for each aspect. This results in each source document having two kinds of gold standard annotations, namely (1) the set of sentences relevant to each of the aspects and (2) abstractive reference summaries. We evaluate the quality of our dataset through manual validation from domain experts.

Using **ACLSum**, we perform three lines of experiments to benchmark the existing state-of-the-art baseline models and to validate a heuristic commonly used to construct extractive summarization datasets. First, we compare two approaches for text summarization with pretrained language models (PLMs): (i) end-to-end summarization, in which the PLM directly produces a summary from the source document, and (ii) extract-then-abstract summarization, in which an extractive model first extracts sentences that are then used as input to the PLM to generate the summary. The unique property of our dataset is having gold annotations for both extractive and abstract summaries. This enables a fine-grained analysis: we quantitatively show that generative models suffer more when the relevant information is scattered across the source document, thus requiring them to perform a higher level of abstraction to produce final summaries. We find that the extract-then-abstract approach outperforms the end-to-end counterpart in most cases even with the potential error propagation issue. Second, we shed light on recently developed large language models (LLMs) by training and evaluating Llama 2 (Touvron et al., 2023) in two different ways, namely through (i) end-to-end instruction-tuning and (ii) extract-then-abstract chain-of-thought-like training, where we instruct-tune the model to first generate references to the sentences in the source document that cover relevant aspects for the summary, and then merge these sentences to produce a final summary. Due to the poor performance in the extraction stage, the prior outperforms the latter by a considerable margin. Third, we evaluate a greedy algorithm used in previous work to induce silver-standard extractive summaries (Nallapati et al., 2017) and empirically show its low quality when properly evaluated against ground-truth annotations from human experts. Our contributions are the following ones:

1. **A new expert-annotated and validated multi-aspect summarization dataset** with both extractive and extreme abstractive summary annotations.
2. **An extensive and fine-grained evaluation** of PLM systems and instruction-tuned LLMs on aspect-based summarization of scientific papers.
3. **A benchmarking assessment of a greedy search heuristic** for extractive summary generation on our domain.

2 Related Work

A common practice to build summarization datasets is to find data on the Internet which can be used as a silver-standard proxy for document-summary pairs (Cohan et al., 2018; Kim et al., 2019; Hayashi et al., 2021), e.g., news articles and their lead sentences (Hermann et al., 2015; Narayan et al., 2018). While scalable and very practical to build datasets on a large scale, the resulting summaries exhibit a few fundamental limitations.

Unfaithful summaries. Maynez et al. (2020) show that XSum (Narayan et al., 2018) contains summaries that are unfaithful to source documents in the sense that these ‘summaries’ are rather ‘eye-catching’ sentences to draw readers to the corresponding articles for the news platform from which the dataset is extracted.

Noisy data. Kryscinski et al. (2019) report that CNN/DailyMail (Hermann et al., 2015) and Newsroom (Grusky et al., 2018) contain much noise, such as URLs and placeholder texts in their summaries. More recently, Koh et al. (2022) show that more than 60% of the reference summaries in the test set of the arXiv dataset which is built for long document summarization contain noises. Moreover, in a preliminary study, we investigated automatic language detection models and found that 0.4% samples in the test set of XSum are, for instance, written in Welsh (as opposed to English).

Legal issues. The two most well-used datasets, namely CNN/DailyMail and XSum, raise various legal issues (e.g., copyrighted content) and are not publicly available (Wang et al., 2022).

Missing gold extractive labels. While extractive and aspect-based summarization are both active research subjects, there are no freely available datasets with ground-truth labels for such tasks. For extractive summarization, the *de facto* standard approach has been relying on a heuristic-based algorithm that automatically induces labels from abstractive summarization datasets without validating its effectiveness (Nallapati et al., 2017), including recent improvements from Xu and Lapata (2022).

The work closest to ours is SQuALITY from Wang et al. (2022). While this work shares the core motivation with our work, which is to build a reliable and validated summarization dataset, our dataset has several different properties. First, besides the abstractive reference summaries, our dataset also has passage annotations (i.e., aspects) that can serve as gold labels for extractive summarization. Second, in contrast to the SQuALITY, which provides question-focused summaries, our dataset has multi-aspect summaries more suitable for our target scholarly domain. Third, SQuALITY uses novel stories as its source documents, whereas our dataset uses research articles from the field of NLP, which makes our dataset highly domain-specific and challenging. Lastly, the size: our **ACLSUM** contains 250 documents, which is more than twice larger than the 100 documents provided in SQuALITY. Another work similar to ours is SciTLDR (Cachola et al., 2020), a collection of papers from computer science and one-sentence summaries, later extended by Takeshita et al. (2022) for cross-lingual summarization. This work has inspired us to design reference summaries in our dataset to have one-sentence summaries. Our work differs from theirs in (i) the structure of summaries: ours has multi-aspect summaries instead of one overview summary, (ii) type of annotations: while SciTLDR contains only abstractive reference summaries, our dataset also contains annotations of the relevant sentences.

3 Dataset creation

Source documents. We focus on the summarization of scholarly documents (Erera et al., 2019; Fok et al., 2022) with a focus on NLP papers because of the availability of large amounts of freely available text (Bird et al., 2008) in a challenging domain setting. The focus on NLP is additionally driven by the surge in recent years of publications in our field, and the consequent need for summarization systems, as well as the availability of domain-expert annotators at our disposal.

We take research papers published in five major NLP conferences, namely ACL, NAACL, EMNLP, EACL, and AACL from 1974 to 2022, and use Grobid (Lope, 2008–2023) to extract Abstract, Introduction, and Conclusion (AIC) sections from PDF files. We apply a bucket-based sampling for selecting documents to be annotated to maintain the diversity of documents in our dataset. Due to the increasing number of published papers in the last decade, random sampling would be biased towards recent publications. To avoid this, we divide the papers into different buckets for each combination of year and venue and uniformly sample from them to create a pool of papers to be annotated.

Summary aspects. Recently, there have been several works proposing datasets with multiple summaries for each document to cover different aspects of source documents (Hayashi et al., 2021; Yang et al., 2023). Research articles are also multifaceted documents with multiple aspects (Fisas et al., 2015). Consequently, our annotated passages and abstractive summaries cover three different aspects, namely: (i) Challenge: *The current situation faced by the researcher; it will normally include a Problem Statement, the Motivation, a Hypothesis and/or a Goal.*, (ii) Approach: *How they intend to carry out the investigation, comments on a theoretical model or framework.*, and finally (iii) Outcome: *Overall conclusion that should reject or support the research hypothesis.*. We operationalize the definition of these three aspects using the definitions from Fisas et al. (2015).

Annotation process. We annotate the dataset by relying on domain experts (graduate students in NLP) instead of crowd-sourcing platforms, which are known to have quality issues (Zhang et al., 2023a). We use a two-stage process to produce reference summaries. In the first stage, we review each sentence in the source document and annotate it with an aspect if it contains information relevant to (one of the aspects of) the summary. In the second stage, the annotator writes a summary using selected sentences with a 25-word limit, to maintain the average sentence length of the source document. This property makes **ACLSUM** an extreme summarization dataset, a more challenging setup (Narayan et al., 2018) than traditional summarization, which is suitable for the scholarly domain since researchers need to consume a steadily increasing number of papers (Bornmann and Mutz, 2015). We do not make use of any models or systems for our annotation task to avoid any biases that could favor certain models in evaluation (Deutsch et al., 2022). The full annotation guidelines can be found in the Appendix A.1.

4 ACLSUM

Table 1 presents statistics of our dataset. **ACLSUM** consists of 250 documents (i.e., AICs) with an average length of approximately 40 sentences and 1,000 words. The average length of the annotated aspects is comparable, with passages describing approaches being slightly longer ('avg. # words' in the Table). The low numbers of words that only appear in the reference summaries but not in the source documents ('avg. # new words') indicate that the summaries in **ACLSUM** have less chance of including information unfaithful to the source documents. The compression ratios based on relevant aspect sentences (namely, the average number of words of AICs to the average number of words per annotated aspect, column 9) ranges between 8.4 and 5.6, whereas for abstractive summaries it ranges between 40.1 and 42.6, thus exhibiting the high level of abstraction required for models to perform abstractive summarization.

We validate the quality of the annotations in our dataset by taking 75 summaries (25 AICs times three aspects) from our validation split and let two additional domain experts evaluate the quality. We use the four criteria proposed by Fabbri et al. (2021), namely relevance, consistency, coherence, and fluency. We do not evaluate *fluency* for extractive summary annotations by assuming the texts published at ACL conferences are well polished, and also *consistency* since the sentences are extracted without modifications from source documents. For abstractive summaries, *coherence* is excluded since this measures "the quality of all sentences collectively" (Fabbri et al., 2021) while our reference summaries are composed of a single sentence. The complete annotation guidelines can be found in the Appendix A.2. On all aspects and criteria, our both extractive and abstractive annotations achieve above 4 on a 1-5 Likert scale (Table 2, 3), and especially high Relevance scores in both and Consistency in abstractive summaries show that our reference summaries capture essential information in the source documents. Because one extractive summary of a document is a set of sentences annotated through the source documents stitched together (see Figure 2), scores on Coherence are lower than other aspects, indicating a limitation of extractive summarization. We measure inter-annotator agreement between both annotators in terms of percentage agreement. The agreement on Relevance is remarkably high (96% for Challenge and Outcome, 76% for Approach) and satisfactory for Consistency (68%, 72% and 76% for Challenge, Approach and Outcome respectively) for scores in Table 3. On Fluency however, annotators agreed less frequently for all three aspects (52%, 52% and 48%), which may suggest a more subjective nature of the Fluency measure.

We show in Figure 2 the relative positions of sentences annotated with aspects, highlighting how the three different aspects are highly interspersed across documents, thus indicating that indeed models are required to attend to many different parts of the document for each aspect.

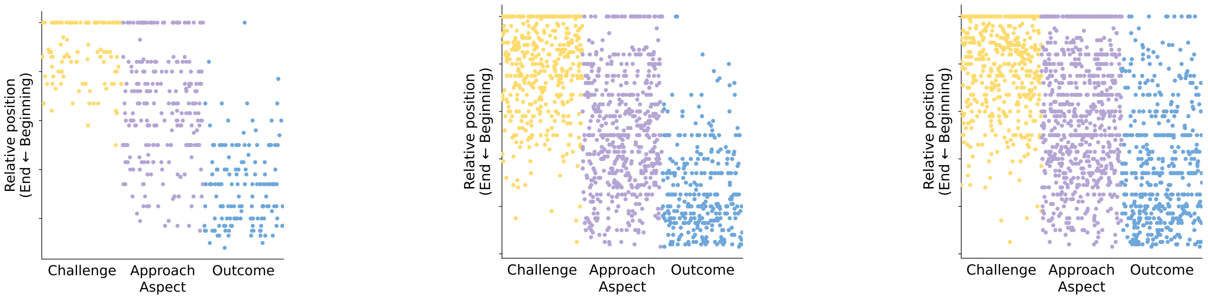
5 Experiments and Results

We next use **ACLSUM** to answer the following research questions:

- **RQ1**: Which approach using PLMs, i.e., two-stage extract-then-abstract or end-to-end abstractive summarization, performs best on our dataset?
- **RQ2**: Which tuning strategy for LLMs, i.e., two-step chain-of-thought or end-to-end abstractive summarization, is better for our task?
- **RQ3**: How does a commonly used heuristic to induce silver-standard extractive summaries perform against our manually annotated aspects?

	Document				Aspects				Summaries				
Aspect	**# doc. train/val/test**	**avg. # words**	**avg. # sent.**	**# vocab**	**avg. # sent.**	**avg. # words**	**# vocab**	**comp. ratio**	**avg. # words**	**avg. # vocab**	**avg. # new words**	**comp. ratio (to doc.)**	**comp. ratio (to asp.)**
Challenge	100/50/100	914.7	38.45	14k	4.3	109.0	4.5k	8.4	22.5	1.8k	3.3	40.1	4.8
Approach					6.6	162.6	4.8k	5.6	22.7	1.7k	2.1	40.1	7.1
Outcome					4.4	110.3	3.9k	8.3	21.3	1.4k	2.2	42.6	5.1

Table 1: Statistics of **ACLSum** including documents, sentences annotated with aspects and summaries.



(a) Abstract

(b) Introduction

(c) Conclusion

Figure 2: Relative positions of relevant sentences for each aspect (Challenge, Approach and Outcome).

Aspect	Relevance	Coherence
Challenge	4.86	4.34
Approach	5.00	4.04
Outcome	5.00	4.26

Table 2: Manual validation of extractive aspect annotations on a 1-5 Likert scale.

Aspect	Relevance	Consistency	Fluency
Challenge	4.98	4.85	4.65
Approach	4.82	4.70	4.56
Outcome	4.96	4.74	4.54

Table 3: Manual validation of reference abstractive summaries on a 1-5 Likert scale.

5.1 RQ1: Extract-then-abstract vs. end-to-end

The extract-then-abstract approach (EtA) has recently become more widely used in the literature (Hsu et al., 2018; Mao et al., 2022). It consists of first using an extractive model to select relevant sentences in the source documents and then deploying an abstractive model to merge extracted sentences into a summary: this is opposed to end-to-end summarization (E2E) in which the model directly generates the summary using the entire source document as input (Liu et al., 2022; Zhang et al., 2020a). While the extract-then-abstract approach potentially suffers from the error propagation problem, it can benefit from more efficient inference (due to the reduced number of sentences fed to the abstractive models) and, arguably, transparency on the provenance of the summary (since summaries are typically generated from a few extracted sentences). In our setting, **ACLSum** enables ground-truth evaluation of both stages since our dataset contains both annotated aspects (which can be used to provide extractive summaries) and abstractive summaries.

Experimental setup. For extractive models, we evaluate the Sentence-T5 model proposed by Ni et al. (2021) in three different sizes (BASE, LARGE, XL) since it was shown to be a powerful text encoder in a recent study by Muennighoff et al. (2023). We use the ST5-Enc mean variant, which only uses the encoder of T5 and applies mean-pooling to get sentence representations. We train a binary logistic regression model on top of Sentence-T5 representations to classify if the text is relevant or irrelevant to producing the summary for the aspect at hand. The annotations for each sentence act as labels (positive class if the sentence was selected by the annotators, negative otherwise.). For the abstractive model, we evaluate BART (Lewis et al., 2020) and T5 (Raffel et al., 2020) in two sizes (BASE, LARGE). Additionally, we evaluate the setup – which we refer to hereafter as 'Gold' – for which we feed manually annotated gold extractive summaries to the abstractive model.

	Challenge			Approach			Outcome		
Model	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
ST5$_{\text{BASE}}$	63.2	75.7	65.0	68.0	77.6	70.0	71.2	84.1	74.9
ST5$_{\text{LARGE}}$	63.6	75.5	65.7	69.7	79.7	72.1	71.5	83.9	75.2
ST5$_{\text{XL}}$	**63.9**	**76.3**	**66.0**	**70.3**	**80.0**	**72.7**	**71.8**	**84.8**	**75.6**

Table 4: Performance of Sentence-T5 in three different sizes on each aspect of our dataset with three metrics, **F1**, (**P**)recision, and (**R**)ecall.

This lets us evaluate the upper-bound capability of the abstractive model within the pipeline in isolation. In the E2E setup, i.e., direct summarization, the abstractive model takes the entire document as input. We carry out separate training procedures for each aspect for both approaches.

We train the binary logistic regression classifier with L2 regularization with $C = 1.0$ regularization strength. For each possible choice of extractive model in the EtA setup, we use its predictions to both train and evaluate the abstractive model that follows in the pipeline. For all abstractive models, we perform grid-search using the following grid: learning rate $lr \in$ {1e-5, 3e-5, 5e-5}, batch size $B \in \{4, 8, 16\}$ for BART$_{\text{BASE/LARGE}}$ and $B \in \{2, 4, 8\}$ for T5$_{\text{BASE/LARGE}}$. During the hyperparameter search, we use the validation split of our dataset with a fixed seed to find the best combination of parameters according to the loss. We report each score as an average over three differently seeded models. For evaluation, we report standard ROUGE scores (Lin, 2004) and BERTScore (Zhang et al., 2020b) with SciBERT (Beltagy et al., 2019) as the underlying model.

Results and discussions. Table 4 shows the performance of the extractive models. As the size of the underlying model increases, the performance on all three aspects improves. All models perform worse on the Challenge aspect compared to the other two, possibly because of the fewer annotated passages for this aspect (cf. Table 1).

Table 5 shows the results for both the EtA and E2E approaches. On the aspects of Approach and Outcome, EtA outperforms E2E when the gold extraction labels are used. For the BART$_{\text{BASE}}$ model, even when predictions from an extractive model (ST5$_{\text{XL}}$) are used instead of gold labels, the two-stage approach outperforms the end-to-end approach thus indicating that the two-staged approach suffers from error propagation when there is only a weak extractive model available.

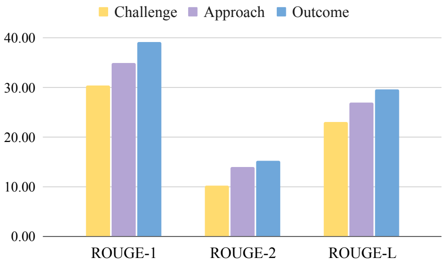


Figure 3: Average ROUGE scores between aspect-annotated sentences and abstractive summaries.

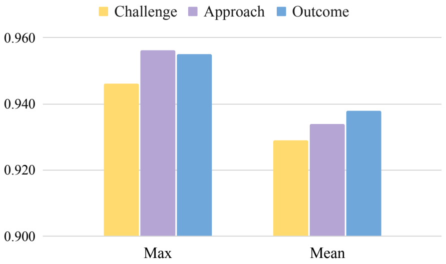


Figure 4: Maximum and average similarity of each aspect-annotated sentence to the centroid of sentences for that aspect using Sentence-T5 embeddings.

All models perform substantially worse on the Challenge aspect, even being outperformed by a simple baseline (Lead-1). To better understand this result, we question the degree to which summarization models are required to synthesize relevant sentences from a source document, that is: how much abstraction is needed? To answer this question, we perform three lines of analysis. We first compute the average ROUGE scores between each sentence annotated as a relevant aspect and the corresponding abstractive summary. Higher scores indicate that aspect-relevant sentences contain more information required to form a summary, i.e., less abstraction is needed. Results in Figure 3 show that the Challenge aspect demands models to perform a higher-level abstraction compared to the other two aspects, making it the most challenging aspect in our dataset. We next compute for each aspect the maximum and average similarity of each aspect-annotated sentence to the centroid of sentences for that aspect in the document using sentence embeddings obtained by the Sentence-T5 (-base) model. Higher numbers indicate that relevant sentences are semantically similar to each other. The results in Figure 4 show that, indeed, the relevant sentences in the Challenge portion are semantically more scattered than other aspects. This asks models to

Model		Extractor	Challenge				Approach				Outcome			
			R-1	R-2	R-L	BS	R-1	R-2	R-L	BS	R-1	R-2	R-L	BS
Lead-1		-	26.86	9.48	20.61	0.604	23.23	7.68	18.31	0.589	16.92	2.50	12.19	0.583
TextRank		-	18.47	2.43	13.42	0.572	19.29	4.79	13.87	0.591	16.86	2.79	11.91	0.594
BART$_{BASE}$	**EtA**	**ST5$_{BASE}$**	18.47	2.43	13.42	0.617	43.31	19.88	35.61	0.730	41.39	19.70	34.86	0.723
		ST5$_{LARGE}$	19.02	2.57	13.86	0.620	44.51	21.30	37.71	0.739	39.47	18.40	33.26	0.719
		ST5$_{XL}$	19.00	2.31	13.76	0.622	45.12	21.17	37.76	0.739	39.82	18.83	34.17	0.722
		Gold	18.66	2.79	13.68	0.618	**45.70**	**22.17**	**38.52**	**0.741**	**45.59**	**21.73**	**37.15**	**0.739**
	E2E	-	**21.59**	**3.88**	**15.63**	**0.627**	42.98	18.75	35.72	0.728	37.89	16.33	31.69	0.709
BART$_{LARGE}$	**EtA**	**ST5$_{BASE}$**	18.35	2.15	13.03	0.611	44.55	21.44	37.67	0.732	40.30	19.28	33.97	**0.719**
		ST5$_{LARGE}$	19.61	2.51	13.95	0.614	44.21	20.69	37.11	0.731	38.14	17.70	31.98	0.708
		ST5$_{XL}$	12.77	1.59	9.43	0.597	43.88	20.48	37.09	0.677	39.18	18.45	33.43	0.665
		Gold	17.18	2.27	12.58	0.585	**47.84**	**23.94**	**40.14**	0.674	**42.82**	**20.62**	**35.17**	0.666
	E2E	-	**20.00**	**3.76**	**14.92**	**0.623**	44.95	21.82	38.27	**0.741**	36.22	15.94	30.61	0.699
T5$_{BASE}$	**EtA**	**ST5$_{BASE}$**	18.93	2.45	13.56	0.610	44.82	22.40	38.36	0.732	42.25	21.72	34.98	0.721
		ST5$_{LARGE}$	18.47	2.35	13.30	0.609	45.10	22.40	38.76	0.735	42.10	21.28	34.56	0.722
		ST5$_{XL}$	18.19	2.31	12.99	0.592	45.97	23.32	39.80	0.668	40.27	19.23	33.30	0.639
		Gold	19.13	2.62	13.58	0.583	**47.78**	**25.07**	**40.96**	0.671	**46.60**	**24.51**	**38.49**	0.653
	E2E	-	**21.32**	**4.75**	**15.84**	**0.623**	47.39	24.51	40.79	**0.746**	42.27	21.53	35.81	**0.724**
T5$_{LARGE}$	**EtA**	**ST5$_{BASE}$**	19.18	2.91	14.04	0.609	45.93	23.21	39.18	0.736	42.88	22.30	35.89	0.726
		ST5$_{LARGE}$	18.80	2.96	13.59	0.612	45.51	22.74	39.24	0.738	41.76	21.52	35.71	**0.724**
		ST5$_{XL}$	19.24	2.76	13.68	0.591	46.30	23.56	40.22	0.666	42.32	21.55	35.94	0.652
		Gold	19.37	3.01	13.85	0.591	**49.05**	**26.04**	**42.30**	0.669	**46.57**	**24.39**	**38.21**	0.659
	E2E	-	**21.17**	**4.92**	**16.13**	**0.626**	46.95	23.67	40.41	**0.743**	41.41	20.90	34.79	0.721

Table 5: Performance of EtA and E2E models (best results for each aspect and metric are **bolded**).

Aspect	Relevance	Consistency	Fluency
Challenge	3.32 (3.06)	3.72 (3.53)	4.64 (4.82)
Approach	3.28 (3.21)	4.00 (4.11)	4.48 (4.58)
Outcome	3.36 (3.38)	3.84 (3.95)	4.20 (4.29)

Table 6: Manual evaluation of best-performing models on a 1-5 Likert scale. Scores in parentheses are computed when summaries which violate the length limitation (25 words) are ignored for the evaluation.

merge more semantically dissimilar sentences to produce a final summary. Lastly, we compute the entropy over the appearance of 1000 most frequent words in aspect-relevant sentences for each aspect and find that the distribution of the Challenge has the highest entropy (Challenge: 9.39, Approach: 9.21, Outcome: 9.06). This indicates that models have fewer cues for aspect-relevant sentences for the Challenge than others making it more challenging to detect relevant information from the source documents. Together, these findings indicate that the Challenge aspect of our dataset is harder to capture in summaries because models are required to perform a higher level of abstraction over sentences dissimilar to one another with fewer cues.

Additionally to the automatic evaluation, we also perform a manual evaluation to qualitatively assess the quality of generated summaries. To this end, we take the best-performing models for each aspect excluding EtA Gold setups which require access to the gold aspect annotations, and evaluate 25 samples from the test split using three criteria as our annotation quality validation setup (described in Section 4). Table 6 shows the results. While generated summaries mark high fluency scores in all aspects, they suffer in being relevant and consistent to source documents. We also observe that the generated summaries are often longer than the length limitation in the reference summaries. Concretely, 76% of evaluated summaries violate the 25-word limitation. Since longer summaries have a higher chance of including relevant information, ignoring the samples violating the length limitation lowers the relevant score on two out of three aspects.

5.2 RQ2: CoT vs. E2E instruct-tuning.

Next, we take the popular Llama 2 model (Touvron et al., 2023) as a representative of recently proposed LLMs to evaluate its summarization abilities using **ACLSUM** by fine-tuning it in two different ways. The first strategy simply fine-tunes the model to generate a summary given an instruction (E2E). In contrast, the second strategy, dubbed extract-

Instruction
Generate the indices of the sentences in the given research paper that are relevant to the paper's challenge, and then summarize them into one sentence.
Input
0: In this paper, we explore correlation... 1: Using the correlation measure... 2: Different from previous studies, we propose an... 3: The correlations are further [...]
Output
Index: 17, 18, 19, 20, 22, 23, 24 Summary: A generally accessible NER system for QA systems produces a larger answer candidate set which would be hard for current surface word-level ranking methods.

Table 7: A training sample for EtA-CoT tuning.

		R-1	R-2	R-L	BS	F1
Challenge	**Zero-Shot**	21.37	5.39	14.55	0.61	-
	E2E	**30.06**	**11.33**	**23.87**	**0.67**	-
	EtA-CoT	12.48	4.84	9.25	0.48	15.9
Approach	**Zero-Shot**	29.25	11.77	21.72	0.66	-
	E2E	**44.01**	**23.03**	**38.58**	**0.73**	-
	EtA-CoT	26.59	13.15	22.46	0.59	10.0
Outcome	**Zero-Shot**	27.89	11.12	20.47	0.65	-
	E2E	**32.85**	**13.39**	**27.23**	**0.68**	-
	EtA-CoT	23.85	11.03	20.03	0.57	5.1

Table 8: Performance of Llama 2 when trained on our dataset (E2E or EtA-CoT) and Llama 2 Chat with zero-shot prompting.

then-abstract chain-of-thought (EtA-CoT), trains the model to generate a summary by first generating a list of indexes to sentences that are relevant to produce the summary as an immediate reasoning step (Wei et al., 2022). We build this instruction-tuning dataset using our extractive and abstractive summarization annotations.

Experimental setup. We follow the training scheme used by Taori et al. (2023): we apply LoRA (Hu et al., 2022) and enable gradient checkpointing to fine-tune the Llama 2 7B. We train one model on a joint dataset of all three aspects and specify the target aspect in the instruction. We only trained the Llama 2 model but not its instruction-tuned variant since in our preliminary study we only observed marginal differences between them. A training data sample used for EtA-CoT is shown in Table 7. We keep the batch size to 1, number of input tokens to 4500, and test for learning rate ∈ {1e-4, 3e-4, 5e-4}. We use the validation split to find the best hyper-parameter and report the average performance of three differently seeded models at test time. We also report results by zero-shot prompting using the instruction-tuned Llama 2 Chat model.

	Challenge			**Approach**			**Outcome**		
Type	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
R-1	80.7	66.1	69.0	82.7	61.1	63.4	82.6	63.9	66.9
R-2	**82.3**	**66.7**	**70.1**	**84.7**	**61.2**	**63.5**	82.9	**64.5**	67.1
R-L	80.2	64.2	66.9	85.7	60.0	61.8	**84.1**	64.2	**67.4**

Table 9: Performance of greedy algorithm. The numbers by the best performing ROUGE function are **bolded**.

Results and discussions. Table 8 presents ROUGE scores, BERTScores, and extraction performance using our gold extractive labels measured by F1 for EtA-CoT models. While it is difficult to compare performance between Llama 2 and T5 due to the massive difference in model sizes (Llama 2: 7B vs. T5_{LARGE}: 770M parameters), the E2E model with Llama 2 substantially outperforms the latter on the Challenge aspect. However, it performs comparably to PLMs-based models on the other two aspects. Between the two training strategies, the E2E outperforms EtA-CoT, although it receives an additional extractive training signal during training. Poor F1 scores indicate they fail at the extraction stage, and the errors propagate to the abstraction stage. To see if models are indeed extracting sentences from the source documents, we compute the average success rates by checking if (1) models predict at least one index to a sentence and (2) predicted indexes are in the valid range, i.e., positive indexes that are smaller than the number of total sentences in the source document. We observe that 99% of the outputs successfully fruitful both criteria. This result shows that the models have learned the required output structure yet perform poorly on prediction. By comparing zero-shot prompting and end-to-end tuning, one can observe that even with LLMs that are shown to be strong in summarization without any training, our dataset can help to improve their performance.

5.3 RQ3: How good is the heuristic for inducing extractive summarization labels?

Existing works on extractive summarization systems use silver labels induced by a heuristic algorithm to work around the lack of ground-truth annotations for extractive summarization (Nallapati et al., 2017; Narayan et al., 2018) by producing extractive labels given a source document and the corresponding abstractive summary. While this approach has been the *de facto* standard (Liu and Lapata, 2019; Pilault et al., 2020; Hsu et al., 2018), no previous work assessed the quality of the heuristically induced labels against manually annotated gold labels because such evaluation would require a dataset, like **ACLSUM**, with both extractive and abstractive annotations.

We use the greedy algorithm proposed by Nallapati et al. (2017), which induces extractive labels by adding one sentence from the source document which maximizes the ROUGE score of the set of selected sentences w.r.t. the abstractive reference summary at each iteration until no remaining sentence can improve the ROUGE scores. The resulting set of selected sentences then is used as labels for extractive summarization. We run this algorithm over our dataset and evaluate the induced extractive labels against our manually annotated gold labels. Because some of the existing works do not explicitly mention the ROUGE function used to select the sentences, we compare the three common variants. Results are shown in Table 9. The best F1 score across ROUGE types and portions in our dataset is 70.1 which arguably indicates the rather low quality of the extractions produced by this method when compared to a human gold standard. To assess the quality of the silver labels as training data, we re-run the experiments with extract-then-abstract (EtA) pipelines in Section 5.1 by training extractive models on the silver labels instead of manually annotated gold labels. While pipelines with extractive models trained on gold labels outperform their counterparts trained on silver labels, the gaps are marginal. This result indicates that even though, manually created gold labels are preferred for accurate evaluations however silver labels would be sufficient for training purpose. Table 12 in the Appendix shows the full result.

6 Conclusion

In this paper, we presented **ACLSUM**, a manually crafted and validated multi-aspect summarization dataset for both extractive and abstractive summarization systems. Using **ACLSUM**, we performed experiments using summarization models based on pretrained language models and more recent large language models such as Llama, as well as evaluating a standard algorithm to create extractive summarization datasets. In future work, we plan to explore ways to use our annotated data to bootstrap and extend our dataset through (semi-)automatic data augmentation methods, as well as build datasets for other fields, including other areas of Computer Science and other domains, possibly in languages other than English, such as, e.g., publications from the social sciences and humanities. We additionally plan to explore ways to use our aspect-based single document summarization models to enable multi-document summarization of scientific publications, a yet under-researched setup with much potential to provide challenging tasks in the age of large-scale text understanding and generation.

7 Limitations

Our dataset is limited in two ways. Due to the difficulty of the annotation process, which needs to rely on experts in the scholarly domain, it contains only one reference summary for each document and aspect, and fewer samples compared to (semi)-automatically generated datasets. Moreover, we focus this initial contribution on scientific publications from a single field and language, namely English NLP papers from the ACL Anthology.

Acknowledgements

The work presented in this paper is funded by the German Research Foundation (DFG) under the VADIS (PO 1900/5-1; EC 477/7-1) and JOIN-T2 (PO 1900/1-2) projects, and also supported by the state of Baden-Württemberg through bwHPC and the German Research Foundation (DFG) through grant INST 35/1597-1 FUGG.

References

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. 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 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Steven Bird, Robert Dale, Bonnie Dorr, Bryan Gibson, Mark Joseph, Min-Yen Kan, Dongwon Lee, Brett Powley, Dragomir Radev, and Yee Fan Tan. 2008. The ACL Anthology reference corpus: A reference dataset for bibliographic research in computational linguistics. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Lutz Bornmann and Rüdiger Mutz. 2015. Growth rates of modern science: A bibliometric analysis based on the number of publications and cited references. *J. Assoc. Inf. Sci. Technol.*, 66(11):2215–2222. Publisher: Wiley.

Isabel Cachola, Kyle Lo, Arman Cohan, and Daniel Weld. 2020. TLDR: Extreme summarization of scientific documents. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4766–4777, Online. Association for Computational Linguistics.

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.

Daniel Deutsch, Tania Bedrax-Weiss, and Dan Roth. 2021. Towards Question-Answering as an Automatic Metric for Evaluating the Content Quality of a Summary. *Transactions of the Association for Computational Linguistics*, 9:774–789.

Daniel Deutsch, Rotem Dror, and Dan Roth. 2022. On the limitations of reference-free evaluations of generated text. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10960–10977, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Daniel Deutsch and Dan Roth. 2020. SacreROUGE: An open-source library for using and developing summarization evaluation metrics. In *Proceedings of Second Workshop for NLP Open Source Software (NLP-OSS)*, pages 120–125, Online. Association for Computational Linguistics.

Shai Erera, Michal Shmueli-Scheuer, Guy Feigenblat, Ora Peled Nakash, Odellia Boni, Haggai Roitman, Doron Cohen, Bar Weiner, Yosi Mass, Or Rivlin, Guy Lev, Achiya Jerbi, Jonathan Herzig, Yufang Hou, Charles Jochim, Martin Gleize, Francesca Bonin, Francesca Bonin, and David Konopnicki. 2019. A Summarization System for Scientific Documents. 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): System Demonstrations*, pages 211–216, Hong Kong, China. Association for Computational Linguistics.

Alexander R Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. SummEval: Re-evaluating summarization evaluation. *Trans. Assoc. Comput. Linguist.*, 9:391–409. Publisher: MIT Press - Journals.

Beatriz Fisas, Horacio Saggion, and Francesco Ronzano. 2015. On the Discoursive Structure of Computer Graphics Research Papers. In *Proceedings of The 9th Linguistic Annotation Workshop*, pages 42–51, Denver, Colorado, USA. Association for Computational Linguistics.

Raymond Fok, Andrew Head, Jonathan Bragg, Kyle Lo, Marti A. Hearst, and Daniel S. Weld. 2022. Scim: Intelligent Faceted Highlights for Interactive, Multi-Pass Skimming of Scientific Papers. Number: arXiv:2205.04561 arXiv:2205.04561 [cs].

Markus Freitag, David Grangier, and Isaac Caswell. 2020. BLEU might be Guilty but References are not Innocent. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 61–71, Online. 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.

Hiroaki Hayashi, Prashant Budania, Peng Wang, Chris Ackerson, Raj Neervannan, and Graham Neubig. 2021. WikiAsp: A Dataset for Multi-domain Aspect-based Summarization. *Transactions of the Association for Computational Linguistics*, 9:211–225.

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*, volume 28, pages 1693–1701, Montréal, Canada. Curran Associates, Inc.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Wan-Ting Hsu, Chieh-Kai Lin, Ming-Ying Lee, Kerui Min, Jing Tang, and Min Sun. 2018. A unified model for extractive and abstractive summarization using inconsistency loss. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 132–141, Melbourne, Australia. Association for Computational Linguistics.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Byeongchang Kim, Hyunwoo Kim, and Gunhee Kim. 2019. Abstractive Summarization of Reddit Posts with Multi-level Memory Networks. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2519–2531, Minneapolis, Minnesota. Association for Computational Linguistics.

Jan-Christoph Klie, Michael Bugert, Beto Boullosa, Richard Eckart de Castilho, and Iryna Gurevych. 2018. The INCEpTION platform: Machine-assisted and knowledge-oriented interactive annotation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 5–9, Santa Fe, New Mexico. Association for Computational Linguistics.

Huan Yee Koh, Jiaxin Ju, Ming Liu, and Shirui Pan. 2022. An empirical survey on long document summarization: Datasets, models, and metrics. *ACM Comput. Surv.*, 55(8).

Wojciech Kryscinski, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Neural text summarization: A critical evaluation. 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 540–551, Hong Kong, China. 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 Pretraining 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.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. 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 3730–3740, Hong Kong, China. Association for Computational Linguistics.

Yixin Liu, Pengfei Liu, Dragomir Radev, and Graham Neubig. 2022. BRIO: Bringing order to abstractive summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2890–2903, Dublin, Ireland. Association for Computational Linguistics.

Patrice Lope. 2008–2023. Grobid. `https://github.com/kermitt2/grobid`.

Ziming Mao, Chen Henry Wu, Ansong Ni, Yusen Zhang, Rui Zhang, Tao Yu, Budhaditya Deb, Chenguang Zhu, Ahmed Awadallah, and Dragomir Radev. 2022. DYLE: Dynamic latent extraction for abstractive long-input summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1687–1698, Dublin, Ireland. Association for Computational Linguistics.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On Faithfulness and Factuality in Abstractive Summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.

Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. 2023. MTEB: Massive Text Embedding Benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2014–2037, Dubrovnik, Croatia. Association for Computational Linguistics.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 3075–3081. AAAI Press.

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

Jianmo Ni, Gustavo Hernández Ábrego, Noah Constant, Ji Ma, Keith B. Hall, Daniel Cer, and Yinfei Yang. 2021. Sentence-T5: Scalable Sentence Encoders from Pre-trained Text-to-Text Models. ArXiv:2108.08877 [cs].

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Jonathan Pilault, Raymond Li, Sandeep Subramanian, and Chris Pal. 2020. On extractive and abstractive neural document summarization with transformer language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9308–9319, Online. 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. *J. Mach. Learn. Res.*, 21(140):1–67.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Sotaro Takeshita, Tommaso Green, Niklas Friedrich, Kai Eckert, and Simone Paolo Ponzetto. 2022. X-SCITLDR: cross-lingual extreme summarization of scholarly documents. In *JCDL '22: The ACM/IEEE Joint Conference on Digital Libraries in 2022, Cologne, Germany, June 20 - 24, 2022*, page 4. ACM.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

Priyam Tejaswin, Dhruv Naik, and Pengfei Liu. 2021. How well do you know your summarization datasets? In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3436–3449, Online. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. ArXiv:2307.09288 [cs].

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.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

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.

Yumo Xu and Mirella Lapata. 2022. Text summarization with oracle expectation. In *The Eleventh International Conference on Learning Representations*.

Xianjun Yang, Kaiqiang Song, Sangwoo Cho, Xiaoyang Wang, Xiaoman Pan, Linda Petzold, and Dong Yu. 2023. OASum: Large-scale open domain aspect-based summarization. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4381–4401, Toronto, Canada. Association for Computational Linguistics.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020a. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org.

Lining Zhang, Simon Mille, Yufang Hou, Daniel Deutsch, Elizabeth Clark, Yixin Liu, Saad Mahamood, Sebastian Gehrmann, Miruna Clinciu, Khyathi Raghavi Chandu, and João Sedoc. 2023a. A Needle in a Haystack: An Analysis of High-Agreement Workers on MTurk for Summarization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14944–14982, Toronto, Canada. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020b. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori B. Hashimoto. 2023b. Benchmarking Large Language Models for News Summarization. ArXiv:2301.13848 [cs].

A Appendix

We provide our full annotation guideline for the dataset creation task in Section A.1 and dataset validation task in Section A.2. Table 10 and 11 list models and software used in our study with external URLs, respectively.

A.1 Annotation Guideline for Multi-aspect Summarization Dataset

A.1.1 Background

While there are a number of datasets for the "single-document summarization" task where one source document is coupled with one generic summary, there is no dataset created from scratch for "multi-aspect summarization" where there are multiple summaries for a document focusing on different aspects.

In this annotation task, we aim to construct a dataset where there are three one-sentence summaries about **challenge**, **approach**, and **outcome** for one research article.

A.1.2 Task Description

The goal of this annotation task is to construct a dataset for multi-aspect summarization systems where one source document is coupled with summaries that each focus on different aspects in the source document. We work with documents from the scholarly domain, i.e., our source documents are academic research papers. Specifically, we annotate papers that have been published in major NLP conferences (ACL, NAACL, EMNLP, EACL, AACL) and the aspects of interests are **CHALLENGE**, **APPROACH** and **OUTCOME**.

For defining each aspect, we take a subset of the categories proposed in Fisas et al. (2015) and make small wording modifications, shown as follows:

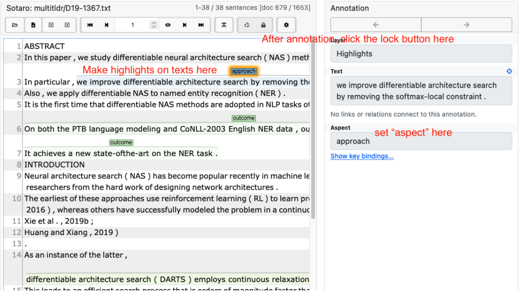
- **CHALLENGE**: The current situation faced by the researcher; it will normally include a Problem Statement, the Motivation, a Hypothesis and/or a Goal.
- **APPROACH**: How they intend to carry out the investigation, comments on a theoretical model or framework.
- **OUTCOME**: Overall conclusion that should reject or support the research hypothesis.

A.1.3 Data

We sample 1000 papers from the ACL anthology and use them as target documents for our annotation. All of the selected papers are from ACL, NAACL, EMNLP, EACL, and AACL.

A.1.4 Annotation Platform

We use INCEpTION (Klie et al., 2018) to perform our annotation task.



Sotaro: multitldr/D19-1367.txt

1-38 / 38 sentences [doc 679 / 1653]

After annotation, click the lock button here

Make highlights on texts here

ABSTRACT
In this paper , we study differentiable neural architecture search (NAS) meth
In particular , we improve differentiable architecture search by removing th
Also , we apply differentiable NAS to named entity recognition (NER) .
It is the first time that differentiable NAS methods are adopted in NLP tasks of
On both the PTB language modeling and CoNLL-2003 English NER data , ou
It achieves a new state-ofthe-art on the NER task .
INTRODUCTION
Neural architecture search (NAS) has become popular recently in machine le researchers from the hard work of designing network architectures .
The earliest of these approaches use reinforcement learning (RL) to learn pr 2016) , whereas others have successfully modeled the problem in a continuc
Xie et al . , 2019b ;
Huang and Xiang , 2019)
.
As an instance of the latter ,
differentiable architecture search (DARTS) employs continuous relaxation

Annotation

Layer: Highlights

Text: we improve differentiable architecture search by removing the softmax-local constraint .

No links or relations connect to this annotation.

Aspect: approach

set “aspect” here

Show key bindings...

Figure 5: Screenshot of annotation procedure with INCEpTION.

A.1.5 Annotation Procedure

Step 0: Open a document Open a new document in INCEpTION to start the annotation.

Step 1: Understand the document Skim through the title (in the spreadsheet), abstract, introduction and conclusion of the document to identify the "main contribution" of the paper. If you find too many PDF parsing errors at this stage, skip the document by just moving on to the next one.

There may be several pages for all the lines, we suggest to increase the "Page size" to 1000 so you can see all the lines in one pages. To do that, with one document opened, you click the "gear" button on top, and increase the number in the "Page size" form.

And we also recommend to change the color scheme for highlights to "dynamic pastelle" in the same configuration page to ease distinguishing highlights for different aspects.

Step 2: Read and highlight relevant text sequences Read again but sentence by sentence the abstract, introduction and conclusion, and highlight text passages (not necessarily an entire sentence) that should be included in final summary using the INCEpTION highlighting tool. After selecting a substring for highlighting, use the ‘Aspect‘ section in the right sidebar to assign the corresponding aspect type to the highlighted text passage. Type ‘c‘ for CHALLENGE, ‘a‘ for APPROACH and ‘o‘ for OUTCOME. Highlight the same information multiple times if it is relevant and appears multiple times with/without different wording.

Some points

- Make sure the highlighted sentences are relevant to the "main contribution" identified in the Step 1
- You will later fuse highlighted sentences into

a summary and not all the information in the sentences need to be included

Step 3: Write summaries Review all the highlights aggregated in the sidebar for each aspect, and create one-sentence summary for each of them. Pack as much as information possible in the word limitation ($<= 25$ words) for each summary. Remove the highlighting if the information is not included in the final summary. Save the summaries in the corresponding row in the spreadsheet.

Each summary needs to fulfill the following constraints:

- Each summary contains only one sentence, and has less than 25 words
- Each summary cannot reference other summaries on different aspects (an example below)
- All the information can be found in the considered sections

A.1.6 FAQ

Is the mention of a newly created dataset APPROACH or OUTCOME? If authors discuss findings based on experiments using the dataset, highlight it as an OUTCOME. If they discuss how and why the dataset has been created, highlight it as an APPROACH.

Is it possible for one sentence to have both aspects? Yes, for instance, there may be a sentence which mentions, CHALLENGE and APPROACH. In this case, highlight the text separately for both aspects in the same sentence, possibly with the some overlaps, and make sure to provide a correct ‘tag‘ for each in the sidebar.

Is "Concluding remarks" same as "Conclusion"? Yes, we consider them to be the same.

A.2 Annotation Guideline for Validating Summaries

A.2.1 Background

The aim of this work is to create multi-document summarization datasets with highlight annotations. The resulting dataset will have gold standards that can be used for development and evaluation for both abstractive and extractive summarization systems.

Since, we build our dataset using research papers from ACL conferences, the experts with the domain knowledge are required to validate the quality of this new dataset.

A.2.2 Dataset

Each data sample is composed of two kinds of annotations.

- **Highlight** for relevant sentences - One-sentence **Summary** which merges the highlighted sentences

Since, this dataset is **multi**-document summarization dataset, we have both kinds of annotations for 3 aspects, namely,

- **Challenge**: The current situation faced by the researcher; it will normally include a Problem Statement, the Motivation, a Hypothesis and/or a Goal.
- **Approach**: How they intend to carry out the investigation, comments on a theoretical model or framework.
- **Outcome**: Overall conclusion that should reject or support the research hypothesis.

Overall, for one research paper, there are 3 sets of highlights and 3 one-sentence summaries.

A.2.3 Task Description

Your task for this annotation project is to validate the quality of summaries according to the following three criteria:

- **Relevance**: measures how well the summary captures the key points of the source document. If you find multiple key points in the source document, check if the most important one is included in the summary. The summaries may not contain all the key points due to the length limitation (less than 25 words per summary).
- **Consistency (Faithfulness)**: measures if the facts in the summary are consistent with the facts in the source document. See the highlighted sentences of the corresponding aspect in the source document and check whether the summary does reproduce all facts accurately and does not make up untrue information.
- **Fluency**: measures the quality of the summary as a sentence. Check if they are well-written and grammatically correct.

A.2.4 Annotation Procedure

In our annotation task, we only use the following sections of a paper instead of the entire document.

- Title
- Abstract

- Introduction
- Conclusion (if does not exist, we use Discussion)

Read only these parts of the paper when working on the annotation task described in the following steps.

Step 0: Open a document Open the corresponding URL to the highlighted PDF file from the spreadsheet, and check if the file matches the item that you are evaluating in spreadsheet.

Step 1: Evaluate Relevance Read the document, and identify the key points regarding to the aspect of **Challenge** in the paper. If there are multiple, consider the most important one.

Then, read the summary about the **Challenge** in the spreadsheet, and check if the key point identified in the source document is mentioned in the summary as well.

Give the scores from 1 to 5 as the following:

- The summary does not include any information
- The summary contains some information but it is not relevant
- The summary contains few points that but they do not convey the main concept of the paper
- The summary contains key point(s) but the most important one is missing
- The summary contains the most important key point correctly

Then, repeat this for other two aspects, **Approach** and **Outcome**.

Step 2: Evaluate Consistency (Faithfulness) Read the summary about the **Challenge**, and check if the facts mentioned in this summary actually appears in the source document as well. In this step, you do not have to read all the source documents but only the sentences highlighted in the color which corresponds to the **Challenge** aspect.

Give the scores from 1 to 5 as the following:

- The summary contains a number of critical untrue information which can critically mislead readers
- The summary contains few critical untrue information which can mislead readers
- The summary contains some minor untrue information
- The summary does not contain any untrue information but readers make wrong interpretations
- The summary does not contain any untrue information according to the paper and there is no space for readers to misunderstand

Then, repeat this for other two aspects, **Approach** and **Outcome**.

Step 3: Evaluate Fluency Read the summary about the **Challenge**, and check if it is well-written and grammatically correct. In this step, you do not have to read the source document at all.

Give the scores from 1 to 5 as the following:

- The summary contains a number of grammatical errors which make it unreadable
- The summary contains a few critical grammatical errors which lead to misunderstandings
- The summary contains a few minor grammatical errors which can lead to misunderstandings
- The summary does contain errors but they would not lead to any misunderstandings
- The summary does not contain any errors and there is no space for readers to misunderstand

Then, repeat this for the other two aspects, **Approach** and **Outcome**.

Model	# Params	Licence	URL
BART$_{\text{BASE}}$	139M	Apache 2.0	https://huggingface.co/facebook/bart-base
BART$_{\text{LARGE}}$	406M	Apache 2.0	https://huggingface.co/facebook/bart-large
T5$_{\text{BASE}}$	223M	Apache 2.0	https://huggingface.co/t5-base
T5$_{\text{LARGE}}$	738M	Apache 2.0	https://huggingface.co/t5-large
ST5$_{\text{BASE}}$	110M	Apache 2.0	https://huggingface.co/sentence-transformers/sentence-t5-base
ST5$_{\text{LARGE}}$	335M	Apache 2.0	https://huggingface.co/sentence-transformers/sentence-t5-large
ST5$_{\text{XL}}$	1.24B	Apache 2.0	https://huggingface.co/sentence-transformers/sentence-t5-xl
Llama 2$_{\text{7B}}$	7B	LLAMA 2 License	https://huggingface.co/meta-llama/Llama-2-7b-hf
Llama 2 Chat$_{\text{7B}}$	7B	LLAMA 2 License	https://huggingface.co/meta-llama/Llama-2-7b-chat-hf

Table 10: A list of models with external URLs used in our study.

Package	Licence	URL
Grobid (Lope, 2008–2023)	Apache 2.0	https://github.com/kermitt2/grobid
INCEpTION (Klie et al., 2018)	Apache 2.0	https://github.com/inception-project/inception
PyTorch (Paszke et al., 2019)	BSD-style	https://github.com/pytorch/pytorch
Transformers (Wolf et al., 2020)	Apache 2.0	https://github.com/huggingface/transformers
Lightning	Apache 2.0	https://github.com/Lightning-AI/pytorch-lightning
scikit-learn (Pedregosa et al., 2011)	BSD 3-Clause	https://github.com/scikit-learn/scikit-learn
Spacy (Honnibal et al., 2020)	MIT	https://github.com/explosion/spaCy/
SentenceTransformers (Reimers and Gurevych, 2019)	Apache 2.0	https://github.com/UKPLab/sentence-transformers
SacreRouge (Deutsch and Roth, 2020)	Apache 2.0	https://github.com/danieldeutsch/sacrerouge

Table 11: A list of software and libraries with external URLs used in our study.

			Challenge				**Approach**				**Outcome**			
	Model	**Label type**	**R-1**	**R-2**	**R-L**	**BS**	**R-1**	**R-2**	**R-L**	**BS**	**R-1**	**R-2**	**R-L**	**BS**
BART$_{\text{BASE}}$	**ST5$_{\text{BASE}}$**	**Gold**	18.47	2.43	13.42	0.617	43.31	19.88	35.61	0.730	41.39	19.70	34.86	0.723
		Heuristic	17.90	2.44	13.25	0.614	44.55	20.60	37.23	0.735	40.43	19.12	33.53	0.723
	ST5$_{\text{LARGE}}$	**Gold**	19.02	2.57	13.86	0.620	44.51	21.30	37.71	0.739	39.47	18.40	33.26	0.719
		Heuristic	18.31	2.57	13.41	0.615	43.35	20.39	36.98	0.734	39.98	18.33	32.77	0.716
	ST5$_{\text{XL}}$	**Gold**	19.00	2.31	13.76	0.622	45.12	21.17	37.76	0.739	39.82	18.83	34.17	0.722
		Heuristic	18.71	2.51	13.95	0.615	44.90	20.79	37.91	0.737	39.03	18.45	32.82	0.711
BART$_{\text{LARGE}}$	**ST5$_{\text{BASE}}$**	**Gold**	18.35	2.15	13.03	0.611	44.55	21.44	37.67	0.732	40.30	19.28	33.97	0.719
		Heuristic	19.02	2.36	13.53	0.607	44.59	20.86	37.97	0.735	39.48	18.38	33.68	0.716
	ST5$_{\text{LARGE}}$	**Gold**	19.61	2.51	13.95	0.614	44.21	20.69	37.11	0.731	38.14	17.70	31.98	0.708
		Heuristic	18.64	2.39	13.50	0.610	44.20	21.52	38.05	0.737	40.18	19.47	34.10	0.720
	ST5$_{\text{XL}}$	**Gold**	12.77	1.59	9.43	0.597	43.88	20.48	37.09	0.677	39.18	18.45	33.43	0.665
		Heuristic	19.12	2.33	13.31	0.607	43.93	20.93	37.14	0.734	38.20	17.32	32.19	0.712
T5$_{\text{BASE}}$	**ST5$_{\text{BASE}}$**	**Gold**	18.93	2.45	13.56	0.610	44.82	22.40	38.36	0.732	42.25	21.72	34.98	0.721
		Heuristic	19.48	2.86	14.18	0.607	45.64	23.76	39.73	0.740	41.19	21.03	34.40	0.716
	ST5$_{\text{LARGE}}$	**Gold**	18.47	2.35	13.30	0.609	45.10	22.40	38.76	0.735	42.10	21.28	34.56	0.722
		Heuristic	18.56	2.45	13.56	0.606	45.47	23.06	39.27	0.739	40.36	20.31	33.66	0.707
	ST5$_{\text{XL}}$	**Gold**	18.19	2.31	12.99	0.592	45.97	23.32	39.80	0.668	40.27	19.23	33.30	0.639
		Heuristic	18.31	2.28	13.08	0.600	44.59	22.79	38.62	0.736	40.36	20.31	33.66	0.707
T5$_{\text{LARGE}}$	**ST5$_{\text{BASE}}$**	**Gold**	19.18	2.91	14.04	0.609	45.93	23.21	39.18	0.736	42.88	22.30	35.89	0.726
		Heuristic	19.35	3.08	14.19	0.611	46.33	23.80	39.98	0.742	42.63	22.36	35.95	0.725
	ST5$_{\text{LARGE}}$	**Gold**	18.80	2.96	13.59	0.612	45.51	22.74	39.24	0.738	41.76	21.52	35.71	0.724
		Heuristic	19.06	3.06	13.83	0.611	44.83	22.10	38.75	0.734	41.50	20.90	35.20	0.721
	ST5$_{\text{XL}}$	**Gold**	19.24	2.76	13.68	0.591	46.30	23.56	40.22	0.666	42.32	21.55	35.94	0.652
		Heuristic	19.04	2.86	13.76	0.608	45.79	22.79	39.40	0.739	41.50	21.26	35.44	0.716

Table 12: Results of extractive models trained on silver and gold data in the extract-then-abstract approach.