

Discourse Understanding and Factual Consistency in Abstractive Summarization

Saadia Gabriel[♦] Antoine Bosselut[♦] Jeff Da[◇] Ari Holtzman[♦]
Jan Buys[♡] Kyle Lo[◇] Asli Celikyilmaz[♦] Yejin Choi^{♦◇}

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

[♦]Microsoft Research [◇]Allen Institute for Artificial Intelligence

[♦]Stanford University [♡]University of Cape Town

{skgabrie, ahai, yejin}@cs.washington.edu, {jeffd, kylel}@allenai.org, antoineb@cs.stanford.edu, jbuys@cs.uct.ac.za, aslicel@microsoft.com

Abstract

We introduce a general framework for abstractive summarization with factual consistency and distinct modeling of the narrative flow in an output summary. Our work addresses current limitations of models for abstractive summarization that often hallucinate information or generate summaries with coherence issues.

To generate abstractive summaries with *factual consistency* and *narrative flow*, we propose *Cooperative Generator – Discriminator Networks* (Co-opNet), a novel transformer-based framework where a generator works with a discriminator architecture to compose coherent long-form summaries. We explore four different discriminator objectives which each capture a different aspect of coherence, including whether salient spans of generated abstracts are hallucinated or appear in the input context, and the likelihood of sentence adjacency in generated abstracts.

We measure the ability of Co-opNet to learn these objectives with arXiv scientific papers, using the abstracts as a proxy for gold long-form scientific article summaries. Empirical results from automatic and human evaluations demonstrate that Co-opNet learns to summarize with considerably improved global coherence compared to competitive baselines.

1 Introduction

Generating summaries with coherent discourse structure and domain knowledge awareness poses a challenge for current methods in summarization. Generative models can commonly produce high quality text (Figure 1), but fail to understand finer-grained details of coherence such as the structure and flow of a narrative. In addition, they often generate factually incorrect content. Prior work on factuality in abstractive summarization has found that current models can hallucinate information more

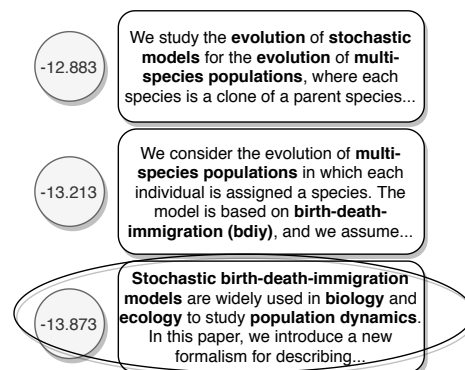


Figure 1: Generated abstracts for a biology article (from the **Bio** subset of our arXiv dataset). Abstracts are ranked from most (top) to least likely (bottom) using the generator model. Abstracts with better narrative structure and domain-specific content (such as the circled abstract) are often out-ranked in terms of likelihood by abstracts with factual errors and less structure.

than 70% of the time when generating summaries of news articles (Maynez et al., 2020).

To address these issues, we focus our study on generating abstractive summaries with *factuality* and *narrative flow*. Given an input document, the goal is to generate a paragraph-length abstractive summary with proper discourse structure that contains factually correct claims. Our study builds on and extends previous work that focuses on either *extractive document-level* summarization (Nenkova and McKeown, 2012; Allahyari et al., 2017) or *abstractive sentence-level* summarization (Rush et al., 2015; Grusky et al., 2019; Narayan et al., 2018).

In pursuit of this goal, we introduce *Cooperative Generator-Discriminator Networks* (Co-opNet), a framework for abstractive summarization that considers subtle aspects of fact-checking and discourse necessary for coherent text generation. In this framework, the generator, a transformer language model fine-tuned for abstractive summarization, proposes a pool of candidate summaries (§2). The

discriminator, also transformer-based, scores the factuality or discourse quality of candidate summaries using one of four different objectives: the overlap between a scientific article introduction and predicted fact-checking evidence spans in generated summaries, the ordering of predicted discourse roles, the coverage of predicted discourse roles, or the likelihood of adjacency between generated sentences (§3). The best summary is chosen cooperatively by combining the generator and discriminator scores (§4).

Most previous works on *abstractive document-level* summarization have difficulty in directly modeling or evaluating *narrative flow* and *factuality* in generated summaries. This weakness is largely due to the inherent limitations of existing datasets, such as the CNN/DailyMail dataset (Hermann et al., 2015). The reference summaries available in these commonly used resources are mainly headlines of news articles or stories. As a result, they are often sets of disconnected sentences that are highly extractive, leading to models that are also extractive (Hoang et al., 2019), rather than abstractive.

In order to address these data challenges, we test our summarization model on a set of arXiv scientific papers. Scientific abstracts are ideal for modeling narrative flow as they are structured with highly coherent discourse flow. They also maintain implicit *abstractive* alignments with respect to the introduction of the article – in contrast to the tight, *extractive* alignments of current models. Scientific article summarization is also a task where factuality is more well-defined than in other domains like story summarization which leave more room for interpretation.

Comprehensive empirical results considering both automatic and human evaluations demonstrate that Co-opNet learns to summarize scientific articles from three domains with considerably improved global coherence compared to competitive baselines (§6). We also demonstrate that the framework is generalizable to multiple coherence objectives, and effective at generating scientific abstracts that are more factually consistent.

2 Generator Networks

We use the transformer architecture of Radford et al. (2019) as our generator’s architecture. Following the work of Liu et al. (2018), we adapt a language model to the task of abstractive summarization by concatenating the ar-

ticle a , a delimiter token [SEP], the summary s , and an end token [END] into one input vector $X = (a_1, \dots, a_{|a|}, [\text{SEP}], s_1, \dots, s_{|s|}, [\text{END}])$, where $|a|$ is the length of the gold article and $|s|$ is the length of the gold summary.

At each time step i , the model produces an output probability distribution over the vocabulary for the next token w_i given all previous output tokens $w_{<i}$. For any arbitrary token w_j preceding w_i , the per-layer representation of that token is computed in the following way:

$$\mathbf{h}_j^0 = \mathbf{W}_e(w_j) + \mathbf{p}_j \quad (1)$$

$$\mathbf{h}_j^l = \text{block}(\{\mathbf{h}\}_{<j}^{l-1}) \quad (2)$$

where *block* refers to each transformer block composed of multi-headed attention, a feedforward network and layer normalization, \mathbf{W}_e is a word embedding matrix, \mathbf{p}_j is the position embedding, \mathbf{h}_j^0 is the initial representation, $\{\mathbf{h}\}_j^l$ is the block output for an arbitrary layer l , and $\{\mathbf{h}\}_{<j}^{l-1}$ is the set of all block outputs from the preceding layer for positions up to j . Finally, for the current position i in the sequence, we compute a distribution over the output vocabulary as follows:

$$P(w_i | w_0, \dots, w_{i-1}) = \text{softmax}(\mathbf{h}_{i-1}^L \mathbf{W}_e) \quad (3)$$

where \mathbf{W}_e is the same embedding matrix as in Equation 1 and \mathbf{h}_{i-1}^L is the final layer transformer block output.

3 Discriminator Networks

Because summarization models are prone to narrative flow and factual consistency issues (Kryściński et al., 2020; Xu et al., 2020), we use a discriminator to score generated summaries for discourse and factuality properties. Due to the challenge of explicitly defining discourse and factuality properties as scores, these properties are approximated using parameterized scoring functions.

These scoring functions determine if generated text demonstrates discourse and factuality properties in three ways: (1) predicting the discourse role of sentences within a full summary, (2) predicting the likelihood of adjacency given a sentence pair, and (3) measuring the presence of salient facts in the generated summary from the original input context. While our discriminators focus on these three properties, we note that this framework is generalizable and could be extended to include other

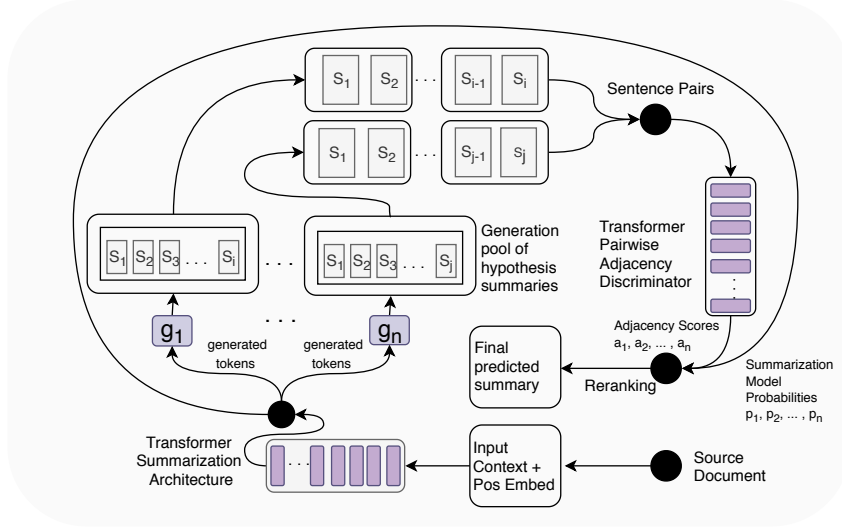


Figure 2: Model architecture for adjacency reranking variation of Co-opNet

discriminator models that encourage different communicative norms associated with high-quality language generation.

3.1 Discourse

We explore different discriminator architectures as additional discourse scoring functions during the generator’s decoding process. For these discriminators, we generally score discourse in two ways. First, we use inferred sentence-level scientific abstract discourse role labels¹ defined by Cohan et al. (2019) and predict them using a sequence classifier² based on SciBERT (Beltagy et al., 2019). Using these predictions, we score the discourse properties of the abstract relative to their coverage (§3.1.1) or ordering (§3.1.2). Second, we learn a function that can score the likelihood that sentences within generated abstracts should be adjacent to one another (§3.1.3).

3.1.1 Coverage

We measure the completeness of the narrative structure within a scientific abstract by defining the following coverage score:

$$\mathcal{L}_{cov} = \log(D_{abs}/D_{all}), \quad (4)$$

where D_{abs} is the number of unique discourse roles appearing in an abstract and D_{all} is the total number of possible discourse roles. This objective al-

¹The labels are {BACKGROUND, METHOD, OBJECTIVE, RESULT, OTHER}.

²See (Cohan et al., 2019) for model and training details.

S_{i-1}	S_i
BACKGROUND	BACKGROUND
BACKGROUND \vee METHOD \vee OBJECTIVE	METHOD
BACKGROUND \vee OBJECTIVE \vee METHOD	OBJECTIVE
OBJECTIVE \vee METHOD \vee OTHER	RESULT

Table 1: Discourse Role Ordering

lows us to penalize abstracts that are missing discourse roles. For example, an abstract that fails to mention anything about the results of the study would be penalized.

3.1.2 Ordering

We also score the order in which discourse labels appear in generated abstracts. In Table 1, we hard-code valid orderings of discourse labels for generated sentences based on each of the abstract discourse roles of Cohan et al. (2019). If the ordering for two adjacent sentences in the abstract $O(s_{i-1}, s_i)$ is valid, the score for the ordering is 1 (-1 otherwise). We sum the scores for all the orderings within a particular abstract and normalize between 0 and 1 (as described by f_n):³

$$\mathcal{L}_{order} = \log(f_n(\sum_{i=1}^S O(s_{i-1}, s_i))) \quad (5)$$

We also impose a rule for s_1 =‘BACKGROUND’ and a rule for s_S =‘RESULT’ to encourage more natural orderings.

³See the Appendix for a more detailed description of the f_n function.

3.1.3 Adjacency Classification

To model the likelihood of adjacency between two sentences s_u and s_v , we first compute a hidden representation of the sentence pair using SciBERT (Beltagy et al., 2019). The encoder input is the concatenation of the sentences: $\mathbf{s} = [\text{CLS}] + s_u + [\text{SEP}] + s_v + [\text{SEP}]$, where $[\text{CLS}]$ is a special token associated with the task and $[\text{SEP}]$ is a sentence delimiter token. Each word in the sequence is encoded by a word embedding w_i and positional embedding p_i and passed through the SciBERT model to yield \mathbf{h}_{cls} , the output state at the position of the $[\text{CLS}]$ token. We then obtain the probability of adjacency between the sentences by a linear projection of \mathbf{h}_{cls} followed by a sigmoid activation:

$$P_{adj}(\mathbf{s}) = \sigma(\mathbf{w}_{disc}^\top \mathbf{h}_{cls}) \quad (6)$$

We define the training objective for the adjacency discriminator to minimize the negative log likelihood of predicting whether two sentences are adjacent or not:

$$\mathcal{L}_{disc} = -\left(\delta_{adj}(\mathbf{s}) \cdot \log(P_{adj}(\mathbf{s})) + (1 - \delta_{adj}(\mathbf{s})) \cdot \log(1 - P_{adj}(\mathbf{s}))\right), \quad (7)$$

where $\delta_{adj}(\mathbf{s})$ is an indicator function for whether the two sentences in \mathbf{s} are adjacent. We note that while the discourse discriminators mainly focus on narrative structure, they may also capture context-aware aspects of factuality and content selection.

3.2 Factuality and Faithfulness

To measure factuality of generated summaries, we predict which tokens in the summary are likely to belong to a fact-checking evidence span (i.e., a span of the text used to prove a scientific claim using a finetuned BERT token classification model.⁴ Recent work has shown that inspecting attention weights alone is not necessarily a reliable metric for determining saliency of particular aspects in the input context to the output of neural models (Serrano and Smith, 2019). The saliency weights representing the likelihood of tokens belonging to evidence spans provides us with a more explicit representation of factual importance.

We obtain proxy saliency labels for the importance of a particular token t appearing in an abstract using a BERT model trained on evidence

⁴See Appendix A.4 for details of token classification model.

Topic	Spans
NLP	existing semantic schema, annotation effort, music knowledge representation, siri assistant
BIO	biological system, ptotic, cybernetics entropy, shannon established fundamental limits

Table 2: Salient spans extracted using factuality discriminator.

spans annotated for scientific fact-checking (Wadden et al., 2020). Specifically, if t is not a stopword and $t \in E$, where E is an evidence span used to check a scientific claim, then we assign a label of 1 to t . Otherwise, the label for t is 0. Examples of extracted spans are given in table 2.

We compare the predicted evidence spans against information presented in the original introduction to capture the degree to which generative models are hallucinating information.

Factuality Objective At inference time, we compare the extracted salient spans, $F(g)$, of the generated summary g against the set of all ngrams in the article input context, $N(a)$, measuring the degree to which salient spans are hallucinated:

$$\mathcal{L}_{fact} = \log\left(\frac{|\{f | f \in F(g), f \in N(a)\}|}{|F(g)|}\right) \quad (8)$$

4 Reranking with Discourse and Factuality Experts

To incorporate the discriminator objective into our summarization framework, we first generate a pool of candidate summaries from the base summarization model (§2) using any decoding strategy (e.g., beam search or top- k sampling). Then, the discriminator is used to re-rank these candidates in conjunction with the original token-level generator scores. For example, in the case of the adjacency discriminator, we maximize the generator token-level probability of a candidate summary g , and the average of adjacency scores for the set of sentences composing g (denoted $S(g)$) – i.e., the probability of each sentence s_u being adjacent to the previous sentence s_{u-1} in $S(g)$:

$$\begin{aligned} \text{score}(g) = & \lambda_{gen} \frac{1}{|g|} \sum_{i=1}^{|g|} \log P(w_i | w_1, \dots, w_{i-1}) \\ & + \lambda_{disc} \frac{1}{|S(g)| - 1} \sum_{u=2}^{|S(g)|} \log P_{adj}(s_u, s_{u-1}), \end{aligned} \quad (9)$$

Split	CS	BIO	AAN
Train	44900	4104	10106
Validation	5622	555	892
Test	5670	522	892

Table 3: Domain subset sizes

where λ_{gen} and λ_{disc} are hyper-parameters controlling the contribution of the generator and adjacency discriminator to the final predicted summary. The same procedure is followed for the other discourse and factuality objectives, replacing $P_{adj}(s_u, s_{u-1})$ with the scores from these discriminators.

5 Data

5.1 Datasets

Since the focus of this work is on generating summaries with more coherent narrative flow and greater factual consistency, we concentrate on datasets requiring discourse structure to generate good summaries. Particular attributes of the discourse structure of these datasets include:

- Length of summaries \rightarrow Are the summaries long enough to clearly show narrative flow properties and factual correctness?
- Abtractiveness of gold summaries \rightarrow Do the summaries exhibit particular sentence-level flow, or are the summary sentences extracted highlights from the context?

ArXiv We crawled over 700K samples (472K abstracts) from scientific articles on `arxiv.org`. In our experiments we primarily focus on the CS⁵ and Bio⁶ domain subsets. The task we define is to generate an abstract given a introduction, which presents a challenge to existing summarization models. This task also requires models to learn relevant domain knowledge for the scientific domain of interest and recognize common discourse structure for papers written in that domain.

AAN Additionally, we include an existing dataset of scientific articles that focuses on papers in the NLP computer science domain. This dataset consists of a 12k paper subset from the ACL Anthology Network (AAN; Radev et al., 2009) with extracted introduction and abstract pairs.

⁵<https://arxiv.org/corr>

⁶<https://arxiv.org/archive/q-bio>

Scientific abstracts in ArXiv and AAN have properties that are missing from existing summarization datasets based on Newswire data. For example, XSum (Narayan et al., 2018) and Newsroom (Grusky et al., 2019) summaries are generally too short to exhibit cross-sentence narrative flow. Meanwhile, CNN/DailyMail (Hermann et al., 2015) summaries are acquired by concatenating extracted highlights, which can be unrelated. Conversely, ArXiv and AAN abstracts are long enough to have multiple sentences,⁷ and generally exhibit strong discourse patterns typical to scientific writing, making them ideal corpora for assessing discourse understanding in abstractive summarization. Table 3 provides details of dataset splits.

6 Experimental Setup

Our implementation is based on the Huggingface implementation⁸ of the BERT (Devlin et al., 2019) and GPT-2 language models (Radford et al., 2019).

Generator We perform WordPiece tokenization for the input context and output summaries. Because of the fixed input size of the transformer language model, the input context is truncated to a maximum of 800 tokens, and summaries are truncated to a maximum of 200 tokens. We use a learning rate of 2e-5 and a batch size of 16 to finetune the generator. We train the base summarization transformer model for 12 epochs. All experiments are run on either a Titan-X or Quadro RTX 8000 GPU. Training time for the AAN and ArXiv Bio datasets is about 30 minutes per epoch. Training time for the ArXiv CS dataset is 2.5 hours per epoch. In our experiments we use top- k sampling with $k=4$ (Fan et al., 2018) to generate candidate summaries for each model.

Discriminator At training time we use a maximum sentence length of 200 tokens to accommodate the fixed input size of BERT (512 tokens), reduce inference time, and discourage the model from generating abnormally long run-on sentences that indicate the presence of coherence issues.⁹

For the adjacency discourse models, we fine-tune the discriminator using a learning rate of 2e-5, a linear warmup learning rate schedule, and a batch size

⁷See Appendix A.6 for comparison of datasets.

⁸<https://github.com/huggingface/transformers>

⁹See the original papers for details of training the SciFact and abstract discourse models.

Model	AAN			CS			Bio		
	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L
Lede-3	27.12	6.62	23.88	28.22	7.06	16.22	27.60	5.70	24.21
LexRank	36.03	10.14	31.37	36.53	10.41	32.09	35.32	8.84	30.76
LSTM	27.80	5.57	18.02	22.74	4.56	20.64	10.73	0.49	9.94
PGen	39.85	12.83	23.24	36.68	11.74	32.55	23.74	4.48	21.65
Generator (Our work)	41.31	12.97	37.05	38.01	10.95	34.46	34.86	8.45	31.38
Co-opNet (Our work)	41.67	12.65	37.23	38.57	10.81	35.11	35.86	8.41	32.56

Table 4: Automatic Evaluation of generative architectures and Co-opNet. For AAN, we provide results using the Factuality discriminator. For CS and Bio, we provide results using the Coverage discriminator.

Model	BERTScore	SciBERTScore
PGen	57.86	59.13
Generator	61.71	62.80
Co-opNet (Adj)	61.87	63.10
Co-opNet (Fact)	62.09	63.21

Table 5: BERTScore results on AAN subset (F1)

of 32. All adjacency discourse discriminator models are fine-tuned for 2 epochs on a Titan-X GPU. The adjacency discriminator models are adapted from the Huggingface implementation of the BERT next sentence prediction classifier. We initialize the 12-layer BERT-base discriminator model with the pretrained weights of the SciBERT-uncased model, which was originally trained on 1.14 million scientific papers (Beltagy et al., 2019). Two discriminators are trained: one is fine-tuned on AAN for decoding both ArXiv CS and AAN, while the other discriminator is fine-tuned on ArXiv Bio and used exclusively for decoding that subset. We weigh the generation and discriminator models equally when decoding by setting $\lambda_{gen}=\lambda_{disc}=.5$. Additional implementation details are provided in Appendices A.3 and A.4.¹⁰

7 Experiments

We compare against extractive approaches using the Lede-3 and LexRank (Erkan and Radev, 2004) baselines. We also compare against two abstractive approaches: a 2-layer bi-LSTM sequence-to-sequence model with attention (LSTM), and a pointer-generator model (PGen; See et al., 2017). Training details of the supervised baselines can be found in the Appendix A.2. In addition, we compare to a subset of our approach that only uses the generator to produce summaries, rather than the full framework.

¹⁰Our code/data is released here: <https://github.com/skgabriel/coopnet>.

7.1 Automatic Evaluation

Following previous work on summarization, we use the ROUGE metric (Lin, 2004) for automatic evaluation of generative models and Co-opNet. Specifically, we report ROUGE-1, ROUGE-2 and ROUGE-L F1 scores. To capture similarity in contextual meaning, we look at BERTScore F1 (Zhang et al., 2020a), which has been shown to more closely correlate with human judgements than other generation metrics.

Results on the AAN, CS and Bio subsets of ArXiv are shown in Table 4. Co-opNet outperforms all baselines on ROUGE-1 and ROUGE-L by a consistent margin. Notably, Co-opNet’s performance is superior to the generator-only model, illustrating the importance of the discriminators for generating more coherent summaries. Interestingly, on the more domain-specific AAN subset, our model is over 12% better on ROUGE-L compared to the PGen baseline and 5.86% better than the best extractive model. Our model also outperforms the strongest baselines on BERTScore.

When we break down results for various Co-opNet architectures (see Table 6), we find that the factuality and discourse role discriminators lead to the best performance in terms of ROUGE scores with the adjacency discriminator achieving lower performance on ROUGE than the base generator. However, as shown by Table 5, the adjacency discriminator outperforms the base generator when we consider BERTScore, a more contextual evaluation metric, indicating that this generator-discriminator combination selects summaries that capture the same linguistic patterns and meaning as reference summaries without directly copying.

7.2 Human Evaluation

Since coherence of generated text is difficult to measure with automatic metrics (Kilickaya et al., 2017; Sun et al., 2019; Clark et al., 2019), we

Model	AAN			CS			Bio		
	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L
Coverage (Cov)	41.29	12.09	37.14	38.57	10.81	35.11	35.86	8.41	32.56
Order	41.20	12.20	37.11	38.50	10.87	35.10	35.66	8.46	32.39
Adjacency (Adj)	40.97	12.46	36.70	37.44	10.67	33.86	34.89	8.45	31.57
Factuality	41.67	12.65	37.23	38.23	11.03	34.64	35.46	8.41	31.96

Table 6: Comparison of different Co-opNet discriminators

PGen vs. Co-opNet-Adj			Generator vs. Co-opNet-Adj		
Criteria	PGen	Co-opNet	Criteria	Generator	Co-opNet
Abstractiveness	41.89	47.30	Abstractiveness	20.41	38.10
Coherence	42.57	50.00	Coherence	23.81	34.01
Factuality	39.86	45.95	Factuality	22.98	30.41
Overall	34.90	53.02	Overall	25.00	31.08

PGen vs. Co-opNet-Fact			Generator vs. Co-opNet-Fact		
Criteria	PGen	Co-opNet	Criteria	Generator	Co-opNet
Abstractiveness	51.02	39.46	Abstractiveness	27.33	35.33
Coherence	43.92	50.00	Coherence	30.87	32.21
Factuality	43.84	48.63	Factuality	27.52	32.21
Overall	43.54	50.34	Overall	30.87	32.21

Table 7: Human Evaluation of Co-opNet Architectures (% of judgements for each model)

conduct human evaluations to assess how the discriminator affects generation quality using pairwise model comparisons.

Setup We use four key criteria in all evaluations – abstractiveness, coherence, factuality and best overall quality, which we define as follows:

- Abstractiveness → Which abstract rewords information from the introduction instead of directly copying from the introduction?
- Coherence → Which abstract is more structured, and presents a complete and coherent story about the work done in the paper?
- Factuality → Which abstract is more factually consistent, presenting the same information that appears in the introduction and not producing hallucinated information?
- Overall → Which abstract is better overall?

We conduct human evaluations on Amazon Mechanical Turk (AMT) considering 4 different abstractive baseline model variants over 100 randomly sampled AAN test set examples. Given a gold introduction, AMT evaluators are asked to compare a corresponding abstract generated from Co-opNet against an abstract generated by a baseline or our generator model. To reduce bias, the

ordering of generated abstracts are randomized and evaluators are not told that abstracts are machine-generated.

Each abstract pair is judged by three unique annotators. For each criteria, we filter to 50 abstracts based on the amount of time AMT workers spent (≥ 20 seconds) and inter-annotator agreement (at least $\frac{2}{3}$ of annotators should agree on which abstract is best). We also prime annotators to consider subtler aspects of discourse coherence by providing examples that capture good or bad narrative flow without complete text degeneration.

We test the Co-opNet framework using both the factuality and adjacency discriminators, as these are the highest and lowest performing discriminator architectures in terms of automatic metrics on the AAN domain. We allow for ties, as Co-opNet and the generator baseline sometimes assign the highest probability to the same abstract, or generated abstracts in the candidate pool are high quality enough that there is little room for improvement.

Results We find that Co-opNet is preferred across all criteria for all comparisons, when we use the adjacency discriminator (see Table 7). When using the the factuality discriminator, Co-opNet is superior to baselines in all cases except when compared on abstractiveness to the PGen model.

In particular, human evaluators prefer Co-

Gold	We investigate mutual benefits between syntax and semantic roles using neural network models, by studying a parsing->SRL pipeline, a SRL->parsing pipeline, and a simple joint model by embedding sharing. The integration of syntactic and semantic features gives promising results in a Chinese Semantic Treebank...
PGen	In this paper, we propose a novel approach to learn <i>syntactic and semantic role labeling models to semantic role labeling (wsd)</i> . In the first neural network models induce non-linear feature features from word and <i>part-of-speech (pos) parsing</i> . We show that semantic features can be used to learn...
Generator	<i>Syntax-semantic relations</i> play a crucial role in natural language processing. In contrast, <i>semantic role labeling (srl)</i> models typically rely on parser output features to improve accuracy. In this work, we propose a joint <i>srl and syntactic parsing srl pipeline</i> using the <i>chinese treebank (qiu et al., 2016)</i> ...
Co-opNet (Adj)	In this paper, we explore the use of neural network models to jointly train <i>semantic role labelers</i> and <i>parsers</i> for <i>semantic role labeling (srl)</i> . We first propose a simple neural <i>srl</i> model that uses a neural <i>long shortterm memory (lstm)-based parser</i> to represent the output of an <i>srl</i> system...

Table 8: Example of gold and generated abstracts from baseline Pointer Networks + Coverage (See et al., 2017) (PGen) and two of our proposed models, Generator and Co-opNet, on the NLP scientific domain. Coherence issues and factual errors in generated abstracts are highlighted in *italics*. We highlight correct terminology and transitional phrases that contribute to coherent flow by properly delineating sections of abstracts in *bold* and *italics*.

opNet with the adjacency discriminator over baselines by over 8% on the coherence metric and 18.12% compared to PGen on overall quality. Notably, the adjacency discriminator encourages more abstractiveness in generated abstracts while still maintaining higher levels of factual consistency. We also find that Co-opNet with the factuality discriminator improves coherence and overall quality in addition to factuality. However, Co-opNet generations with the factuality discriminator were found to be more extractive than abstracts generated by PGen.

As shown in Table 8, generations selected by the adjacency discriminator more closely match the distribution of abstracts, while the generator sometimes favors copying from the introduction at the loss of narrative structure. For example, the generator will select a summary that opens with “*we present a method for jointly solving penn treebank style empty category (e.g. figure 1)*...”, while the adjacency discriminator selects a summary that opens with “*we present a method to jointly solve the problem of empty categories*...” and does not refer to a particular figure. Both summaries are faithful to the introduction, but the discriminator-selected summary makes more sense in the context of a paper abstract.

8 Related Work

Narrative Flow and Factuality Modeling coherent narrative flow remains a major challenge in the field of text generation, due to the need for accurate understanding of narrative structure (Chris-

tensen et al., 2013; Nikolov et al., 2018; Holtzman et al., 2018; Qin et al., 2019; Koncel-Kedziorski et al., 2019; Gabriel et al., 2021). Early approaches to incorporating structure include integration of explicit discourse markers into automatic summarization (Alonso i Alemany and Fuentes Fort, 2003). Recently proposed solutions include global-tracking of entities (Kiddon et al., 2016; Bosselut et al., 2018; Mei et al., 2016), as well as discourse-aware attention (Cohan et al., 2018). While there has been prior work on factual consistency (Cao et al., 2018; Gao et al., 2019; Kryściński et al., 2020; Zhang et al., 2020b), these works did not focus on scientific paper summarization.

Neural Abstractive Summarization In the past, abstractive summarization models (Rush et al., 2015; Gehrmann et al., 2018) have relied upon seq2seq encoder-decoder architectures (Sutskever et al., 2014; Narayan et al., 2018; Celikyilmaz et al., 2018). Transformer models have emerged as a promising architecture for text generation and summarization (Liu et al., 2018; Hoang et al., 2019; Khandelwal et al., 2019; Zhang et al., 2019). While our model builds upon this work, it is, to our knowledge, the first transformer summarization framework to explicitly model narrative flow and scientific fact-checking across domains.

9 Conclusion

In this work, we introduced *Cooperative Generator-Discriminator Networks*, a framework for more coherent natural language generation with transformer language models through the integration

of discriminators that encourage proper narrative flow and factual consistency. Through our analyses over scientific papers from ArXiv and AAN, we empirically showed that our framework selects generations that are more relevant and narratively coherent than previous approaches.

Acknowledgments

We thank the anonymous reviewers, as well as Lianhui (Karen) Qin, Jungo Kasai, Rik Koncel-Kedziorski, Elizabeth Clark, Dave Wadden and Rowan Zellers for helpful feedback. We also thank the annotators who contributed to the human evaluations in this work. This research was supported in part by NSF (IIS-1524371), DARPA CwC through ARO (W911NF-15-1-0543), and Samsung AI Research.

References

- Laura Alonso i Alemany and Maria Fuentes Fort. 2003. [Cohesion and coherence for automatic summarization](#). In *Student Research Workshop*.
- Mehdi Allahyari, Seyedamin Pouriyeh, Mehdi Assefi, Saeid Safaei, Elizabeth D. Trippe, Juan B. Gutierrez, and Krys Kochut. 2017. [Text summarization techniques: A brief survey](#). *International Journal of Advanced Computer Science and Applications*, 8(10).
- Iz Beltagy, Arman Cohan, and Kyle Lo. 2019. Scibert: Pretrained contextualized embeddings for scientific text. In *EMNLP*.
- Antoine Bosselut, Asli Celikyilmaz, Xiaodong He, Jianfeng Gao, Po-Sen Huang, and Yejin Choi. 2018. Discourse-aware neural rewards for coherent text generation. In *NAACL-HLT*.
- Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. 2018. Faithful to the original: Fact aware neural abstractive summarization. In *AAAI*.
- Asli Celikyilmaz, Antoine Bosselut, Xiaodong He, and Yejin Choi. 2018. Deep communicating agents for abstractive summarization. In *NAACL-HLT*.
- Janara Christensen, Mausam, Stephen Soderland, and Oren Etzioni. 2013. [Towards coherent multi-document summarization](#). In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1163–1173, Atlanta, Georgia. Association for Computational Linguistics.
- Elizabeth Clark, Asli Celikyilmaz, and Noah A. Smith. 2019. Sentence mover’s similarity: Automatic evaluation for multi-sentence texts. In *ACL*.
- Arman Cohan, Iz Beltagy, Daniel King, Bhavana Dalvi, and Daniel S. Weld. 2019. Pretrained language models for sequential sentence classification. In *EMNLP/IJCNLP*.
- 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*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.
- Günes Erkan and Dragomir R. Radev. 2004. Lexrank: Graph-based centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22:457–479.
- Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *ACL*.
- Saadia Gabriel, Chandra Bhagavatula, Vered Shwartz, Ronan Le Bras, Maxwell Forbes, and Yejin Choi. 2021. Paragraph-level commonsense transformers with recurrent memory. In *AAAI*.
- Shen Gao, Xiuying Chen, Piji Li, Zhangming Chan, Dongyan Zhao, and Rui Yan. 2019. How to write summaries with patterns? learning towards abstractive summarization through prototype editing. In *EMNLP*.
- Sebastian Gehrmann, Yuntian Deng, and Alexander Rush. 2018. [Bottom-up abstractive summarization](#). In *EMNLP*, pages 4098–4109, Brussels, Belgium.
- Max Grusky, Mor Naaman, and Yoav Artzi. 2019. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In *NAACL-HLT*.
- Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Neural Information Processing Systems*.
- Andrew Hoang, Antoine Bosselut, Asli Celikyilmaz, and Yejin Choi. 2019. Efficient adaptation of pre-trained transformers for abstractive summarization. *ArXiv*, abs/1906.00138.
- Ari Holtzman, Jan Buys, Maxwell Forbes, Antoine Bosselut, David Golub, and Yejin Choi. 2018. Learning to write with cooperative discriminators. In *ACL*.
- Urvashi Khandelwal, Kevin Clark, Dan Jurafsky, and Lukasz Kaiser. 2019. Sample efficient text summarization using a single pre-trained transformer. *ArXiv*, abs/1905.08836.
- Chloé Kiddon, Luke S. Zettlemoyer, and Yejin Choi. 2016. Globally coherent text generation with neural checklist models. In *EMNLP*.

- Mert Kilickaya, Aykut Erdem, Nazli Ikizler-Cinbis, and Erkut Erdem. 2017. Re-evaluating automatic metrics for image captioning. In *ACL*.
- Rik Koncel-Kedziorski, Dhanush Bekal, Yi Luan, Mirella Lapata, and Hannaneh Hajishirzi. 2019. Text generation from knowledge graphs with graph transformers. In *NAACL-HLT*.
- Wojciech Kryściński, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. Evaluating the factual consistency of abstractive text summarization. In *EMNLP*.
- Chin-Yew Lin. 2004. [ROUGE: A package for automatic evaluation of summaries](#). In *Text Summarization Branches Out*, pages 74–81. Association for Computational Linguistics.
- Peter J. Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. 2018. Generating wikipedia by summarizing long sequences. In *ICLR*.
- Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *ACL*.
- Hongyuan Mei, Mohit Bansal, and Matthew R. Walter. 2016. Coherent dialogue with attention-based language models. In *AAAI*.
- 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 *EMNLP*.
- Ani Nenkova and Kathleen McKeown. 2012. A survey of text summarization techniques. In *Mining Text Data*.
- Nikola I. Nikolov, Michael Pfeiffer, and Richard H. R. Hahnloser. 2018. Data-driven summarization of scientific articles. In *7th International Workshop on Mining Scientific Publications, LREC 2018*.
- Lianhui Qin, Antoine Bosselut, Ari Holtzman, Chandra Bhagavatula, Elizabeth Clark, and Yejin Choi. 2019. Counterfactual story reasoning and generation. In *EMNLP*.
- Dragomir R. Radev, Pradeep Muthukrishnan, Vahed Qazvinian, and Amjad Abu-Jbara. 2009. The acl anthology network corpus. In *Language Resources and Evaluation*, volume 47, pages 919–944.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. [Language models are unsupervised multitask learners](#). Unpublished manuscript.
- Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. [A neural attention model for abstractive sentence summarization](#). In *EMNLP*, pages 379–389.
- Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *ACL*.
- Sofia Serrano and Noah A. Smith. 2019. Is attention interpretable? In *ACL*.
- Simeng Sun, Ori Shapira, Ido Dagan, and Ani Nenkova. 2019. How to compare summarizers without target length ? pitfalls , solutions and re-examination of the neural summarization literature. In *NAACL 2019 Workshop on Optimizing and Evaluating Neural Language Generation (NeuralGen)*.
- Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Neural Information Processing Systems*, pages 3104–3112.
- David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or fiction: Verifying scientific claims. In *EMNLP*.
- Jiacheng Xu, Zhe Gan, Y. Cheng, and Jing jing Liu. 2020. Discourse-aware neural extractive text summarization. In *ACL*.
- Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020a. Bertscore: Evaluating text generation with bert. In *ICLR*.
- Xingxing Zhang, Furu Wei, and Ming de Zhou. 2019. Hibert: Document level pre-training of hierarchical bidirectional transformers for document summarization. In *ACL*.
- Yuhao Zhang, Derek Merck, Emily Bao Tsai, Christopher D. Manning, and Curtis P. Langlotz. 2020b. Optimizing the factual correctness of a summary: A study of summarizing radiology reports. In *ACL*.

A Appendices

A.1 Additional Implementation Details

A.2 Baselines

For the sequence-to-sequence RNN model, a bi-LSTM is used to encode a given source article a and a separate decoder LSTM produces the generated summary g . At each decoding time step, the decoder attends to all the context vectors produced by the encoder as well as the maintained state from the previous decoder tokens to produce the next token in the summary.

The Pointer-Generator (PGEN + Cov) model extends the base LSTM model (LSTM + Cov) to allow tokens to be copied from the input during generation. Baselines are trained for up to 40000 steps with a batch size of 16. Following previous work, we decode from these baselines using beam search with a beam size of 4.

A.3 Generator Model

We use the 345M parameter GPT-2 model. The model is trained to minimize the negative log likelihood of the next word w_i given all preceding words:

$$\mathcal{L}_{gen} = - \sum_{i=1}^{|a|+|s|} \log P(w_i | w_0, \dots, w_{i-1}) \quad (10)$$

where w_i is the i^{th} token of our full input vector X , a is our article and s is our summary. At test time, X only consists of the gold article and delimiter token ($a_1, \dots, a_{|a|}, [\text{SEP}]$) and we decode generated summaries g starting from this input.

During generation, we filter candidate summaries from the hypothesis generation pool that contain sentences longer than a fixed max length of 200 tokens, a clear sign of coherence deterioration. We use a candidate pool size of 30 for ATLAS and 20 for AAN.

A.4 Discriminator Training

Factuality Discriminator Details For the token-level classification model, we use the BERT base model with binary labels for whether or not a token should be included in a salient span. We predict for all spans in an abstract at once.

Order Discriminator Details We set the max length of summaries considered by the order discriminator to be 10 sentences, truncating longer summaries. Given the max length of a summary,

we have a fixed number of orderings $|O|$ that can be scored. We calculate the final score from the order discriminator based on the unnormalized sum of scores from these orderings, S , and the following function f_n :

$$f_n(S, |O|) = \frac{S - (-|O|)}{|O| - (-|O|)} \quad (11)$$

Sentence Selection for Discriminator Models

To train an adjacency discriminator model, we use a subset of adversarial and positive sentence pair examples extracted from the training set. The sentence pairs are extracted from gold abstracts containing at least five sentences using the following approach: For a randomly selected sentence s_u from the abstract, we randomly select an adjacent sentence, s_{u-1} or s_{u+1} , as a positive example and any nonadjacent sentence $s_{v \notin [u-1, u, u+1]}$ as a negative example.

Discriminator Performance We measure the performance of discriminator models using recall, precision, accuracy and F1. Table 9 provides summary statistics of discriminator performance on the various discourse and factuality objectives. Discourse-Adj denotes the adjacency discriminators, while Discourse-Abs denotes the discourse role label prediction model (Cohan et al., 2019) and Factuality denotes the token saliency prediction model.

Model	Training Data	Prec	Rec	F1	Acc
Discourse-Adj	ArXiv-AAN	86.05	85.25	85.65	86.81
Discourse-Adj	ArXiv-Bio	90.30	93.44	91.84	92.32
Discourse-Abs	CSAbstract	88.99	89.09	89.04	89.00
Factuality	SciFact	73.70	70.50	72.10	75.70

Table 9: Automatic Evaluation of discriminator architectures

A.5 Details on Model Performance

Automatic results for Co-opNet selection were given using a context size of 800 tokens for the input, while a context size of 800 characters was used to select Co-opNet summaries for the human eval. The automatic results for the summaries used in the human eval were lower than the ones using the longer context size. Using a smaller context size leads to faster and more efficient Co-opNet selection (less memory usage), but slightly lower overall automatic performance (while maintaining the same ordering in terms of highest and lowest scores for Co-opNet variants on ROUGE).

A.6 Comparison of Datasets

We removed duplicates and articles without abstracts from AAN. From this subset, we extract introduction and abstract pairs.

A.7 Additional Analysis

Comparison with Gold Summaries To obtain an upper-bound comparison for the human evaluation and verify the effectiveness of our human evaluation pipeline for judging the quality of abstracts, we used the same intro-abstract pairs and Mturk annotation framework as the model comparison to conduct a Turing-style evaluation. In this evaluation, we presented a Co-opNet (adj) generated abstract and a gold abstract to the annotators in a random ordering without noting whether either of the abstracts were human-written or machine-generated. We found that annotators consistently selected the gold abstract over the machine-generated abstract when considering factuality and coherence, though they found the machine-generated abstracts to be slightly more abstractive. We provide the results for this full evaluation in Table 11.

Dataset	Narrative Flow?	# Summaries	Avg # Sents	Avg # Words
XSum (Narayan et al., 2018)	✗	226,711	1.00	23.26
Newsroom (Grusky et al., 2019)	✗	1,321,995	1.45	26.70
CNN (Hermann et al., 2015)	✗	92,579	3.59	45.70
DailyMail (Hermann et al., 2015)	✗	219,506	3.86	54.65
ArXiv	✓	472,493	6.11	150.85
AAN	✓	11,890	5.03	106.76

Table 10: Statistics of gold summaries in different summarization datasets.

Gold vs. Co-opNet-Adj		
Criteria	Gold	Co-opNet
Abstractiveness	47.37	52.63
Coherence	66.67	31.06
Factuality	66.92	32.33
Overall	61.36	32.33

Table 11: Human Evaluation of Co-opNet Architectures (% of judgements for each model)