Detection and Mitigation of the Negative Impact of Dataset Extractivity on Abstractive Summarization

Yubin Ge[1], **Sullam Jeoung**[1], **Ly Dinh**[2], **Jana Diesner**[1]
[1]University of Illinois at Urbana-Champaign, USA
[2]University of South Florida, USA
`{yubinge2, sjeoung, jdiesner}@illinois.edu`
`lydinh@usf.edu`

Abstract

In text summarization, extractivity is defined as a measurement of the degree of overlap between a source document and its summary. Previous research has shown that the extractivity level of training data can influence both output extractivity and the amount of factual information (i.e. faithfulness) in outputs for abstractive summarization. However, it remains unclear if and how extractivity impacts the performance of abstractive models. In this work, we investigate the relationship between dataset extractivity and model performance by comparing the performance of trained models under different degrees of extractivity. We find that while low levels of extractivity can improve performance, as extractivity increases, performance is negatively impacted. Furthermore, through an analysis of the model's copy continuity of content, we discover that higher extractivity leads to a greater tendency for the model to copy text continuously from the source document rather than identifying and summarizing important content that should be covered in the target summary. To address these issues, we propose a simple and effective method to design copy labels for fixing the model's copying behaviors and train the model with a copy mechanism. The experimental results illustrate the effectiveness of our strategy in alleviating the negative impact on model performance resulting from high dataset extractivity, and that our method outperforms several competitive baselines.

1 Introduction

Text summarization is the task of reducing the content of an original text while preserving its salient content (Gupta and Gupta, 2019). A multitude of techniques to generate summaries exists, which can be categorized into extractive and abstractive approaches. With extractive techniques, a summary is generated by extracting salient sentences from the text (Kasture et al., 2014). With abstractive techniques, new sentences or commonly called paraphrased sentences are generated so that the resulting summaries contain new words and expressions that do not occur in the original text (Widyassari et al., 2020).

In addition to the extractivity for summarization models, previous work has proposed the idea of extractivity for summarization datasets specifically in terms of the summarization styles in datasets (Grusky et al., 2018). More specifically, extractivity is a metric meant to capture the overlap between a text and its summary, and quantifies the extent to which a summary is a derivative of a text.

Summarization datasets typically contain aligned pairs of texts and human-generated or human verified summaries. As an intrinsic property of summarization datasets, extractivity has been shown to influence various characteristics of output summaries, such as output extractivity (See et al., 2017; Zhang et al., 2018a) and the amount of factual information (i.e. faithfulness) (Ladhak et al., 2022). Generally, datasets with high extractivity are treated as suboptimal for abstractive summarization (Bommasani and Cardie, 2020), but there has been a lack of systematic research investigating the relationship between the magnitude of extractivity in training datasets and the performance of abstractive summarization models.

To address the lack of understanding about the relationship between extractivity in summarization training data and the performance of abstractive summarization models, in this work, we first develop a framework for evaluating the change in model performance over baseline models (control) at three levels of amount of extractivity. Specifically, we split the training data into different subsets based on their increasing extractivity and then train the control models on each resulting subset. By comparing the results, we find that as the extractivity in training data increases, model performance initially improves with increasing extractivity but then drops when extractivity is too high. This naturally raises the next question, which we also address in this paper: Why and how does extractivity cause model performance to change?

Based on our observations, we posit that high extractivity in training data can lead to an increased likelihood of overfitting an abstractive model whereby the model continuously copies from source documents. To validate this hypothesis, we propose new metrics to measure how much content in output summaries is copied continuously from a source text, and how important this continuously copied content is by comparing it with target summaries. We apply these metrics to outputs from the control models defined above. The results support our hypothesis ahd provide a new direction for mitigating the negative effects of high extractivity by promoting the selection of truly important content that is covered in target summaries for copying.

Through our analysis, we have identified a primary harm associated with highly extractive training data, namely the propensity for models to excessively rely on continuous copying rather than focusing on the essential content that should be included in summaries. To alleviate this potential harm, we propose and test a simple and effective strategy based on a copy mechanism (See et al., 2017). Our solution first identifies the salient parts in a source document that the model should copy into the summary. We do this by solving an integer linear programming. We then create ground-truth labels based on the optimal solution for the copy distribution, with the goal of guiding the model to focus on the content it should copy rather than blindly copying continuous text spans from the source document. Lastly, we incorporate this copy mechanism into BART and train it with an auxiliary loss based on the copy distribution and the constructed copy labels. The experimental results demonstrate that our method not only is effective in mitigating the negative effects of high extractivity, but can also improve model performance as it outperformed several recent competitive baselines.

In general, we aim to explore and answer the following research questions in this work:

• **RQ1** How does dataset extractivity impact the performance of abstractive summarization models? (Section 2)

• **RQ2** How does dataset extractivity cause the performance of abstractive models to change? (Section 3)

• **RQ3** For cases when dataset extractivity hurts model performance, how can we mitigate this negative impact? (Section 4)

2 Extractivity Analysis

In this section, we introduce our research design, following a similar process as Ladhak et al. (2022), to answer our first research question: How does the extractivity in summarization datasets influence the performance of abstractive models?

2.1 Datasets and Extractivity

We selected two popular summarization datasets, one from the domain of news (i.e. CNN/DM) and one from academic publishing (i.e. arXiv/Pubmed). We opted for these two datasets as they are similar in size and dataset extractivity, and thus would warrant a fair comparison:

CNN/DM (Hermann et al., 2015; Nallapati et al., 2016) is a dataset of news articles from CNN and Daily Mail. The summaries are formed from highlighted bullet points. There are 311,971 document - summary pairs in the dataset.

arXiv/PubMed (Cohan et al., 2018) contains 346,187 papers from arXiv and PubMed. This dataset consists of the processed full text of papers as input documents, and the abstracts are used as summaries of the papers.

We used the common measurements of extractivity detailed in Grusky et al. (2018), and specifically chose extractive fragment coverage as Ladhak et al. (2022). Specifically, coverage calculates the percentage of words in a summary that are from the source document. The higher the coverage of the summary, the higher the overlap of the summary with the source article.

2.2 Models and Performance Metrics

We focused on two widely-used baselines:

BART (Lewis et al., 2020) is a Transformer-based denoising autoencoder for pre-training seq2seq tasks including both natural language understanding and generation tasks.

PEGASUS (Zhang et al., 2020) is a Transformer-based encoder-decoder model pre-trained with a new objective function by generating removed important sentences from the remaining sentences.

Following existing work, ROUGE (Lin, 2004) and BERTScore (Zhang et al., 2019) are used as our performance metrics to evaluate models.

Dataset	DE	BART					PEGASUS				
		OE	R1	R2	RL	BERTScore	OE	R1	R2	RL	BERTScore
CNN/DM											
full	86.41	94.89	44.12	21.21	40.86	34.85	93.41	44.15	21.41	41.02	34.98
T1	70.96	89.71	40.80	18.17	37.65	30.43	90.14	43.14	19.84	40.01	33.73
T2	83.08	93.26	43.18	20.18	40.05	33.60	92.77	44.01	20.83	40.89	34.60
T3	92.01	96.48	38.50	16.45	35.22	27.31	95.25	43.37	20.74	40.24	34.00
arXiv/PubMed											
full	93.64	88.20	45.81	18.15	41.57	23.89	89.11	45.20	18.93	41.83	23.96
T1	78.66	85.92	44.32	17.10	39.93	22.32	87.44	43.44	17.42	39.26	22.98
T2	92.11	88.18	45.48	17.95	40.88	23.34	88.80	44.90	18.25	41.10	23.73
T3	96.37	88.59	44.03	16.91	39.53	22.10	89.73	43.14	16.98	39.66	22.40

Table 1: Empirical results of model performance under different level extractivity. DE and OE denote dataset extractivity and output extractivity, respectively, which correspond to the extractivity calculated for training data or model outputs. T1, T2, and T3 represent the triplet subsets we split.

2.3 Extractivity-Performance Trade-off

To examine the relationship between dataset extractivity and model performance, we first fine-tuned models on training data with different levels of extractivity. We then evaluated these models with the same test data to explore the extractivity-performance trade-off and further investigated how dataset extractivity influences model performance.

Specifically, we split the training data into three equally-sized subsets by computing the coverage values for all data instances and sorting the training data based on these values. For each subsets, we fine-tuned a separate model with the corresponding level of extractivity, resulting in three fine-tuned models. We call each of these trained models a *triplet model*. In addition, we fine-tuned a model on the full training data for comparison.

2.3.1 Results and Analysis

The results are presented in Table 1. By comparing the outputs from triplet models, we learn that the extractivity of output summaries increases as the extractivity of training data increases. This aligns with previous empirical findings where limited abstractivity was shown in abstractive systems trained on highly extractive datasets (See et al., 2017; Zhang et al., 2018b). Overall, it is evident that dataset extractivity does indeed influence the extractivity of output summaries in trained abstractive models. It is therefore worthwhile to investigate the potential impact of dataset extractivity on model performance.

More importantly, we observed for both training datasets that as the extractivity in training data increases, a similar pattern in changes in performance metrics as per BART and PEGASUS occurs: **model performance initially improves but subsequently drops when dataset extractivity goes too high**. By too high, we mean that there may exist a certain threshold for extractivity, and once the extractivity surpasses this threshold, a subsequent decline in model performance may be observed. However, the magnitude of the performance gap between the metrics varies for different model-dataset pairs. This suggests that the impact of dataset extractivity on model performance depends on the degree of extractivity in training data. As different summarization datasets may have various degrees of extractivity (Bommasani and Cardie, 2020), a detailed analysis is required to understand how this impact may bring changes to model performance and identify potential solutions to mitigate its negative effects on model performance. We provide this analysis next.

3 Continuity Analysis

Now that we have reached the conclusion that high extractivity can negatively influence abstractive model performance, our our second question to answer is: How does dataset extractivity cause the performance of abstractive models to change?

After examining the model outputs, we formed a plausible hypothesis: **extractivity could lead to an increased tendency or preference for an abstractive model to copy content continuously, potentially neglecting the inclusion of important content that should have been covered in a summary.** To validate this hypothesis, it is necessary to develop new metrics to measure the continuity and salience of text spans that are continuously copied by the model. These metrics can be applied to the output summaries from all triplet models (as defined in Sec. 2) and the results can be compared to test the validity of the our hypothesis.

3.1 Characterizing Continuity

We examined the summary continuity at the sentence level using two metrics which we designed to evaluate the proportion of the text in a summary that is continuously copied from the source document and the importance of these copied texts by comparing them with the ground-truth summary.

Given a source document to be summarized $X = \{x_1, x_2, \cdots, x_{|X|}\}$ consisting of a sequence of sentences x_j and the output summary $\hat{Y} = \{\hat{y}_1, \hat{y}_2, \cdots, \hat{y}_{|\hat{Y}|}\}$ consisting of sentences $\hat{y}_i$, we define the set of *continuous spans* $\mathcal{C}(X, \hat{Y})$ as the text spans in $\hat{Y}$ that are copied continuously from X. To obtain these continuous spans, we first identified whether a sentence in $\hat{Y}$ is copied from X and then searched for any index subsets of copied sentences that are continuous. Specifically, for a given $\hat{y}_i \in \hat{Y}$ we calculated its similarity to each $x_j \in X$, and we marked the most similar x_j corresponding to $\hat{y}_i$ as $x_i^* = \operatorname*{argmax}_{x_j \in X} \text{sim}(\hat{y}_i, x_j)$. Then we set a threshold t, and if $\text{sim}(\hat{y}_i, x_i^*) > t$ [1] we treated $\hat{y}_i$ as having been copied from x_i^* and recorded the index of x_i^* in X as $k_i \in [1, n]$. Conversely, if $\text{sim}(\hat{y}_i, x_i^*) < t$, we regarded $\hat{y}_i$ as not having been copied from any sentence in X and set k_i to an arbitrary negative number -100. After conducting this process for every $\hat{y}_i \in \hat{Y}$, we got a sequence of indices $K = \{k_1, k_2, \cdots, k_m\}$, where $k_i \in -100 \cup [1, n]$. Lastly, we looped through the sequence K, and obtained all subsequences such that any non-negative element k_i in a subsequence satisfies $k_i = k_{i+1}$ or $k_i = k_{i+1} - 1$. For example, $K = \{-100, 5, 5, 6, 1\}$ contains a continuous span $\{5, 5, 6\}$ and this means the 2-nd to the 4-th sentences in $\hat{Y}$ are copied from the 5-th and the 6-th sentences in X continuously. Note that we skipped subsequences whose first element is equal to the last element, so $K = \{-100, 5, 5, 5, 1\}$ does not contain any continuous span. We also showed this process in Algorithm 1 and computed two metrics using $\mathcal{C}(X, \hat{Y})$: continuity and continuity salience.

Continuity We define*continuity* as the percentage of continuous spans in the output summary and the ratio is calculated based on sentence counts.

[1]We used ROUGE-2 as the function sim(.,.) and set the threshold $t = 0.6$. We tried other threshold values from 0.5 to 0.7 and obtained the same conclusions.

Algorithm 1 Identify Continuous Spans

```
function $\mathcal{C}(X, \hat{Y}, t)$
    $K \leftarrow \oslash, j \leftarrow 1$
    for $i \leftarrow 1$ to len($\hat{Y}$) do
        $max\_score \leftarrow -1, max\_idx \leftarrow -1$
        for $j \leftarrow 1$ to len($X$) do
            if sim($\hat{y}_i, x_j$) $\geq max\_score$ then
                $max\_score \leftarrow$ sim($\hat{y}_i, x_j$)
                $max\_idx \leftarrow j$
        if $max\_score > t$ then
            $K \leftarrow K \cup \{max\_idx\}$
        else
            $K \leftarrow K \cup \{-100\}$
    $\mathcal{C} \leftarrow \oslash, start\_idx \leftarrow -1, flag \leftarrow$ false
    for $i \leftarrow 1$ to len($K$) $- 1$ do
        if $k_i = k_{i+1} - 1$ or $0 < k_i = k_{i+1}$ then
            if $flag =$ false then
                $start\_idx \leftarrow i$
                $flag \leftarrow$ true
        else
            if $flag =$ true and $k_{start\_idx} \neq k_i$ then
                $c \leftarrow < k_{start\_idx}, \cdots, k_{i-1}, k_i >$
                $\mathcal{C} \leftarrow \mathcal{C} \cup \{c\}$
    return $\mathcal{C}$
```

We use $|.|$ to represent the sentence count of a sequence.

$$\textsc{Cont}(X, \hat{Y}) = \frac{1}{|\hat{Y}|} \sum_{c \in \mathcal{C}(X, \hat{Y})} |c|$$

Continuity Salience To quantify the salience of continuous spans, we propose *continuity salience* to calculate the normalized ROUGE scores[2] between the target summary and the text spans that a continuous span copies from X. This serves as an approximation of whether the copied text spans are important to be included in the summary. We denoted the target summary as Y and used a mapping function g to return the text spans in X that a continuous span copies from. We used $\|.\|$ to represent the word count of a sequence and AVG to represent the average function.

$$\textsc{Cont Salience}(X, \hat{Y}) = \operatorname*{AVG}_{c \in \mathcal{C}(X, \hat{Y})} \frac{\text{ROUGE}(g(c), Y)}{\|c\|}$$

3.2 Evaluating Continuity

We utilized the above-defined continuity and continuity salience to investigate the potential relationship between increasing dataset extractivity and the tendency of models to copy text more continuously, as well as whether the changes in the salience of continuous spans may contribute to the changes in model performance.

[2]We used ROUGE-2 specifically.

Dataset	DE	BART		PEGASUS	
		Cont	Cont Salience	Cont	Cont Salience
CNN/DM					
T1	70.96	10.00	0.057	14.31	0.078
T2	83.08	23.98	0.153	20.11	0.157
T3	92.01	43.68	0.107	34.66	0.107
arXiv/PubMed					
T1	78.66	12.22	0.082	21.13	0.105
T2	92.11	25.49	0.125	25.17	0.140
T3	96.37	31.86	0.123	29.91	0.101

Table 2: Empirical results of continuity and continuity salience under different level extractivity. DE denotes dataset extractivity. T1, T2, and T3 represent the triplet subsets we split.

The results are shown in Table 2. As one might reasonably expect, for both datasets and models, the continuity (Cont) increases as the training extractivity (TE) increases. This basically suggests that as there are more overlaps between source documents and corresponding summaries, the model may learn to copy more directly from a source document such that the model ignores to include the truly important content from the source documents into a summary. This may explain why there are more continuous spans in the output summaries from the model trained on the dataset with higher extractivity, as evidenced by higher continuity scores. Hence, it can be inferred that when dataset extractivity increases, the model is more likely to copy, and once it starts to copy it tends to keep copying the following content.

However, in summarization done by humans, there is no inherent rule that important content appearing in a summary usually clusters together in a source document. As a result, a summary that contains too many continuous spans might inadvertently exclude important content that should have been included in the summary. This is supported by our experimental results, which show that as dataset extractivity increases, continuity salience first increases and soon decreases. This also aligns with the change in model performance shown in Table 1. Based on connecting to previous observations in Section 2, we conclude that the more overlaps there are between source documents and target summaries in training data, the more likely are trained model to copy content continuously, which results in low salience in continuous spans and hurts model performance when dataset extractivity is excessively high.

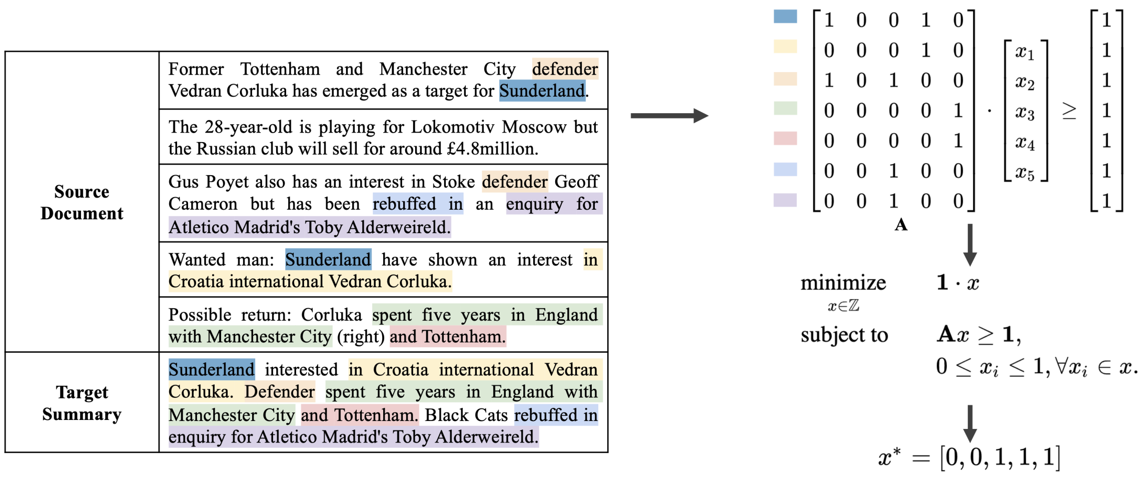
4 Mitigating the Negative Impact of High Extractivity on Model Performance

A key takeaway from previous sections in this paper is that high dataset extractivity can result in a tendency to excessively copy text from source documents while neglecting important content that should be included in a summary. To address this issue, a natural solution is to focus the model's attention on important content during training to encourage the model to cover the important content when copying during inference. To achieve this, we first identified a mapping between tokens in a source document X and extractive fragments in the target summary Y, where extractive fragments are the set of shared sequences of tokens in X and Y (Grusky et al., 2018). It should be noted that extractive fragments comprise all tokens the model can copy from the source document. Then, a copy mechanism was implemented (See et al., 2017) and we transformed the above mapping as the label for the copy distribution so that the model can attend to important texts when copying text from source documents.

4.1 Identifying Important Content to Copy

The process is illustrated in Figure 1. Essentially, this process was formulated as a simple optimization problem, with the goal of using the least number of sentences in X to cover all extractive fragments. We focused on extractive fragments since they are the tokens that a model can copy. The decision to select the least number of sentences in X was made because we usually require a summary to be concise, i.e., it would be preferable if fewer sentences were needed to cover extractive fragments.

We first converted the extractive fragments into



Source Document	Former Tottenham and Manchester City defender Vedran Corluka has emerged as a target for Sunderland.
	The 28-year-old is playing for Lokomotiv Moscow but the Russian club will sell for around £4.8million.
	Gus Poyet also has an interest in Stoke defender Geoff Cameron but has been rebuffed in an enquiry for Atletico Madrid's Toby Alderweireld.
	Wanted man: Sunderland have shown an interest in Croatia international Vedran Corluka.
	Possible return: Corluka spent five years in England with Manchester City (right) and Tottenham.
Target Summary	Sunderland interested in Croatia international Vedran Corluka. Defender spent five years in England with Manchester City and Tottenham. Black Cats rebuffed in enquiry for Atletico Madrid's Toby Alderweireld.

$$\begin{array}{c} \text{🟦} \\ \text{🟨} \\ \text{🟧} \\ \text{🟩} \\ \text{🟥} \\ \text{🟦} \\ \text{🟪} \end{array} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}}_{\mathbf{A}} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} \geq \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$$\begin{aligned} \underset{x \in \mathbb{Z}}{\text{minimize}} \quad & \mathbf{1} \cdot x \\ \text{subject to} \quad & \mathbf{A}x \geq \mathbf{1}, \\ & 0 \leq x_i \leq 1, \forall x_i \in x. \end{aligned}$$

$$x^* = [0, 0, 1, 1, 1]$$

Figure 1: An example from the CNN/DM dataset that shows how to set up the optimization for identifying important content to copy. We highlight different fragments with different colors and use squares with the corresponding colors on the left of the matrix $\mathbf{A}$ to represent the bipartite mapping between a fragment and sentences in the source document. For example, the first row in $\mathbf{A}$ means the fragment Sunderland appears in the 1-st and 4-th sentences in the source document. After solving the optimization problem, the optimal solution x^* indicates that to cover all fragments we only need to select the last three sentences.

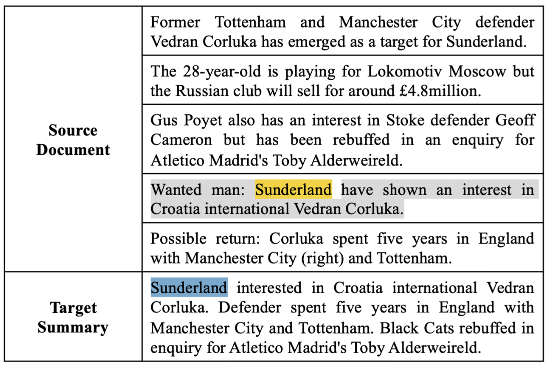
a bipartite mapping between fragments in the target summary Y and sentences in the source document X, and represented the mapping in a binary matrix $\mathbf{A}$ with each row corresponding to a fragment and each column corresponding to a sentence in X. We set $\mathbf{A}_{ij}$ to 1 if the i-th fragment appears in the j-th sentence in X, otherwise $\mathbf{A}_{ij}$ is set to 0. A fragment can appear in multiple sentences. Then we defined a binary vector x of size equal to the number of sentences in X where each element x_i indicates whether the i-th sentence in X should be selected. The optimization problem is formalized as the following integer linear programming (ILP):

$$\begin{aligned} \underset{x \in \mathbb{Z}}{\text{minimize}} \quad & \mathbf{1} \cdot x \\ \text{subject to} \quad & \mathbf{A}x \geq \mathbf{1}, \\ & 0 \leq x_i \leq 1, \forall x_i \in x. \end{aligned} \tag{1}$$

In this ILP, $\mathbf{1}$ means all-one vector, and $\mathbb{Z}$ represents the set of integers. Our objective function basically counts the number of selected sentences in the source document X. The first constraint guarantees that all fragments can be covered and the second constraint assures that each sentence can be selected either once or never. We can prove that this optimization is always feasible. The proof is included in Appendix Section A.1.

Proposition 1 *The optimization problem* (1) *is always feasible.*

By solving this ILP, the optimal solution x^* indicates which sentences in X should be selected and



Source Document	Former Tottenham and Manchester City defender Vedran Corluka has emerged as a target for Sunderland.
	The 28-year-old is playing for Lokomotiv Moscow but the Russian club will sell for around £4.8million.
	Gus Poyet also has an interest in Stoke defender Geoff Cameron but has been rebuffed in an enquiry for Atletico Madrid's Toby Alderweireld.
	Wanted man: Sunderland have shown an interest in Croatia international Vedran Corluka.
	Possible return: Corluka spent five years in England with Manchester City (right) and Tottenham.
Target Summary	Sunderland interested in Croatia international Vedran Corluka. Defender spent five years in England with Manchester City and Tottenham. Black Cats rebuffed in enquiry for Atletico Madrid's Toby Alderweireld.

Figure 2: The same example from CNN/DM as used before, this time to show the gold copied token and silver copied tokens for the fragment Sunderland, as defined in Section 4.2. The highlighted Sunderland is its *gold copied token* and the tokens highlighted in silver are its *silver copied tokens*, including Sunderland.

can be further regarded as important content that a model should copy.

4.2 Creating Copy Labels

Our next goal is to transform the optimal solution to the ILP into labels to correct the model's copy behaviors. In this context, a source document X is considered as a sequence of tokens $\{x^w_1, x^w_2, \cdots, x^w_a\}$ and the target summary Y is represented as a token sequence $\{y^w_1, y^w_2, \cdots, y^w_b\}$. Our target copy labels are represented as a matrix $\mathbf{M} \in \mathbb{R}^{b \times a}$ where $\mathbf{M}_{ij}$ indicates the importance of the source token x^w_j for the model to copy it in order to generate the target token y^w_i.

Given a target token y^w_i that appears in the extractive fragment f, we first referred to the optimal solution x^* and the bipartite mapping $\mathbf{A}$ to find the selected source sentence x_f^{sent} that covers the fragment f. We then defined two categories for source tokens $\{x_1^w, x_2^w, \cdots, x_a^w\}$ corresponding to the target token y_i^w, as shown in Figure 2.

Gold Copied Tokens We define *gold copied tokens* as the source tokens that appear in the selected sentence x_f^{sent} and also present in the extractive fragment f. These *gold copied tokens* are the expected source tokens where the target token y_i^w should be coped from, based on the optimal solution x^*.. We used the function $\text{GOLDCOPY}(x_j^w, y_i^w)$ to represent whether a source token x_j^w is a gold copied token to the target token y_i^w, and $\text{GOLDCOPY}(x_j^w, y_i^w) = 1$ if it is otherwise 0.

Silver Copied Tokens We define *silver copied tokens* as the source tokens that appear in the selected sentence x_f^{sent}. Similarly, the function $\text{SILVERCOPY}(x_j^w, y_i^w)$ is set to return 1 if x_j^w is a silver copied token to the target token y_i^w otherwise 0.

Additionally, we initialized $\mathbf{M}$ based on the ROUGE scores $\mathbf{R}$ between the two sentences where x_j^w and y_i^w belong, for all source and target tokens pairs. Basically $\mathbf{R}_{ij} = \text{ROUGE}(x_m, y_n)$ where $x_j^w \in x_m$ and $y_i^w \in y_n$. Finally we obtained the copy labels as follows:

$$\begin{aligned}\widetilde{\mathbf{M}_{ij}} =& \lambda_1 \cdot \text{GOLDCOPY}(x_j^w, y_i^w) \\ &+ \lambda_2 \cdot \text{SILVERCOPY}(x_j^w, y_i^w) \qquad (2)\\ &+ \lambda_3 \cdot \mathbf{R}_{ij}, \\ \mathbf{M} =& \text{softmax}(\widetilde{\mathbf{M}}) \qquad (3)\end{aligned}$$

where λ_1, λ_2 and λ_3 are hyper-parameters. Please note that for target tokens that do not appear in any fragments, all source tokens are neither their gold nor silver copied tokens, as there is no way for the model to generate them via copying. As we intended to adopt $\mathbf{M}$ as the ground truth labels to guide the model to copy correct content, it is crucial that the model focuses the most on gold copied tokens when copying fragments, and then attend more to the silver copied tokens than the rest of tokens. Therefore, we set $\lambda_1 > \lambda_2 > \lambda_3$.

4.3 Training with Copy Mechanism

It has been widely shown that a copy mechanism can be interpreted as modeling copy distributions. Numerous previous studies have also demonstrated the effectiveness of the copy mechanism in abstractive summarization (See et al., 2017; Xu et al., 2020; Li et al., 2021). Due to limited space, we refer readers to See et al. (2017) for more details. Inspired by their work, we utilized a copy mechanism and took the encoder-decoder attention based on the last encoder and decoder hidden layers as the copy distribution:

$$\alpha_{ij} = \text{softmax}\Big(\frac{(\mathbf{W}^e h_j^e)^\top \mathbf{W}^d h_i^d}{\sqrt{d_k}}\Big), \qquad (4)$$

where $\mathbf{W}^e$ and $\mathbf{W}^d$ are weight matrices for encoder and decoder hidden states respectively, h_j^e is the j-th hidden state of the encoder, h_i^d is the i-th hidden state of the decoder and d_k is the dimension of hidden states. Note that for the multi-head attention, we calculated the copy distributions as the average of multiple heads. Finally, in order to guide the model to copy correctly, we adopted an auxiliary loss function to encourage the consistency between the overall copy distribution α and the copy labels $\mathbf{M}$ based on the Kullback-Leibler (KL) divergence:

$$\mathcal{L} = -\frac{1}{\|Y\|}\sum_i \log P(y_i^w) + \lambda \text{KL}(\alpha, \mathbf{M}), \quad (5)$$

where $P(y_i^w)$ is the likelihood of the target token y_i^w, $\|Y\|$ is the length of the target token sequence, and λ is the hyper-parameter.

4.4 Results and Analysis

We implemented our proposed strategy for mitigating the extractive-performance trade-off in BART and refer to that as *BART + Copy Labels*. We evaluated our method on CNN/DM and arXiv/PubMed.

4.4.1 Results on CNN/DM

Except for BART and PEGASUS, we compared our method to several competitive baselines that also use copy mechanism:

- **SAGCopy**(Xu et al., 2020) fine-tunes MASS (Song et al., 2019) by incorporating the importance scores for source words into copying mechanism.
- **PALM** (Bi et al., 2020) incorporates the copy mechanism into the pre-training model.
- **CoCoNet**(Li et al., 2021) enhances the copying mechanism by encouraging the model to

Model	R1	R2	RL
BART[†]	44.12	21.21	40.86
PEGASUS[†]	44.15	21.41	41.02
BART + AttnCopy[‡]	44.26	21.31	40.98
BART + SAGCopy[‡]	44.31	21.35	41.00
PALM[‡]	44.30	21.12	41.41
CoCoNet[‡]	44.39	21.41	41.05
BART + Copy Labels	**45.21**	**21.83**	**41.80**

Table 3: Experimental results on CNN/DM. † indicates the results are from our implementation, and ‡ means the results are taken from the corresponding papers.

Model	OE	Cont	Cont Salience
BART[†]	94.89	26.92	0.169
PEGASUS[†]	93.41	22.15	0.171
BART + Copy Labels	95.24	20.23	0.184

Table 4: Extractivity analysis results on CNN/DM. **OE** denotes output extractivity

copy the input word that is relevant to the previously copied one.

From Table 3 we can see that our method has superior performance compared to other baselines. We believe this is because our copy labels provide more effective supervision for the copy distribution. Additionally, we performed an extractivity analysis and the results are shown in Table 4. It can be observed that adding copy mechanism with the proposed copy labels does not result in significant changes in the output extractivity, but lowers the continuity in output summaries. Meanwhile, since our copy labels encourage the model to focus on important content when copying from source documents, the continuity salience of our method is higher than that of other baselines, which shows the effectiveness of our method in mitigating the negative impact of high dataset extractivity. A case study is shown in Appendix Section A.3.

4.4.2 Results on arXiv/PubMed

We conducted similar evaluations on arXiv/PubMed, and the results are shown in Table 5 and Table 6. These results align with the findings from the CNN/DM evaluations, in that our copy labels can improve the model's performance, as demonstrated by the ROUGE scores. Besides, the extractivity analysis further supports the conclusion that our proposed copy labels can guide the model to copy important content, such as the increase of continuity salience, which leads to an improvement in performance. In summary, our method is lightweight, as it only involves a copy mechanism with an auxiliary loss. Furthermore, it effectively alleviates the harm from high dataset extractivity and improves model performance.

Model	R1	R2	RL
BART[†]	45.81	18.15	41.57
PEGASUS[†]	45.20	18.93	41.83
BART + Copy Labels	**45.97**	**19.04**	**41.87**

Table 5: Experimental results on arXiv/PubMed. † indicates the results are from our implementation.

Model	OE	Cont	Cont Salience
BART[†]	88.20	27.00	0.133
PEGASUS[†]	89.11	26.85	0.144
BART + Copy Labels	88.73	24.12	0.151

Table 6: Extractivity analysis results on arXiv/PubMed. **OE** denotes output extractivity

5 Related Work

Abstractive Summarization Text summarization can be broadly classified into two categories: *extractive* and *abstractive*, and our work focuses on abstractive summarization. Recent works in abstractive summarization have explored various approaches, such as contrastive learning (Xu et al., 2022; Liu et al., 2022), offline reinforcement learning (Pang and He, 2020), and a two-staged framework (Liu and Liu, 2021). Our approach focuses on characteristics of training datasets and aims to identify and mitigate negative impact of high extractivity in training data on model performance, which can be orthogonal to model structures or training strategies.

Extractivity in Summarization Bommasani and Cardie (2020) studied the quality of summarization datasets and found that a high degree of extractivity is present in many datasets. As existing datasets display significant amounts of extractivity, it is necessary to investigate how a model trained on such data may be influenced by extractivity. Similar to our approach, Ladhak et al. (2022) examined the effect of extractivity on the faithfulness of models. They showed that an increase in extractivity improves the faithfulness of the model but also that a trade-off exists between abstractivity and faithfulness. Our work focuses on the relationship between dataset extractivity and model performance.

Zhang et al. (2018a) explored the abstractivity of summarization models and found that abstractive models exhibit near-extractive behaviors in practice. However, their analysis is constrained to CNN/DM dataset, which we also used, and RNN-based models, whereas our approach extends to arXiv/PubMed and transformer-based pre-trained models. Kryściński et al. (2018) proposed methods to increase abstractivity in the output. Unlike their work, our work introduces and evaluates a method to mitigate the negative impacts of high dataset extractivity on model performance.

6 Conclusion

In this work, we have explored how the amount of extractivity in summarization training datasets influences the performance of abstractive models. By comparing model performance under different levels of dataset extractivity, we showed that low levels of extractivity can improve model performance while the impact becomes negative as extractivity is high. Furthermore, with the analysis of the model's copy continuity of content, we found that high dataset extractivity encourages the model to copy text continuously, which can cause models to ignore important content. In order to mitigate these negative effects, we presented a novel, simple, and effective strategy that creates labels for fixing the model's copying behaviors. By training the model with a copy mechanism and our copy labels, our experimental results show that our method can effectively alleviate the harm resulting from dataset extractivity and outperforms several competitive baselines.

7 Limitations

Our work has the following limitations. First, our analysis of continuity and continuity salience only focused on the sentence level. This is limiting since actual continuous spans can be a part of tokens in a identified copied sentence. Besides, we only utilized string-based overlap for salience estimation, i.e. ROUGE. This can be limiting since semantic salience may not be captured. Furthermore, even if our method can alleviate the negative impact of high dataset extractivity, it may not fully address this issue. In the future, we plan to extend our analysis to token-based continuous spans identification and semantic based measurement for more accurate continuity quantification.

Ethics Statement

The progress in deep neural network architectures and the availability of large pre-trained language models have led to significant advancements with single document summarization. However, current state-of-the-art natural language processing (NLP) solutions still face challenges in consistently generating factual and faithful summaries without any instances of hallucination (Maynez et al., 2020). Therefore, it is imperative to acknowledge that our proposed solution, like previous approaches, is not yet suitable for deployment as it does not specifically address the issue of hallucination. To bridge this gap, future research efforts should prioritize the development of more effective evaluation measures and solutions for text summarization, aiming to ensure highly faithful summaries that accurately represent the source content and enhance the overall trustworthiness of summarization systems. Additionally, in the case of applying the proposed method to sensitive data domains such as medical patient records and legal documents, it becomes essential to incorporate privacy-preserving policies to safeguard the confidentiality of personal information (Da Silva et al., 2006). These measures are critical to instill confidence in the practical implementation of text summarization techniques.

References

Bin Bi, Chenliang Li, Chen Wu, Ming Yan, Wei Wang, Songfang Huang, Fei Huang, and Luo Si. 2020. Palm: Pre-training an autoencoding&autoregressive language model for context-conditioned generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8681–8691.

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

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.

Josenildo Costa Da Silva, Matthias Klusch, Stefano Lodi, and Gianluca Moro. 2006. Privacy-preserving agent-based distributed data clustering. *Web Intelligence and Agent Systems: An International Journal*, 4(2):221–238.

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)*, New Orleans, Louisiana. Association for Computational Linguistics.

Som Gupta and Sanjai Kumar Gupta. 2019. Abstractive summarization: An overview of the state of the art. *Expert Systems with Applications*, 121:49–65.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. *Advances in neural information processing systems*, 28.

NR Kasture, Neha Yargal, Neha Nityanand Singh, Neha Kulkarni, and Vijay Mathur. 2014. A survey on methods of abstractive text summarization. *Int. J. Res. Merg. Sci. Technol*, 1(6):53–57.

Wojciech Kryściński, Romain Paulus, Caiming Xiong, and Richard Socher. 2018. Improving abstraction in text summarization. *arXiv preprint arXiv:1808.07913*.

Faisal Ladhak, Esin Durmus, He He, Claire Cardie, and Kathleen Mckeown. 2022. Faithful or extractive? on mitigating the faithfulness-abstractiveness trade-off in abstractive summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1410–1421.

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

Haoran Li, Song Xu, Peng Yuan, Yujia Wang, Youzheng Wu, Xiaodong He, and Bowen Zhou. 2021. Learn to copy from the copying history: Correlational copy network for abstractive summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4091–4101.

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.

Yixin Liu and Pengfei Liu. 2021. SimCLS: A simple framework for contrastive learning of abstractive summarization. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, Online. 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.

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.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290.

Richard Yuanzhe Pang and He He. 2020. Text generation by learning from demonstrations. In *International Conference on Learning Representations*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. Mass: Masked sequence to sequence pre-training for language generation. In *International Conference on Machine Learning*, pages 5926–5936. PMLR.

Adhika Pramita Widyassari, Supriadi Rustad, Guruh Fajar Shidik, Edi Noersasongko, Abdul Syukur, Affandy Affandy, et al. 2020. Review of automatic text summarization techniques & methods. *Journal of King Saud University-Computer and Information Sciences*.

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 (EMNLP)*.

Shusheng Xu, Xingxing Zhang, Yi Wu, and Furu Wei. 2022. Sequence level contrastive learning for text summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11556–11565.

Song Xu, Haoran Li, Peng Yuan, Youzheng Wu, Xiaodong He, and Bowen Zhou. 2020. Self-attention guided copy mechanism for abstractive summarization. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 1355–1362.

Fang-Fang Zhang, Jin-ge Yao, and Rui Yan. 2018a. On the abstractiveness of neural document summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 785–790.

Fangfang Zhang, Jin-ge Yao, and Rui Yan. 2018b. On the abstractiveness of neural document summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium. Association for Computational Linguistics.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

A Appendices

A.1 Proof of Proposition 1

According to the definition of extractive fragments (Grusky et al., 2018), a fragment must appear in one of the sentences in a source document, or there is no overlap between the source document and the target summary for the fragment. Therefore, for each row in the matrix $\mathbf{A}$, it must contain at least one 1 so that the row corresponds to a valid fragment.

Then by setting x to the all-one vector $\mathbf{1}$, $\mathbf{A} \cdot \mathbf{1}$ is equivalent to count how many 1 in each row of $\mathbf{A}$. Since $\mathbf{A}$ must contain at least one 1, setting x to the all-one vector $\mathbf{1}$ satisfies all constraints and $\mathbf{1}$ is always a feasible solution to our ILP. Therefore, the optimization problem 1 is always feasible.

A.2 Implementation Details

Models are implemented by Pytorch framework (Paszke et al., 2019) and Huggingface transformers (Wolf et al., 2020). We initialized BART with "facebook/bart-large-cnn" for CNN/DM, and with "facebook/bart-large" for arXiv/PubMed. As for PEGASUS, we chose "google/pegasus-large" for both datasets. We trained the models using Adam optimization with a learning rate of $5e-5$, and weight decay of 0.01. The learning rate is updated using a polynomial decay schedule and warm-up steps is set to 500. We trained models with 10 epochs and used early stopping with patience 5. All experiments were performed on NVIDIA GeForce RTX 3090, and it took about 4 days for 1 epoch on both datasets. The hyperparameters in Equation 2 are set as $\lambda_1 = 5$, $\lambda_2 = 2$, $\lambda_3 = 1$. λ in Equation 5 is set to 0.4 in a range $[0.3, 0.7]$ based on performance on validation sets.

During inference, we used beam search with a beam size 4, length penalty 2 and non-repeat n-gram size is set to 3. For evaluation, we used ROUGE python package[3].

A.3 Case Study

We presented generated examples from our model and BART in Table 7. We highlighted the texts that BART copied from the source document in yellow, and we bolded the text that our model copied. By comparing to the target summary, we can observe that BART copied longer continuous spans and the continuous spans contain content that should not be covered in the target summary, such as *"MacLaren was announced as director of the movie in November"*. By contrast, in the output from out model, it does not cover those texts that shouldn't appear in the summary. However, both BART and our model still fail to generate the important content about *"MacLaren left the project over "creative differences""*. This indicates that there is still room for improvement in our model.

[3]https://github.com/google-research/google-research/tree/master/rouge

Source Document	(CNN)Wanted: film director, must be eager to shoot footage of golden lassos and invisible jets. **CNN confirms that Michelle MacLaren is leaving the upcoming "Wonder Woman" movie** (The Hollywood Reporter first broke the story). MacLaren was announced as director of the movie in November . CNN obtained a statement from Warner Bros. Pictures that says, "Given creative differences, Warner Bros. and Michelle MacLaren have decided not to move forward with plans to develop and direct 'Wonder Woman' together." (CNN and Warner Bros. Pictures are both owned by Time Warner.) **The movie**, starring Gal Gadot in the titlerole of the Amazon princess, **is still set for release on June 23, 2017.**It's the first theatrical movie centering around the most popular female superhero. Gadot will appear beforehand in "Batman v. Superman: Dawn of Justice," due out March 25, 2016. In the meantime, Warner will need to find someone new for the director's chair.
Target Summary	Michelle MacLaren is no longer set to direct the first "Wonder Woman" theatrical movie. MacLaren left the project over "creative differences". Movie is currently set for 2017.
BART Output	CNN confirms that Michelle MacLaren is leaving the upcoming "Wonder Woman" movie. MacLaren was announced as director of the movie in November. The movie, starring Gal Gadot in the title role of the Amazon princess, is set for release on June 23, 2017. It's the first movie centering around the most popular female superhero.
Our Output	CNN confirms that Michelle MacLaren is leaving the upcoming "Wonder Woman" movie. The movie is still set for release on June 23, 2017.

Table 7: Example outputs based on a test instance from CNN/DM

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section 7

☑ A2. Did you discuss any potential risks of your work?
Section 7

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract and Section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☒ Did you use or create scientific artifacts?

Left blank.

☐ B1. Did you cite the creators of artifacts you used?
No response.

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
No response.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
No response.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
No response.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
No response.

☐ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
No response.

C ☑ Did you run computational experiments?

Section 2-4

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Appendix Section A.2

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Appendix Section A.2

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 2-4

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Appendix Section A.2

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?
Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.