Enhancing Multi-Document Summarization with Cross-Document
Graph-based Information Extraction

Zixuan Zhang\textsuperscript{1}, Heba Elfardy\textsuperscript{2}, Markus Dreyer\textsuperscript{2}, Kevin Small\textsuperscript{2}, Heng Ji\textsuperscript{2}, Mohit Bansal\textsuperscript{2}

\textsuperscript{1}University of Illinois at Urbana-Champaign
\textsuperscript{2}Amazon Alexa

zixuan11@illinois.edu
{helfardy, mddreyer, smakevin, jihj, mobansal}@amazon.com

Abstract

Information extraction (IE) and summarization are closely related, both tasked with presenting a subset of the information contained in a natural language text. However, while IE extracts structural representations, summarization aims to abstract the most salient information into a generated text summary – thus potentially encountering the technical limitations of current text generation methods (e.g., hallucination). To mitigate this risk, this work uses structured IE graphs to enhance the abstractive summarization task. Specifically, we focus on improving Multi-Document Summarization (MDS) performance by using cross-document IE output, incorporating two novel components: (1) the use of auxiliary entity and event recognition systems to focus the summary generation model and; (2) incorporating an alignment loss between IE nodes and their text spans to reduce inconsistencies between the IE graphs and text representations. Operationally, both the IE nodes and corresponding text spans are projected into the same embedding space and pairwise distance is minimized. Experimental results on multiple MDS benchmarks show that summaries generated by our model are more factually consistent with the source documents than baseline models while maintaining the same level of abstractiveness.\textsuperscript{1,2}

1 Introduction

Information extraction (IE) (Lin et al., 2020; Li et al., 2021) and summarization (Xiao et al., 2022; Pasunuru et al., 2021) are inherently similar tasks, sharing the objective of identifying and presenting a targeted subset of the information present in a natural language text. However, there are also conceptual and methodological distinctions. First of all, IE aims to extract specific structured information from natural language text while abstractive text summarization targets abstracting the most salient information of a given text into a natural language summary. Secondly, IE methods frequently have access to world knowledge via external schema and knowledge resources (e.g., Wikidata) whereas summarization methods often rely on the information encoded in large-scale pretrained embeddings to produce coherent summaries. The complementary aspects of these tasks imply an opportunity to transfer knowledge from one task to another.

Hence, in this paper, our primary motivation is to take advantage of the complementary nature of IE and summarization tasks, using the structured output of entity and event extraction systems to improve abstractive text summarization by focusing text generation toward explicitly observable grounded concepts. There are a few previous research studies exploring the mutual enhancement between IE and summarization. For example, Lu et al. (2022) use text summarization to improve relation extraction and Pasunuru et al. (2021) adopt open-domain IE to provide additional structural inputs for Multi-Document Summarization (MDS). However, these approaches have two notable limitations. First, IE is performed on single documents without analyzing cross-document interactions between the extracted knowledge elements. Such cross-document interactions could be essential to identifying salient parts of the source documents, which is especially useful for MDS. Moreover, previous studies use linearized graphs without actually constructing the graph as a whole, and hence failing to capture some global interactions between the extracted knowledge elements.

Based on these motivations, in this paper, we propose a text summarization model which is enhanced by IE. We focus on multi-document summarization (MDS) and improve the MDS model with cross-document IE graphs. Specifically, given a cluster of documents related to the same topic, we

\textsuperscript{1}All our code will be publicly available at \url{https://github.com/amazon-science/IESum}.
\textsuperscript{2}The work was done during the first author’s internship at Amazon Alexa.
Detroit Police Officer Dan Donakowski said the 16-year-old boy walked into the bank located in the 15,000 block of West 7 Mile around 2 p.m. and demanded money. Donakowski said the teen gave the teller a note and threatened to use a bomb if she didn’t fork over the cash. “The teller complied, gave him some money and as he attempted to leave, the teller hit the button for the doors that automatically lock and the suspect was trapped inside,” Donakowski said. Police say a 16-year-old, of Detroit, entered Chase Bank located on Seven Mile on Detroit’s east side about 2:30 p.m. Monday. He walked up to the counter and told the teller he was strapped with a bomb and to give him all the money. The teller did. The teen set off for the doorway. He opened the first set of doors into the causeway. The sidewalk was only steps away. He made it to the outermost set of doors, inches from the outside world. He’d make it no further.

After realizing they wouldn’t budge, he tried to retreat through the door he’d just passed. They wouldn’t budge either. police can arrive and take him safely into custody ...

Figure 1: An example of an extracted cross-document IE graph. There are two documents in the cluster and both of them describe a story where a boy failed to rob a bank. However, each document lists different details about the event. For example, the first document mentions that the bank teller hit the button to lock the boy inside; while the second document mentions that boy was trapped between two different doors of the bank. We highlight the unique nodes and edges in the document graph. The merged graph has a more comprehensive description of the story.

first use a cross-document fine-grained IE system to extract a cluster-level information graph, where each node could be an entity or an event trigger and each edge could be “event-event” temporal relations, “event-argument” links, or “entity-entity” relations. Each node in the graph is merged from separate documents according to entity and event coreference. After obtaining the cluster-level IE graph, we use an edge-conditioned graph attention network to encode the IE graph and to merge the graph information into the sequence-to-sequence summary generation pipeline. To better utilize the signals from IE, we further propose two novel training objectives. First, we propose an auxiliary task of entity and event recognition, where an additional classification module is incorporated to train a model to select the important entities and event triggers when performing summarization. The purpose of this auxiliary task is to help the model better recognize and remember the important events and entities which could be crucial for generating high-quality summaries. Second, we propose a graph and text alignment loss that minimizes the distance between IE graph nodes (e.g., nodes A and B in Figure 1) and their corresponding text segments (e.g., retreat and trapped) in a shared latent embedding space. Such an alignment loss can effectively incorporate IE graph information into the text representations and also mitigate the errors and inconsistencies caused by inevitable noise in the automatically extracted IE graphs. We conduct extensive experiments on multiple MDS benchmarks and show that our model outperforms several strong baselines both in terms of ROUGE scores as well as factual consistency metrics, all while maintaining the same level of abstractiveness. In summary, our main contributions are:

- We improve multi-document summarization (MDS) with cross-document IE graphs.
- We propose two novel training objectives to help the model better utilize the guidance from IE: (1) an entity and event recognition task loss and (2) a node-text alignment loss.
- Our proposed approach is proven effective by extensive experiments on multiple MDS benchmarks while achieving new state-of-the-art performance.

2 Problem Formulation

Our problem definition follows the typical formulation of abstractive multi-document summarization (MDS). Specifically, given a cluster of input documents \( D = \{D_1, D_2, \ldots, D_N\} \), we aim to build a model to generate a summary \( S \) of the document cluster. In this paper, we particularly focus on using IE to enhance summarization using the IE graph \( G \) merged from the individual graphs \( \{G_1, G_2, \ldots, G_N\} \) extracted from \( N \) documents.

2.1 Cross-Document Information Extraction

We first perform cross-document information extraction on each document cluster using a state-of-the-art entity extraction and disambiguation system ReFinED (Ayoola et al., 2022) and event extraction and tracking system RESIN-11 (Du et al., 2022). Specifically, we first extract individual entity men-
Figure 2: An overview of our IE-enhanced summarization pipeline. We first truncate and concatenate all documents in the cluster and feed them into the text encoder to obtain token representations. Meanwhile, we use a cross-document IE system to generate a cluster-level IE graph, and then use a GNN to get the node representations. During training, in addition to minimizing the distance between the generated and the reference summaries, we further use an entity and event recognition task and a node-text alignment loss to take advantage of the guidance from IE and improve MDS performance.

We merge all coreferenced entity and event nodes with their corresponding edges to form a cross-document IE graph. To further connect document-level IE results into a cross-document IE graph, we then perform cross-document entity and event coreference resolution.

We merge all coreferenced entities and event nodes as nodes from each document in the cluster. We then perform relation extraction, event argument role labeling, and event-event temporal relation extraction to add edges and to obtain a complete IE graph for each document. As shown in Figure 1, an example extracted event mention could be a “Transport” event triggered by “walked” with two event arguments “boy” and “Chase bank”, where all such events are connected to form a unified document-level IE graph. To further connect document-level IE results into a cross-document IE graph, we then perform cross-document entity and event coreference resolution.³

We merge all coreferenced entities and event nodes with their corresponding edges to form a cross-document IE graph. Specifically, if two nodes are labeled as the same entity or event, we merge these two nodes into a unified node and connect all related edges to it. It is worth noting that our framework does not rely on a specific IE system and/or schema. Hence any form of structured IE outputs will work with our proposed method.

**Notation** Each node \( v \in \mathcal{V} \) could be an entity or event trigger. We use \( E = \{ e_1, e_2, \ldots, e_{|E|} \} \) and \( T = \{ t_1, t_2, \ldots, t_{|T|} \} \) to denote the set of entities and event triggers respectively, where each \( e_i \) and \( t_i \) also act as a node in \( \mathcal{V} \). Accordingly, there are three types of edges in \( \mathcal{E} \) and we use \( p_{ij} \), \( q_{ij} \), and \( r_{ij} \) to represent the “event-event” temporal relations, “event-entity” argument roles, and “entity-entity” relations respectively. As shown in Figure 1, each blue node represents an event trigger (e.g., gave) while each brown node is an entity (e.g., bank), where the unique event triggers and entities are highlighted. The IE results include “entity-entity” relations connecting two different entities (e.g., \(<\text{button, door}>\), “event-argument” links connecting an event trigger and an entity mention (e.g., \(<\text{gave, teller}>\), and “event-event” temporal relations connecting two events (e.g., \(<\text{hit, retreat}>\).

### 3 Approach

In this paper, our main goal is to improve multidocument summarization (MDS) with the extracted cross-document IE graph. As illustrated in Figure 2, we first concatenate all documents in a cluster and feed this concatenated input into a Longformer encoder (Beltagy et al., 2020) that is capable of handling long text sequences. We also use the cross-document IE system to obtain a cluster-level IE graph, as shown in the example highlighted in Figure 1, and use a graph attention network to obtain the node representations. During training, in addition to the cross-entropy summary loss between the generated and the ground-truth summaries, we propose two additional novel training objectives: (1) an entity and event recognition task that makes the model aware of the locations of important events and entities; and (2) an alignment loss between the IE graph nodes and their corresponding text spans.

³The entity mentions extracted from ReFinED are merged according to the Wikidata IDs, while the event coreference resolution is done by a neural model (Lai et al., 2021).
to ensure that they are factually consistent in the latent space. We will go into details of our model design in the following sections.

### 3.1 Document Encoding

To handle the long input sequences, we use the encoder of the pre-trained PRIMERA (Xiao et al., 2022) model which is continually pre-trained from the Longformer-Encoder-Decoder (LED) model (Beltagy et al., 2020) to encode the documents and obtain the token representations \{w_1, w_2, \cdots \}. We truncate each document to the size of \(L_{\text{max}}/N\) (where \(N\) is the number of documents in the cluster), and concatenate all documents with a special token [doc-sep] to fit the maximum input length \(L_{\text{max}}\) of the LED model.\(^4\)

\[
\{w_1, w_2, \cdots \} = \text{Enc}(D_1, D_2, \cdots, D_N).
\]

Similar to the work of Xiao et al. (2022), we assign the global attention on the [doc-sep] tokens to make sure that the model is aware of the document boundaries and that it analyzes the relationships between the documents.

In addition to directly encoding the documents, we also use the cross-document IE system described in Section 2.1 to extract a cross-document IE graph \(G = \{V, E\}\). Similar to Zhang and Ji (2021), we use an edge-conditioned graph attention network to encode the entity nodes \(E\) and event nodes \(T\) respectively. The initial node representations of entities and events are computed by the average of the representations over all tokens in the entity mention or event trigger.

\[
e_i = \frac{1}{|T-T_{E}|} \sum_{j \in E} w_j, \quad t_i = \frac{1}{|T-T_{T}|} \sum_{j \in T} w_j,
\]

where \([E_T, E_S]\) and \([T_T, T_S]\) denote the entity and event trigger spans respectively. After initializing the node embeddings, the updated entity embeddings are computed as follows:

\[
e_i^{L+1} = e_i^L + \gamma \cdot \sum_{j \in N_i} \alpha_{ij} f_n(v_j^L).
\]

In this equation, \(f_n(\cdot)\) is a linear transformation layer and \(\gamma\) is a hyper-parameter controlling the level of neighborhood aggregation, where a larger \(\gamma\) means more information from the neighbors is incorporated when updating the node representations. The attention weights \(\alpha_{ij}\) are determined by the node pair and the type of the edge connecting the pair of nodes.

\[
\alpha_{ij} = \frac{\exp(\text{MLP}([v_i, r_{ij}, v_j]))}{\sum_{k=1}^{N_i} \exp(\text{MLP}([v_i, r_{kj}, v_j]))},
\]

where \(r_{ij}\) and \(r_{kj}\) are from a pre-initialized edge embedding matrix which could be optimized during training.\(^5\) The event trigger embeddings are computed in the same way as the entities do. We use the node representations from the final layer as the output node representations.

### 3.2 Summary Generation

We use the pre-trained LED decoder to generate the summaries based on both token and node representations. In addition to the original pre-trained cross-attention mechanism \(f_T(\cdot)\) for token representations, we include another similar cross-attention mechanism \(f_G(\cdot)\) for \(T_T\) in all decoder layers for the system to model the relationships between each node in the graph and each token in the generated text. We use the pre-trained weights for text cross-attention mechanism \(f_T(\cdot)\) and the graph cross-attention mechanism \(f_G(\cdot)\) is randomly initialized, where both of them are continually optimized during the downstream training. An illustration of the pipeline in each decoder layer is shown in Figure 3. Therefore, each summary \(S_i\) is generated in an auto-regressive manner using the LED decoder \(\text{Dec}(\cdot)\) with both text and graph cross-attention mechanism:

\[
S_i = \text{Dec}([\text{BOS}], \{w_1, w_2, \cdots \}, \{v_1, v_2, \cdots \}),
\]

where \([\text{BOS}]\) is the start token in transformer decoders. Given a set of reference summaries

\(^4\)The maximum length \(L_{\text{max}}\) is set as 4096 in pre-trained PRIMERA and LED-large models.

\(^5\)We only consider three edge types here: event-event temporal relations, event-entity argument relations, and entity-entity relations.
\( \hat{S}_1, \cdots, \hat{S}_N \) and a set of generated summaries \( S_1, \cdots, S_N \), the summary loss is defined to minimize the cross-entropy distance \( f_{CE}(\cdot) \) between each pair of summary sequences.

\[
\mathcal{L}_{\text{summ}} = \frac{1}{N} \sum_{i=1}^{N} f_{CE} \left( S_i, \hat{S}_i \right). \tag{1}
\]

### 3.3 Entity and Event Recognition

The main goal of our model is to use IE results to enhance the performance of the summarization task. We first add an auxiliary entity and event recognition task to make the model more sensitive to the locations of important events and entities. This will ensure that the model will not miss these events and entities when summarizing the document. Specifically, we use a Multi-Layer Perceptron (MLP) based classifier to classify each token by

\[
p_i = \text{softmax} \left( \text{MLP} \left( w_i \right) \right). \tag{2}
\]

Given a set of input tokens \( w_1, w_2, \cdots, w_M \), the entity and event recognition loss is computed as:

\[
\mathcal{L}_{\text{recognition}} = - \sum_{i=1}^{M} p_{ij},
\]

where \( j \) is the index of the correct label for \( w_i \).

### 3.4 Node and Text Alignment

Incorporating graph information for summarization could be challenging, since the IE graphs are extracted from automatic extraction systems which may introduce noise and errors. To this end, we propose a novel alignment loss to minimize the distance between node representations and their corresponding texts to ensure coordination between the graphs and summarization text. Specifically, we first use two MLPs to map the node and text

\[
z_i^w = \text{MLP}_w \left( w_i \right), \quad z_i^v = \text{MLP}_v \left( v_i \right),
\]

where \( z_i^w \) and \( z_i^v \) denote the representations for token \( w_i \) and node \( v_i \) in the shared embedding space. Given each node \( v_i \) and the set of its corresponding text tokens \( W_i \), we minimize the cosine similarity between the node embedding \( v_i \) and the average embedding of all tokens in \( W_i \):

\[
\mathcal{L}_{\text{align}} = \sum_{w_i \in W_i} d_{\cos} \left( z_i^v, \frac{1}{|W_i|} \sum_{w_j \in W_i} z_j^w \right). \tag{3}
\]

The intuition behind \( \mathcal{L}_{\text{align}} \) is to ensure that the node embedding is centered around its corresponding text. This helps ensure that the graphs and input text are factually consistent with each other, thereby reducing the errors and noise propagated from the IE system. As an example in Figure 1, the latent distance between each pair of nodes and texts (e.g., the node representation of \textit{boy} and the text representation of its corresponding tokens \textit{16-year-old boy}) are minimized to reduce the noise of the extracted graph.

**Multi-Task Training.** We conduct multi-task training where the total loss is a weighted sum from Equation (1), (2), and (3). The weighting coefficients \( \beta_1, \beta_2, \) and \( \beta_3 \) are tunable hyper-parameters.

\[
\mathcal{L} = \beta_1 \cdot \mathcal{L}_{\text{summ}} + \beta_2 \cdot \mathcal{L}_{\text{recognition}} + \beta_3 \cdot \mathcal{L}_{\text{align}}
\]

### 4 Experiments

#### 4.1 Data

Our experiments are conducted on three most widely-used MDS benchmarks, where the detailed dataset statistics are shown in Table 1.

<table>
<thead>
<tr>
<th>Dataset</th>
<th># Train / Val / Test</th>
<th>Docs per Cluster</th>
<th>Average Summary Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-News</td>
<td>44972 / 5622 / 5622</td>
<td>2.8</td>
<td>217</td>
</tr>
<tr>
<td>WCEP-10</td>
<td>8158 / 1020 / 1022</td>
<td>9.1</td>
<td>28</td>
</tr>
<tr>
<td>DUC-2004</td>
<td>0 / 0 / 50</td>
<td>10</td>
<td>115</td>
</tr>
</tbody>
</table>

**Table 1: Statistics of the MDS Datasets**

**Multi-News.** The Multi-News benchmark (Fabbri et al., 2019) is the most widely-used dataset for multi-document summarization. The summaries are long and informative news abstracts written by human editors, and the documents are extracted from multifarious news articles.

**WCEP-10.** The WCEP-10 (Gholipour Ghalandari et al., 2020) dataset is extracted from Wikipedia Current Event Portal, where each document cluster also describes a news event. Compared to Multi-News, the WCEP dataset has a much...
larger number of documents in each cluster, and we manually reduce them to a maximum of 10 documents per cluster as previous research (Xiao et al., 2022; Parnell et al., 2022) did to obtain the WCEP-10 version of dataset. We include both Multi-News and WCEP-10 in our experiments to evaluate whether our model can stay effective in both long-summary and short-summary scenarios.

**DUC-2004.** There are only 50 test document clusters in DUC-2004 benchmark, and we use this dataset to evaluate our model’s zero-shot transfer ability. We train our model on Multi-News and directly test it on DUC-2004 since these two datasets have similar lengths of summaries.

### 4.2 Baselines and Implementation Details

For baselines, we mainly compare our model with state-of-the-art multi-document summarization models PRIMERA (Xiao et al., 2022) and REFLECT (Song et al., 2022). REFLECT only reports ROUGE scores on the Multi-News dataset and we directly use the reported scores for comparison. Besides, we also include a previous model BART-Graph (Pasunuru et al., 2021), which uses a linearized IE graph to improve summarization. We compare our model with it to see whether encoding the graph structurally improves the summarization performance. We also experiment with three ablation variants of our proposed model: (1) **Recognition-Only:** for the model with only the entity and event recognition loss; (2) **Alignment-Only:** for the model with only the graph encoder and the node-text alignment loss. (3) **Separate-Graphs:** for encoding the IE graphs for each document separately and using a collated matrix as the node representations. For Multi-News and WCEP-10, we train all of these models on the training set, choose the best model checkpoint based on the performance on the validation set, and test the models on the test set. For DUC-2004, we use the trained checkpoint on Multi-News dataset for evaluation, since the summary length on Multi-News is more similar to DUC-2004 compared with WCEP-10.\(^7\)

### 4.3 Evaluation Metrics

**Co-occurrence.** Similar to previous research studies, we first include the most widely-used **ROUGE-F1** score which measures the overlap between the generated summaries and the reference summaries in terms of overlapping n-grams and longest common subsequence.

**Factual Consistency.** Intuitively, our proposed IE-enhanced summarization should improve factual consistency of the generated summary with the source documents, since the entities and events in the original documents are mined and memorized by the model through the two proposed IE enhancement loss. Therefore, we include several factuality metrics to measure the improvements in terms of factuality of the generated summaries. Specifically we use **FactCC** (Kryscinski et al., 2020), **FactGraph** (Ribeiro et al., 2022), **EntityPrecision** (Nan et al., 2021), **SUMMAC** (Laban et al., 2022), and **BERTSCORE** (Pagnoni et al., 2021).

**Abstractiveness.** To measure abstractiveness of our generated summaries, we use the **MINT** score (Dreyer et al., 2023), which is based on contiguous and non-contiguous extractive overlaps between summaries and their source documents. Our goal is to measure whether the novelty of the generated summary is sacrificed due to the improvements of factual consistency, e.g., by generating a more extractive summary.

### 4.4 Results

Table 2 shows the results of our proposed model, as well as the baselines on the three datasets. In general, the full version of our proposed model outperforms the baselines in terms of both ROUGE scores and factuality metrics while maintaining the same level of MINT scores. This shows that our model can generate high-quality summaries that are factually consistent without sacrificing any novelty. Specifically, entity and event recognition mainly improve factual consistency, while node-text alignment improves the similarity with the referenced summaries. This follows our intuition since the recognition task is mainly designed to help the model better notice the important event triggers and entity mentions, which prevents the model from hallucination and thereby improves factual consistency. On the other hand, the alignment loss can reduce the noise and errors in those extracted IE graphs, which makes the model better optimized on the ground-truth summaries.

### 4.5 Human Evaluation

We conduct a human evaluation on Amazon Mechanical Turk to evaluate the effect of adding our

\(^6\)https://duc.nist.gov/duc2004/

\(^7\)More detailed hyper-parameter settings can be found in Appendix A.
Table 2: Evaluation results with various metrics on the three MDS datasets. We primarily compare our results with three most recent transformer-based baselines BART-Graph, REFLECT, and PRIMERA. We also include two variants of our own model for ablation study, where we remove the recognition loss and the alignment loss respectively and test the model on these MDS datasets.

4.6 Qualitative Analysis

To better understand the effects made by our proposed training objectives, we look into the prediction results and show a typical example in Figure 4, explaining how our proposed method works to improve the summaries. In this example, the document cluster is mainly talking about a shut-down incident of the Nasdaq trading market. Compared to the summary from the baseline model, our model is better at memorizing the important facts and showing them in the output summary, e.g., the exact Nasdaq Index (3631.17) when the trading was suddenly suspended. Some other facts such as “three hours” are also memorized by our model but ignored by the baseline model. Moreover, our model is able to generate more informative mentions of those key entities (e.g., NYSE), where the baseline model fails to generate a named mention and only writes "the exchange".

Table 3: Human evaluation results.
SAN FRANCISCO (MarketWatch) — Trading in all Nasdaq-listed stocks and options was halted on Thursday due to technical problems on the bourse, according to Nasdaq OMX Group (NASDAQ:NDAQ).... In response, the New York Stock Exchange has also stopped trading in all Nasdaq securities at the request of Nasdaq OMX. "All orders in those securities have been cancelled back to customers," said NYSE in a statement. The Nasdaq Composite Index (NASDAQ:COMP) was last at 3631.17, up 31.38 points, before trading was suspended. There was no immediate word on when transactions will resume. Credit: MarketWatch

Figure 4: A qualitative example from our full model compared to the baseline PRIMERA model. Our model is better at preserving important facts and utilizing more informative mentions of the key entities.

### 5 Related Work

**Multi-Document Summarization.** Abstractive multi-document summarization (MDS) aims to build models to generate summaries given a set of similar documents related to the same topic. With the tremendous success of sequence-to-sequence pre-trained language models such as BART (Lewis et al., 2020) and T5 (Guo et al., 2022), finetuning on pre-trained models, like DeYoung et al. (2021); Parnell et al. (2022); Zhao et al. (2022); Moro et al. (2022); Song et al. (2022); Ernst et al. (2022), has become the primary style of methods for summarization tasks. There are also research studies on how to handle cross-document information overlap and redundancy. For example, Pasunuru et al. (2021) propose to use graph structures generated by OpenIE systems to make the model more sensitive about the main message of the document cluster. More recently, Xiao et al. (2022) propose to integrate entity overlap into the pre-training scheme, where the overlapping entities are used to select out salient sentences for pre-training.

**Cross-Document Information Extraction.** Information Extraction (IE) aims to extract structured representations from unstructured text, which includes various subtasks from Named Entity Recognition (Reich et al., 2022; Ayoola et al., 2022; Ding et al., 2021), to Relation Extraction (Yu et al., 2022; Tian et al., 2022), and Event Extraction (Xu et al., 2021; Yu et al., 2021) on news documents. There are also a number of research studies (Yao et al., 2021; Wu et al., 2022; Du et al., 2022) focusing on corpus-level cross-document extraction models. However, all these models still rely on cross-document entity and event coreference systems, which could bottleneck the efficiency and effectiveness of corpus-level IE models.

**Joint IE and Summarization.** IE and summarization share inherent similarities; both of them are designed to find the main information from an input natural language text. Therefore, it is promising to design a joint learning framework so that the two tasks could provide each other with mutual enhancement. There are some preliminary explorations of previous studies to train a model to learn IE and natural language generation (NLG) tasks jointly. For example, Li et al. (2021) train a template-based generative model for event argument extraction, and Du and Cardie (2020) propose to generate natural questions to ask the model for event extraction. However, although generation-based methods are proposed, these models are still doing a single task (IE) without multi-task settings for both IE and NLG. Recently, Lu et al. (2022) use summarization to provide indirect training signal for relation extraction tasks, however, their method is only suitable for relation extraction tasks and cannot cover general-concept IE tasks.

### 6 Conclusions

In this paper, we focus on improving multi-document summarization (MDS) model with cross-document Information Extraction (IE). We propose two novel training objectives — an entity and event recognition loss and a node-text alignment loss — that can help the model better utilize the signals from IE. Experimental results show that our model can generate summaries that are more factual, while not losing any abstractiveness.
7 Limitations

One limitation of our proposed method is the IE graphs are pre-extracted separately, where the IE model is not optimized during the model training and the IE results are only used as side inputs for summarization. It would be more exciting if we can really build a joint IE and Summarization model which are trained simultaneously in the pipeline, although it is very difficult since passing the gradients through a cross-document system is nearly intractable. We intend to address this limitation in our future work.

Acknowledgement

We thank the anonymous reviewers for their valuable feedback.

References


A Experiment Details

We list our detailed hyper-parameter settings for training our model on each of the datasets in Table 4 and Table 5, where each hyper-parameter is determined based on grid search among 5 candidate values. We train our model on 8 NVIDIA V100 GPUs with 32GB memory, and the total training time is about 7 hours for Multi-News and 3 hours for WCEP-10.

<table>
<thead>
<tr>
<th>Hyper-parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Num of features for each node</td>
<td>1,024</td>
</tr>
<tr>
<td>Num of GNN layers</td>
<td>1</td>
</tr>
<tr>
<td>Message Passing Level $\gamma$</td>
<td>0.01</td>
</tr>
<tr>
<td>Weights of the losses $\beta_1, \beta_2, \beta_3$</td>
<td>1.0, 0.2, 0.2</td>
</tr>
<tr>
<td>Learning Rate</td>
<td>3e-5</td>
</tr>
<tr>
<td>Batch Size</td>
<td>16</td>
</tr>
<tr>
<td>Maximum Length of Generated Summaries</td>
<td>256</td>
</tr>
<tr>
<td>Maximum Training Steps</td>
<td>25,000</td>
</tr>
<tr>
<td>Warm-up Steps</td>
<td>2,500</td>
</tr>
<tr>
<td>Beam Size for Generation</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 4: Detailed hyper-parameter settings for model training on Multi-News.

<table>
<thead>
<tr>
<th>Hyper-parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Num of features for each node</td>
<td>1,024</td>
</tr>
<tr>
<td>Num of GNN layers</td>
<td>1</td>
</tr>
<tr>
<td>Message Passing Level $\gamma$</td>
<td>0.005</td>
</tr>
<tr>
<td>Weights of the losses $\beta_1, \beta_2, \beta_3$</td>
<td>1.0, 0.1, 0.1</td>
</tr>
<tr>
<td>Learning Rate</td>
<td>3e-5</td>
</tr>
<tr>
<td>Batch Size</td>
<td>16</td>
</tr>
<tr>
<td>Maximum Length of Generated Summaries</td>
<td>50</td>
</tr>
<tr>
<td>Maximum Training Steps</td>
<td>5,000</td>
</tr>
<tr>
<td>Warm-up Steps</td>
<td>500</td>
</tr>
<tr>
<td>Beam Size for Generation</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 5: Detailed hyper-parameter settings for model training on WCEP-10.

B Annotation Guidelines

We use Amazon MTurk to do human evaluation, where the detailed annotation guidelines for human evaluators are shown in Figure 5.
Figure 5: Annotation instructions to annotate factual consistency on Mechanical Turk.