Knowledge Graph-Augmented Abstractive Summarization with Semantic-Driven Cloze Reward

Luyang Huang[1] **Lingfei Wu**[2] and **Lu Wang**[1]
[1]Khoury College of Computer Sciences, Northeastern University, Boston, MA 02115
[2]IBM Research AI, IBM T.J. Watson Research Center, Yorktown Heights, NY 10598
[1]luyang.huang96@gmail.com, luwang@ccs.neu.edu
[2]wuli@us.ibm.com

Abstract

Sequence-to-sequence models for abstractive summarization have been studied extensively, yet the generated summaries commonly suffer from fabricated content, and are often found to be near-extractive. We argue that, to address these issues, the summarizer should acquire semantic interpretation over input, e.g., via structured representation, to allow the generation of more informative summaries. In this paper, we present **ASGARD**, a novel framework for Abstractive Summarization with Graph-Augmentation and semantic-driven RewarD. We propose the use of *dual encoders*—a sequential document encoder and a graph-structured encoder—to maintain the global context and local characteristics of entities, complementing each other. We further design *a reward based on a multiple choice cloze test* to drive the model to better capture entity interactions. Results show that our models produce significantly higher ROUGE scores than a variant without knowledge graph as input on both New York Times and CNN/Daily Mail datasets. We also obtain better or comparable performance compared to systems that are fine-tuned from large pretrained language models. Human judges further rate our model outputs as more informative and containing fewer unfaithful errors.

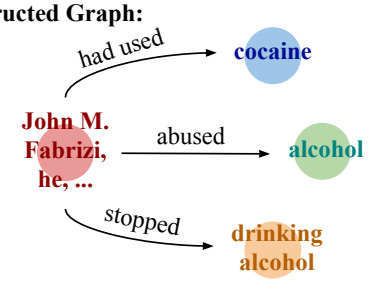
1 Introduction

Abstractive summarization aims to produce concise and informative summaries with the goal of promoting efficient information consumption and knowledge acquisition (Luhn, 1958). Significant progress has been made in this area by designing sequence-to-sequence-based neural models for single-document abstractive summarization (Gehrmann et al., 2018; Liu et al., 2018; Liu and Lapata, 2019). However, due to the limitations of model structure and word prediction-based learning objectives, these models frequently produce unfaithful content (Cao et al., 2018) and near-extractive summaries (See et al., 2017; Kryściński et al., 2018). These observations suggest that existing models lack semantic interpretation over the input, which is critical for summarization.

We argue that the generation of informative and succinct abstracts requires structured representation to facilitate the connection of relevant subjects, and the preservation of global context, e.g. entity interactions and topic flows. Take Fig. 1 as an ex-

Input Article of New York Times:
John M. Fabrizi, the mayor of Bridgeport, admitted on Tuesday that **he** had used **cocaine** and abused **alcohol** while in office.
Mr. Fabrizi, who was appointed mayor in 2003 after the former mayor, Joseph P. Ganim, went to prison on corruption charges, said **he** had sought help for his drug problem about 18 months ago and that **he** had not used drugs since.
About four months ago, **he** added, **he** stopped **drinking alcohol**.

Constructed Graph:



Summary by Human:
The Week column. **Mayor John Fabrizi** of Brigeport, Conn, publicly admits **he** used **cocaine** and abused **alcohol** while in office; says **he** stopped **drinking alcohol** and sought help for his drug problem about 18 months ago.

Figure 1: Sample knowledge graph constructed from an article snippet. The graph localizes relevant information for entities (color coded, e.g. "*John M. Fabrizi*") or events (underlined) and provides global context.

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 5094–5107
July 5 - 10, 2020.

ample. Complex events related with the same entity may span multiple sentences, making it challenging for existing sequential models to capture. A graph representation, on the contrary, produces a structured summary and highlights the proximity of relevant concepts.

To this end, we present **ASGARD**, a framework for Abstractive Summarization with Graph-Augmentation and semantic-driven RewarD.[1] Under the encoder-decoder framework, *we enhance the regular document encoder with a separate graph-structured encoder to maintain the global context and local characteristics of entities* by using the outputs from an open information extraction (OpenIE) system.

Specifically, we experiment with two graph variants, one mainly capturing entities' document-level interactions and the other reflecting such interactions within each paragraph plus topic shifts across paragraphs. Both graphs can capture interactions among entities that are positioned far from one another in the document and significantly reduce redundancy, as shown in Fig. 1. The document encoder and the graph encoder then cooperate during abstract generation, wherein the model is trained to identify salient content by aligning graphs with human summaries. Though structured representation has been studied before for summarization (Fernandes et al., 2019), to the best of our knowledge, we are the first to utilize graph neural networks to *explicitly* encode entity-centered information for abstractive summary generation.

Moreover, we propose *a novel multi-choice cloze reward to drive the model to acquire semantic understanding over the input*. Concretely, we design cloze questions by removing *pairwise* entities that are connected with a predicate or co-occur in a human summary sentence, whereas prior work only considers single entities to construct questions (Eyal et al., 2019). In tandem with our graph encoding of knowledge, the cloze reward further facilitates the acquisition of global entity interactions with reinforcement learning.

We carry out automatic and human evaluations on popular summarization datasets. Models based on ASGARD yield significantly better ROUGE scores (Lin and Hovy, 2003) than a variant without access to the knowledge graph on two popular news summarization datasets, New York Times corpus and CNN/Daily Mail dataset. Moreover, ASGARD models attain performance better than or comparable to others that are fine-tuned from large pretrained language models, including BERTSum (Liu and Lapata, 2019), UniLM (Dong et al., 2019), and BART (Lewis et al., 2019). Human judges further confirm that our models generate more informative summaries with less unfaithful errors than their counterparts without the graph encoder. Importantly, we find that automatic evaluation metrics only weakly correlate with these errors, implying that new evaluation methods are needed to better gauge summary quality.

The rest of the paper is organized as follows. We describe related work in the next section (§ 2). We then discuss the knowledge graph construction in § 3 and formulate our graph-augmented summarization framework in § 4. In § 5, we introduce reinforcement learning with cloze reward. Experiments and results are presented in § 6 and § 7. Finally, we conclude in § 8.

2 Related Work

Graph-Augmented Summarization and Generation. Graph structures have long been used for extractive summarization, such as in Textrank (Mihalcea and Tarau, 2004) and Lexrank (Erkan and Radev, 2004). For neural models, Tan et al. (2017) design graph-based attention to identify important sentences. For generating abstractive summaries, Fernandes et al. (2019) enhance a sequence-based encoder with graph neural networks (GNNs) to consider token-level entity types, however, entity interactions are largely ignored. On multi-document summarization, Fan et al. (2019) demonstrate the usefulness of encoding a linearized knowledge graph from OpenIE outputs. In this work, we design a graph encoder, which improves upon Graph Attention Networks (GATs) (Veličković et al., 2018), to capture the global context in a more effective manner.

Also related is the graph-to-sequence framework that has been adopted for text generation (Song et al., 2018). Both Gated Graph Neural Networks (GGNNs) (Beck et al., 2018) and Graph Convolutional Networks (GCNs) (Damonte and Cohen, 2019) are shown to be effective in generating sentences from AMR graphs. Since Graph Attention Networks can better handle sparse graphs, they are used by Koncel-Kedziorski et al. (2019) with a transformer model to create scientific paper ab-

[1] Our code is available at https://github.com/luyang-huang96/GraphAugmentedSum.

stracts from knowledge graphs. Here we use graphs *in addition to* document encoder, both carrying complementary information for summarization.

Reinforcement Learning and QA Reward for Abstractive Summarization. As pointed out by Ranzato et al. (2016), word-level maximum likelihood training brings the problem of exposure bias. Recent work utilizes reinforcement learning to directly optimize the model to maximize the informativeness of summaries by using different forms of ROUGE scores (Paulus et al., 2018; Chen and Bansal, 2018; Sharma et al., 2019). However, ROUGE does not always distinguish good summaries from bad ones (Novikova et al., 2017), and ignores entity interactions.

Since question answering (QA) has been used for summary evaluation (Narayan et al., 2018), and is shown to correlate with human judgment of summaries qualities (Eyal et al., 2019), QA-based rewards have been studied for summarization model training. Arumae and Liu (2019) demonstrate that using fill-in-the-blank questions by removing entities or root words leads to improved content selection. Scialom et al. (2019) consider a similar setup, but use both F1 score and QA system confidence as rewards in abstractive summarization. Previous work, however, mainly focuses on single entities or words in human-written summaries, thereby losing contexts and relations. Moreover, fill-in-the-blank questions by prior work give credits only when the answers exactly match the ground-truths, thus causing inaccuracies for rephrased answers and discouraging abstract content generation. In contrast, we design a semantic-driven cloze reward by measuring how well a QA system can address *multiple choice* cloze questions which better *encode entity interactions* and *handle paraphrased answers*.

3 Knowledge Graph Construction

To construct a knowledge graph from an input document, we utilize Stanford CoreNLP (Manning et al., 2014) to first obtain outputs from coreference resolution and open information extraction (OpenIE) models (Angeli et al., 2015). Note that we do not conduct global entity linking across documents. Next, we take the ⟨subject, predicate, object⟩ triples extracted by OpenIE and remove any triple whose argument (subject or object) has more than 10 words. If two triples differ only by one argument, and the arguments overlap, we keep the longer triple.

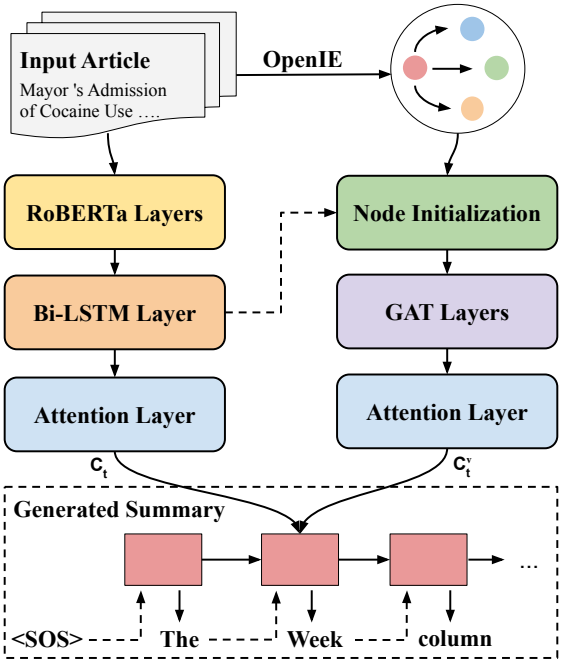


Figure 2: Our ASGARD framework with document-level graph encoding. Summary is generated by attending to both the graph and the input document.

We begin constructing the graph by treating subjects and objects as nodes connected by directed edges, with predicates as attributes. We further collapse coreferential mentions of the same entity into one node. With this, we can localize salient content related to each entity as well as make connections of spread-out entities through graph paths.

4 Summarization Model

In this section, we describe our graph-augmented abstractive summarization framework, as displayed in Fig. 2. Our model takes as input a document, represented as a sequence of tokens $\mathbf{x} = \{x_k\}$, and a knowledge graph G consisting of nodes $\{v_i\}$. $\mathbf{x}$ and G are separately consumed by a document encoder and a graph encoder, as presented in § 4.1. Importantly, we present two types of graphs: DOCGRAPH, focusing on the global context, and SEGGRAPH, which additionally captures topic shift. The summary decoder then generates an abstractive summary by attending to both the document and the graph (§ 4.2). In § 4.3, we formulate a maximum likelihood training objective which leverages the detection of salient nodes in the graph.

4.1 Encoders

Document Encoder. We first feed input $\mathbf{x}$ to RoBERTa (Liu et al., 2019) and take the last layer

output as token embeddings. We then employ a single-layer bidirectional LSTM (BiLSTM) over token embeddings, producing encoder hidden states $\mathbf{h}_k$ at time step k.

Graph Encoder. Built on the graph constructed in § 3, we create nodes for predicates as done in previous graph-to-sequence work (Beck et al., 2018) to reduce model parameters. Directed, unlabeled edges are added from subject to predicate, and from predicate to object. We further add reverse edges and self-loops to enhance the information flow, and this forms the graph G.

Node Initialization. Each node often contains multiple mentions of an entity; we thus initialize node representation $\mathbf{v}_i$ by using the average embedding of its tokens. We leverage document encoder hidden states $\mathbf{h}_k$ as the contextual representation of tokens. Number of mentions in the node is added as an extra encoding to $\mathbf{v}_i$, to signify entity salience.

Contextualized Node Encoding. Our graph encoder improves upon Graph Attention Networks (GATs) (Veličković et al., 2018) by adding residual connections between layers as discussed in Koncel-Kedziorski et al. (2019). Each node $\mathbf{v}_i$ is represented by a weighted average of its neighbors:

$$\hat{\mathbf{v}}_i = \mathbf{v}_i + \|_{n=1}^{N} \sum_{v_j \in \mathcal{N}(v_i)} \alpha_{i,j}^n \mathbf{W}_{0,n} \mathbf{v}_j \quad (1)$$

$$\alpha_{i,j}^n = \text{softmax}((\mathbf{W}_{1,n}\mathbf{v}_i)^T (\mathbf{W}_{2,n}\mathbf{v}_j)) \quad (2)$$

where $\|_{n=1}^{N}$ denotes the concatenation of N heads, each producing a vector of the same dimension as $\mathbf{v}_i$. We use $N = 4$ in our experiments with two layers of GATs. $\mathcal{N}(v_i)$ denotes the neighbors of v_i in graph G. $\mathbf{W}_*$ are trainable parameters.

The graph encoder described above encodes document-level global context by merging entity mentions throughout the document and capturing their interactions with graph paths. It is henceforth denoted as **DocGragh**.

Encoder Extension to Capture Topic Shift (SegGragh). Modeling topic transitions and recurrences enables the identification of notable content, thus benefiting summarization (Barzilay and Lee, 2004). Since paragraphs naturally divide a document into different topic segments, we extend DocGragh by first encoding each paragraph as a subgraph G_p (for the p-th paragraph) using the same graph encoder, and then connecting all subgraphs with a BiLSTM. If two nodes in separate subgraphs refer to the same entity, they are initialized with the same embedding (as in the first occurrence). Concretely, we first apply max-pooling over all nodes in subgraph G_p from the outputs of the final GAT layer; the max-pooling results are then used as inputs for a BiLSTM to produce the final subgraph representation $\mathbf{h}_p^g$ for G_p.

4.2 Summary Decoder

Our summary decoder uses a single-layer unidirectional LSTM with a hidden state $\mathbf{s}_t$ at step t; it generates summary tokens recurrently by jointly attending to the input document and the graph.

Attending the Graph. At each decoding step t, we compute a graph context vector $\mathbf{c}_t^v$ with the attention mechanism (Bahdanau et al., 2014):

$$\mathbf{c}_t^v = \sum_i a_{i,t}^v \hat{\mathbf{v}}_i \quad (3)$$

$$a_{i,t}^v = \text{softmax}(\mathbf{u}_0^T \tanh(\mathbf{W}_3\mathbf{s}_t + \mathbf{W}_4\hat{\mathbf{v}}_i)) \quad (4)$$

where $\mathbf{u}_*$ are also trainable parameters. We omit bias terms for simplicity.

Attending the Document. Similarly, the document context $\mathbf{c}_t$ is computed over input tokens by additionally considering the graph context $\mathbf{c}_t^v$:

$$\mathbf{c}_t = \sum_k a_{k,t}\mathbf{h}_k \quad (5)$$

$$a_{k,t} = \text{softmax}(\mathbf{u}_1^T \tanh(\mathbf{W}_5\mathbf{s}_t + \mathbf{W}_6\mathbf{h}_k + \mathbf{W}_7\mathbf{c}_t^v)) \quad (6)$$

Token Prediction. Graph and document context vectors, treated as salient content summarized from both sources, are concatenated with the decoder hidden state $\mathbf{s}_t$ to produce the vocabulary distribution P_{vocab}:

$$P_{vocab} = \text{softmax}(\mathbf{W}_{out}[\mathbf{s}_t|\mathbf{c}_t|\mathbf{c}_t^v]) \quad (7)$$

We use *weight-sharing* between the input embedding matrix and the matrix $\mathbf{W}_{out}$ to allow reusing linguistic knowledge as proposed by Paulus et al. (2018). We further add a copy mechanism similar to See et al. (2017), with copy probability as:

$$P_{copy} = \sigma(\mathbf{W}_{copy}[\mathbf{s}_t|\mathbf{c}_t|\mathbf{c}_t^v|\mathbf{y}_{t-1}]) \quad (8)$$

where $\mathbf{y}_{t-1}$ denotes the embedding for the token predicted at step $t-1$.

Modified Hierarchical Attention for SegGraph. As mentioned in § 4.1, SegGraph captures content salience by modeling topic shift across paragraphs. We thus seek to leverage paragraph-level importance to redistribute the node attentions, e.g., giving

more attentions to nodes in important paragraphs. In particular, we utilize hierarchical attention (Hsu et al., 2018), where we first calculate attention $\mathbf{a}_t^g$ over subgraphs as done in Eq. 3 by replacing $\hat{\mathbf{v}}_i$ with subgraph representation $\mathbf{h}_p^g$.

We then combine subgraph attentions $\mathbf{a}_t^g$ with the previously calculated attentions $\mathbf{a}_t^v$ for nodes in the subgraph using scalar multiplication and renormalization over all nodes in input. This results in the new attention weights $\hat{\mathbf{a}}_t^v$, which are used to obtain graph context vector $\mathbf{c}_t^v$ as done in Eq. 3 for SegGraph.

4.3 Training Objectives

We first consider a maximum likelihood (ML) training objective that minimizes the following loss:

$$\mathcal{L}_{\text{seq}} = -\frac{1}{|D|} \sum_{(\mathbf{y},\mathbf{x})\in D} \log p(\mathbf{y} \mid \mathbf{x}; \theta) \tag{9}$$

where $\mathbf{x}$ are documents and $\mathbf{y}$ are references from the training set D, and θ are model parameters.

Node Salience Labeling. In addition to modeling local characteristics of nodes, we further enhance the model by adding an objective to label node salience, e.g., whether the entities in a node are mentioned in the reference summaries. We introduce a soft mask layer over each node before it is passed into the graph encoder, to signify its salience. This layer, serving as an information gate, predicts a real number m_i in $[0, 1]$ for each node $\mathbf{v}_i$ and multiplies to itself, i.e. $m_i\mathbf{v}_i$. For node $\mathbf{v}_i$, the mask is calculated as $\hat{m}_i = \text{sigmoid}(\mathbf{u}_2\mathbf{v}_i)$. During training, the gold-standard mask m_i for a node is set to 1 if it contains at least one content word in the reference summary; otherwise, 0. We add the following objective for all nodes in the dataset D:

$$\mathcal{L}_{mask} = -\frac{1}{N_v} \sum_{v_i\in D} m_i \log(\hat{m}_i) + (1 - m_i) \log(1 - \hat{m}_i) \tag{10}$$

where N_v represents the number of nodes in the dataset. Finally, the ML training objective takes the following form: $\mathcal{L}_{\text{ml}} = \mathcal{L}_{\text{mask}} + \mathcal{L}_{\text{seq}}$.

5 Reinforcement Learning with Cloze

After maximum likelihood training with $\mathcal{L}_{\text{ml}}$, we further design a *multiple choice cloze reward* in a second-stage reinforcement learning (RL), leading the model to generate more faithful and informative summaries.

For RL, we use a *self-critical policy gradient* algorithm (Rennie et al., 2017). During training, two summaries are generated: first, a summary $\mathbf{y}^s$, *sampling* tokens based on the probability distribution $p(\mathbf{y}^s \mid \mathbf{x}; \theta)$ at each decoding step; and second, a *baseline* summary $\hat{\mathbf{y}}$ which greedily selects the tokens of the highest probability at each step. The objective of RL is defined based on the rewards of the two summaries, $R(\mathbf{y}^s)$ and $R(\hat{\mathbf{y}})$, as follows:

$$\mathcal{L}_{\text{rl}} = -\frac{1}{|D|} \sum_{(\mathbf{y}^s,\mathbf{x})\in D} (R(\mathbf{y}^s) - R(\hat{\mathbf{y}})) \log p(\mathbf{y}^s \mid \mathbf{x}; \theta) \tag{11}$$

Our **reward** function uses the combination of ROUGE and the multiple choice cloze score introduced below, i.e., $R(\mathbf{y}) = R_{rouge}(\mathbf{y}) + \gamma_{cloze} R_{cloze}$. For ROUGE, it considers F1 scores of ROUGE-1, ROUGE-2, and ROUGE-L calculated against the reference summary, and takes the form of $R_{rouge}(\mathbf{y}) = \gamma_1 R_{rouge-1}(\mathbf{y}) + \gamma_2 R_{rouge-2}(\mathbf{y}) + (1 - \gamma_1 - \gamma_2) R_{rouge-L}(\mathbf{y})$.

Multiple Choice Cloze Reward. Here, we present a novel multiple choice cloze reward to work with our knowledge graph and guide the summarization model towards improved awareness of entity interactions. We treat the system-generated summary as **context**. We provide a set of **questions** automatically constructed from the corresponding reference summary written by a human. We separately train a question answering (QA) model to address the questions by reading the context. Intuitively, if the system summary shares salient information with the reference, the QA model will assign the correct answers with high probability. We decide to use the average probability of the correct answers as our **cloze reward**. Below, we give details on how to construct the questions and candidate answers with examples shown in Fig. 3.

Question Construction. We run the OpenIE tool on human-written summaries, retaining triples with arguments not longer than 5 words. For each triple of ⟨subject, predicate, object⟩, we create two types of questions: (1) **argument pair questions**, by removing the subject and object, and (2) **predicate questions**, by removing the predicate.

Candidate Answer Construction. Because fill-in-the-blank style cloze may incorrectly penalize QA systems with answers paraphrased from the ground-truth, we opt for a multiple choice cloze. We construct three **candidate answers** in addition to the

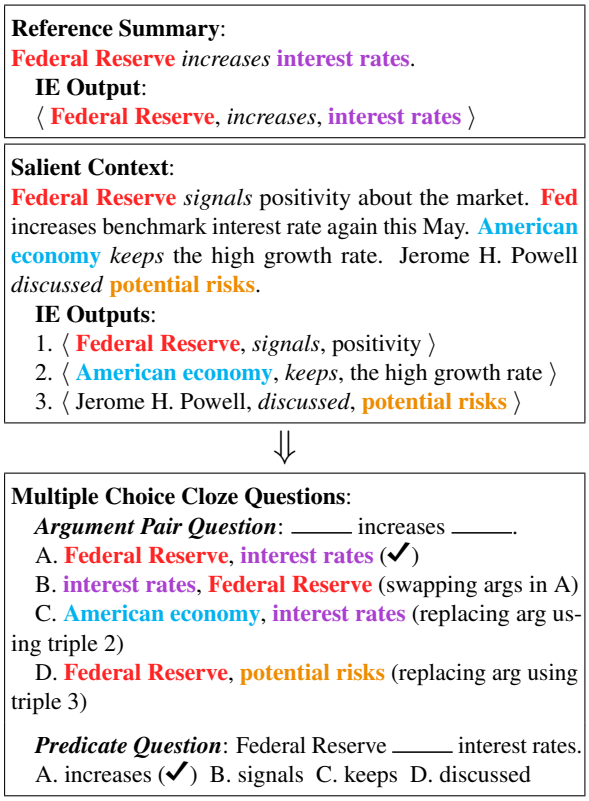


Figure 3: Sample construction of multiple choice cloze questions and candidate answers from reference summary and salient context. Arguments and predicates in candidate answers are color-coded and italicized.

gold-standard from the **salient context**, which are summary-worthy sentences selected from the input. Specifically, we use greedy search to select the best combination of sentences that maximizes ROUGE-2 F1 with reference to human summary. We further include a sentence in the salient context if it has a ROUGE-L recall greater than 0.6 when compared with any sentence in the reference.

We first select OpenIE triples from the salient context and filter out those that have any overlapping content word with the correct answer. For *argument pair questions*, we create one candidate answer by swapping the subject and the object (e.g. candidate B as in Fig. 3) and two candidates by replacing the subject or the object with another argument of the same role extracted from the salient context (e.g. candidates C and D). If not enough answers are created, we further consider randomly selecting sentences from the input. For *predicate questions*, we use predicates in other triples from the context as candidate answers. Among all candidates, we select the three that are able to construct the most fluent questions using perplexity predicted by BERT (Devlin et al., 2019).

In case reference summaries do not yield OpenIE triples, we create additional entity pair questions. We remove two co-occurring entities from the summary and create three candidate answers in the same way as described above.

QA Model. We fine-tune RoBERTa (Liu et al., 2019) to build our QA model. We use the salient context described above as the context for training. We then concatenate the context, the question, and each of the four candidate answers, and pass the final [CLS] representation through a fully-connected layer, from which the answer is predicted.

6 Experimental Setups

Datasets. We experiment with two popular summarization datasets with summaries containing multiple sentences: the New York Times annotated corpus (NYT) (Sandhaus, 2008) and the CNN/Daily Mail dataset (CNN/DM) (Hermann et al., 2015). We follow the preprocessing steps and experimental setups from prior work (Paulus et al., 2018; See et al., 2017) for both datasets. For NYT, the training, validation, and test sets contain $588,909$, $32,716$, and $32,703$ samples. For CNN/DM, the numbers are $287,188$, $13,367$, and $11,490$.

To train our cloze QA model for NYT, we construct $1,414,336$ question-answer pairs from human-written summaries in the training set based on the method described in § 5. On CNN/DM, we collect $1,361,175$ question-answer samples from the training set. For both datasets, we set aside $20,000$ samples as a validation set and $20,000$ samples as a test set. Our QA model achieves an accuracy of 97% on NYT and 95% on CNN.

Training Details and Parameters. We use the base version of RoBERTa model to extract token features for all experiments. We truncate input articles to 1024 (NYT) and 512 (CNN/DM) BPEs. We employ LSTM models with 256-dimensional hidden states for the document encoder (128 each direction) and the decoder. For the residual connection of the graph encoder, we use 4 heads, each with a dimension of 72. For DocGraph training and inference, we prune isolated graphs with fewer than three nodes to increase robustness and reduce redundancy. We set $\gamma_1 = 0$, $\gamma_2 = 0.75$ on NYT and $\gamma_1 = 0.33$, $\gamma_2 = 0.33$ on CNN/DM after tuning on the validation set. For both datasets, we set $\gamma_{cloze} = 0.05$. More details about parameters and graph statistics are in the Appendices.

Baselines and Comparisons. For both datasets,

System	ROUGE-1	ROUGE-2	ROUGE-L
LEAD-3	32.59	16.49	29.17
POINTGEN+COV	41.06	25.71	37.28
DEEPREINFORCE	47.03	30.72	43.10
BOTTOMUP	47.38	31.23	41.81
DCA	48.08	31.19	42.33
SENECA	47.94	31.77	44.34
BART	**53.25**	**36.61**	**48.78**
Our Models			
NOGRAPH	47.15	32.02	43.65
$+R_{rouge}$	49.17	33.19	46.44
ASGARD-DOC	49.51	33.82	45.72
$+R_{rouge}$	50.18	33.91	46.84
$+R_{rouge} + R_{cloze}$	50.59	33.98	48.24
ASGARD-SEG	49.54	33.84	45.75
$+R_{rouge}$	50.47	33.95	47.43
$+R_{rouge} + R_{cloze}$	*51.29*	*34.97*	*48.26*

Table 1: Automatic evaluation with ROUGE on New York Times. Best results are in **boldface**. Best of our models are in *italics*. ASGARD-SEG+R_{rouge}+R_{cloze} yields significantly higher scores than our other models with approximate randomization test ($p < 0.0005$).

System	ROUGE-1	ROUGE-2	ROUGE-L
LEAD-3	40.23	17.52	36.34
POINTGEN+COV	39.53	17.28	36.38
DEEPREINFORCE	41.16	15.75	39.08
BOTTOMUP	41.22	18.68	38.34
DCA	41.69	19.47	37.92
BERTSUMEXTABS	42.13	19.60	39.18
UNILM	43.33	20.21	40.51
BART	**44.16**	**21.28**	**40.90**
Our Models			
NOGRAPH	39.55	17.89	36.75
$+R_{rouge}$	41.37	17.63	37.99
ASGARD-DOC	40.38	18.40	37.51
$+R_{rouge}$	43.10	17.58	39.41
$+R_{rouge} + R_{cloze}$	*43.93*	*20.37*	*40.48*
ASGARD-SEG	40.09	18.30	37.30
$+R_{rouge}$	42.94	17.93	39.36
$+R_{rouge} + R_{cloze}$	43.81	20.22	40.37

Table 2: Automatic evaluation with ROUGE on CNN/Daily Mail. Best results of our model variants are in *italics*. Both ASGARD-SEG+R_{rouge}+R_{cloze} and ASGARD-DOC+R_{rouge}+R_{cloze} obtain significantly better scores than other model variants ($p < 0.0005$).

we include an extractive baseline LEAD-3. We further add the following abstractive models for comparison: (1) a pointer-generator model with coverage (See et al., 2017) (POINTGEN+COV); (2) a deep reinforcement learning-based model (Paulus et al., 2018) (DEEPREINFORCE); (3) a bottom-up model (Gehrmann et al., 2018) (BOTTOMUP); (4) a deep communicating agents-based summarization model (Celikyilmaz et al., 2018) (DCA). We also report results by fine-tuning BART model (Lewis et al., 2019). In Lewis et al. (2019), fine-tuning is only performed on CNN/Daily Mail. We apply the same method for NYT.

For NYT, we add results by SENECA model (Sharma et al., 2019) from our prior work, which previously achieved the best ROUGE-2.

On CNN/Daily Mail, we include comparisons of a two-stage fine-tuned model (first on an extractor, then on an abstractor) with BERT (Liu and Lapata, 2019) (BERTSUMEXTABS), and a unified pretrained language model for generation (Dong et al., 2019) (UNILM).

In addition to ASGARD-DOC and ASGARD-SEG, which are trained with an ML objective, we report results trained with ROUGE as the reward (R_{rouge}), and with an additional cloze reward (R_{cloze}). Lastly, we consider a variant NOGRAPH by ablating the graph encoder.

7 Results

7.1 Automatic Evaluation

Results on NYT. As displayed in Table 1, our ASGARD-SEG model trained with ROUGE and cloze rewards achieves better ROUGE scores (Lin and Hovy, 2003) than all other comparisons except the fine-tuned BART. However, our ASGARD-SEG's ROUGE-L score is comparable to BART. This indicates the effectiveness of our graph-augmented summarization framework.

Moreover, both our ASGARD-DOC and ASGARD-SEG models yield significantly higher ROUGE scores than the variant without the graph encoder (NOGRAPH). This demonstrates the benefit of using structured representation to encode entity interactions. Furthermore, both ASGARD-DOC and ASGARD-SEG with cloze reward (R_{cloze}) obtain significantly higher scores compared to the models trained with ROUGE reward only. This signifies that our multi-choice cloze reward can guide better semantic interpretation of content, leading to the generation of more informative summaries. We also find that ASGARD-SEG outperforms ASGARD-DOC, indicating that ASGARD-SEG better captures topic drift through multiple paragraphs.

Results on CNN/DM. We observe similar trends on the CNN/DM articles as shown in Table 2. No-

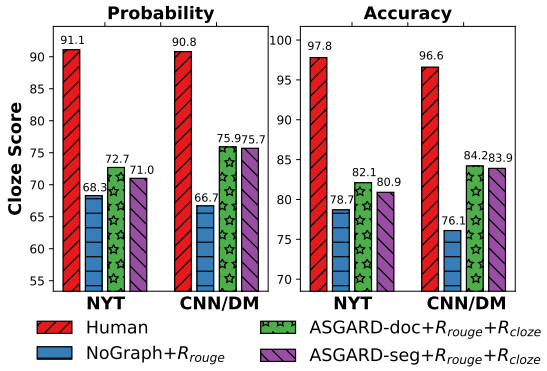


Figure 4: Evaluation with QA model prediction probability and accuracy on our multiple choice cloze test, with higher numbers indicating better summaries.

ticeably, ASGARD-DOC trained with the combined ROUGE and cloze reward produces better ROUGE scores than BERTSUMEXTABS and UNILM, which are carefully fine-tuned from large pretrained language models, and the numbers are also comparable to the fine-tuned BART.

Evaluation with Cloze Test. We further evaluate model-generated summaries with our proposed cloze test. Here, we report two scores in Fig. 4: the average **probability** of the correct answers output by our QA model, and its prediction **accuracy**. We first calculate one score per summary, then take the average over all summaries. We can see that our models with graph encoders perform better than the variant without it.

7.2 Human Evaluation

We further conduct human evaluation to analyze the informativeness and fluency of the generated summaries, as well as to investigate the unfaithful errors made by different models. We sample 100 articles from the NYT test set and hire three native or fluent speakers of English to rate summaries generated by our two systems, NOGRAPH+R_{rouge} and ASGARD-SEG+R_{rouge} + R_{cloze}, along with outputs by BART and human-written summaries (presented in random order). After reading the articles, each judge scores summaries on a Likert scale from 1 (worst) to 5 (best) on **informativeness**—whether the summary covers important information from the input, and **fluency**—whether the summary is grammatically correct.

We consider three types of unfaithful errors: (i) **hallucination error**—creating content not present in the input, (ii) **out-of-context error**—generating facts without including required context or within incorrect context, and (iii) **deletion or substitution error**—mistakenly deleting or substituting subjects, objects, or clauses. We ask the annotators to label each type as 1 for existence of errors, and 0 otherwise. Detailed guidelines are in the Appendices.

From Table 3, we can see that our ASGARD-SEG model obtains better scores in informativeness and fluency, compared to the variant without the graph encoder. This indicates the effectiveness of leveraging knowledge graph representation. Sample output summaries by our models can be found in Fig. 5. Meanwhile, fine-tuned BART model produces outputs with similar informativeness and fluency of human-constructed summaries, suggest-

System	Inf.↑	Flu.↑	Hal.↓	Out.↓	Del./Sub.↓
HUMAN	4.47	4.65	21%	10%	10%
NOGRAPH +R_{rouge}	3.94	3.65	**9%***	26%	22%
ASGARD-SEG +R_{rouge} + R_{cloze}	4.12†	3.77†	23%	**14%**†	**9%***
BART	**4.44***	**4.66***	16%	15%	12%

Table 3: Human evaluation on informativeness (Inf.) and fluency (Flu.) (1-to-5), and percentages of unfaithful errors of hallucination (Hal.), out-of-context (Out.) and deletion or substitution (Del./Sub.). $*$: significantly different from all other models. †: ASGARD-SEG is significantly better than NOGRAPH ($p < 0.05$). Inter-rater agreement with Krippendorf's α for all columns: 0.61, 0.70, 0.57, 0.50 and 0.43.

Summary by Human:
Family Court in Burlington County, NJ, rules that lesbian couple can list both their names as parents on birth certificate of newborn; **state attorney general's office drops opposition to move; court ruling negates couple's having to go through adoption proceedings to establish full parental rights for both.**

NoGraph+R_{rouge}:
Lesbian couple in South Jersey wins court approval to have both of their names listed as parents on birth certificate of their newborn. it will no longer oppose such applications

ASGARD-doc+R_{rouge} + R_{cloze}:
Lesbian couple in South Jersey, won court approval to have both of their names listed as parents on birth certificate of their newborn. **attorney general's office says it will no longer oppose such applications**

ASGARD-seg+R_{rouge} + R_{cloze}:
Lesbian couple in South Jersey wins court approval to have both of their names listed as parents on birth certificate of newborn **and attorney general 's office will no longer oppose such applications. decision stems from Oct 0 ruling by New Jersey Supreme Court holding that same-sex couples are entitled to same legal rights and protections as heterosexual couples**

Figure 5: Sample summaries for an NYT article. Summaries by our models with the graph encoder are more informative than the variant without it.

ing a future direction of building our model on top of a large-pretrained encoder-decoder model.

For **unfaithful errors**, we report the percentage of errors calculated by majority voting (i.e., more than one annotator vote as incorrect). First, we find that our ASGARD-SEG model has a comparable error pattern as human summaries. Specifically, for out-of-context and deletion or substitution errors, our graph-enhanced model produces significantly fewer mistakes in these categories, compared to the model without graph information. This implies that knowledge graph-enhanced models can improve summary faithfulness.

Interestingly, human-written summaries are also discerned to contain a nontrivial amount of hallucination errors. After inspection, we find that human tends to leverage world knowledge to include content that is not covered by the articles. For instance, for an article discussing events in "Boston", the human writer may describe them as happening in "Massachusetts" in the summary.

7.3 Analyzing Automatic Metrics and Summary Errors

We further plot the distributions of automatic evaluation scores regarding the three types of unfaithful errors based on majority voting in Fig. 6. First, summaries with out-of-context and deletion or substitution errors receive lower cloze and ROUGE scores overall.

Nevertheless, with regard to hallucination errors, we do not see such pattern; there is even a slightly reversed relation with both cloze scores and ROUGE scores, wherein summaries with more hallucination errors tend to score higher. This echos our previous observation that human summaries can be hallucinatory too, where world knowledge is used for writing the summaries.[2]

Furthermore, we find a weak correlation between the three variants of ROUGE scores and three types of errors, e.g., the minimum and the maximum values of Pearson's r are -0.19 and 0.14. This suggests that new metrics should be designed to better gauge summary quality. We plan to study this direction in future work.

[2]During human evaluation, we do not ask human judges to distinguish the source of hallucination errors, i.e. from world knowledge or out of fabrication, since this requires significant domain knowledge.

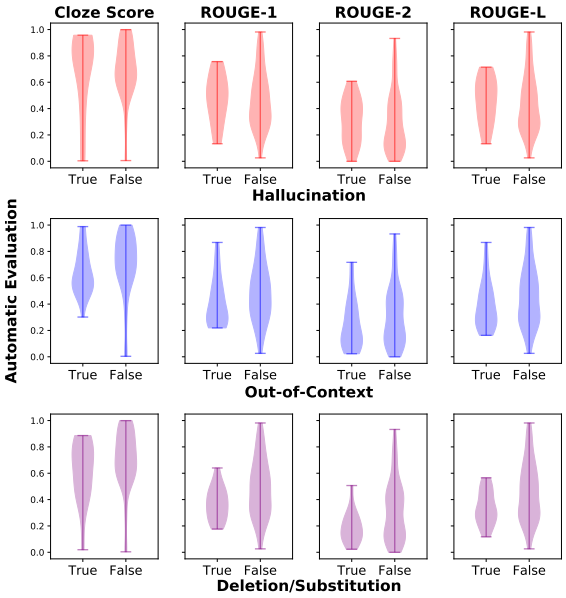


Figure 6: Distribution of automatic summarization metrics with three types of unfaithful errors. "True" indicates summaries **with** such type of error.

8 Conclusion

We presented a novel knowledge graph-augmented abstractive summarization framework, along with a novel multiple choice cloze reward for reinforcement learning. Our models capture both local characteristics and global interactions of entities from the input, thus generating summaries of higher quality. In tandem with the graph representation, our cloze reward further improves summary content. Human evaluation further confirms that our graph-augmented models trained with the cloze reward produce more informative summaries and significantly reduces unfaithful errors.

Acknowledgements

This research is supported in part by National Science Foundation through Grant IIS-1813341, and by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via contract # FA8650-17-C-9116. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein. We thank the anonymous reviewers for their suggestions.

References

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. 2015. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 344–354, Beijing, China. Association for Computational Linguistics.

Kristjan Arumae and Fei Liu. 2019. Guiding extractive summarization with question-answering rewards. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2566–2577, Minneapolis, Minnesota. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Regina Barzilay and Lillian Lee. 2004. Catching the drift: Probabilistic content models, with applications to generation and summarization. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 113–120.

Daniel Beck, Gholamreza Haffari, and Trevor Cohn. 2018. Graph-to-sequence learning using gated graph neural networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 273–283, Melbourne, Australia. Association for Computational Linguistics.

Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. 2018. Faithful to the original: Fact aware neural abstractive summarization. In *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI)*.

Asli Celikyilmaz, Antoine Bosselut, Xiaodong He, and Yejin Choi. 2018. Deep communicating agents for abstractive summarization. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1662–1675, New Orleans, Louisiana. Association for Computational Linguistics.

Yen-Chun Chen and Mohit Bansal. 2018. Fast abstractive summarization with reinforce-selected sentence rewriting. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 675–686.

Marco Damonte and Shay B Cohen. 2019. Structural neural encoders for amr-to-text generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3649–3658.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 13063–13075. Curran Associates, Inc.

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479.

Matan Eyal, Tal Baumel, and Michael Elhadad. 2019. Question answering as an automatic evaluation metric for news article summarization. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3938–3948, Minneapolis, Minnesota. Association for Computational Linguistics.

Angela Fan, Claire Gardent, Chloé Braud, and Antoine Bordes. 2019. Using local knowledge graph construction to scale Seq2Seq models to multi-document inputs. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4177–4187, Hong Kong, China. Association for Computational Linguistics.

Patrick Fernandes, Miltiadis Allamanis, and Marc Brockschmidt. 2019. Structured neural summarization. In *International Conference on Learning Representations*.

Sebastian Gehrmann, Yuntian Deng, and Alexander Rush. 2018. Bottom-up abstractive summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4098–4109, Brussels, Belgium. Association for Computational Linguistics.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman,

and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in neural information processing systems*, pages 1693–1701.

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

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Rik Koncel-Kedziorski, Dhanush Bekal, Yi Luan, Mirella Lapata, and Hannaneh Hajishirzi. 2019. Text Generation from Knowledge Graphs with Graph Transformers. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2284–2293, Minneapolis, Minnesota. Association for Computational Linguistics.

Wojciech Kryściński, Romain Paulus, Caiming Xiong, and Richard Socher. 2018. Improving abstraction in text summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1808–1817.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Chin-Yew Lin and Eduard Hovy. 2003. Automatic Evaluation of Summaries Using N-gram Co-occurrence Statistics. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, pages 71–78.

Peter J. Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. 2018. Generating wikipedia by summarizing long sequences. In *International Conference on Learning Representations*.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3721–3731, Hong Kong, China. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Hans Peter Luhn. 1958. The automatic creation of literature abstracts. *IBM Journal of research and development*, 2(2):159–165.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807.

Jekaterina Novikova, Ondrej Dusek, Amanda Cercas Curry, and Verena Rieser. 2017. Why we need new evaluation metrics for nlg. In *2017 Conference on Empirical Methods in Natural Language Processing*, pages 2231–2242. Association for Computational Linguistics.

Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A deep reinforced model for abstractive summarization. In *International Conference on Learning Representations*.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. 2017. Self-critical sequence training for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7008–7024.

Evan Sandhaus. 2008. The new york times annotated corpus. *Linguistic Data Consortium, Philadelphia*, 6(12):e26752.

Thomas Scialom, Sylvain Lamprier, Benjamin Piwowarski, and Jacopo Staiano. 2019. Answers unite! unsupervised metrics for reinforced summarization models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3244–3254, Hong Kong, China. Association for Computational Linguistics.

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, Vancouver, Canada. Association for Computational Linguistics.

Eva Sharma, Luyang Huang, Zhe Hu, and Lu Wang. 2019. An entity-driven framework for abstractive summarization. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3278–3289, Hong Kong, China. Association for Computational Linguistics.

Linfeng Song, Yue Zhang, Zhiguo Wang, and Daniel Gildea. 2018. A graph-to-sequence model for AMR-to-text generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1616–1626, Melbourne, Australia. Association for Computational Linguistics.

Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. 2017. Abstractive document summarization with a graph-based attentional neural model. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1171–1181, Vancouver, Canada. Association for Computational Linguistics.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. *International Conference on Learning Representations*. Accepted as poster.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

A Appendices

A.1 Experiment Details

Statistics of Knowledge Graphs. We show the statistics of knowledge graphs on two datasets in Table 4. On each dataset, we construct a large graph with abundant relations for each article. Note that on CNN/DM we have more arguments but fewer predicates in a document than those on NYT. This indicates CNN/DM has fewer coreferred entities.

Dataset	Doc	DocGraph		SegGraph		
	# word	# Arg.	# Pre.	# Arg.	# Pre.	# Para.
NYT	795.9	131.6	87.3	6.40	3.74	23.5
CNN/DM	789.9	138.1	85.2	6.30	3.57	24.2

Table 4: Statistics of NYT and CNN/DM datasets. # Arg.: number of arguments in each document or paragraph. # Pre.: number of predicates in each document or paragraph. # Para.: number of paragraphs in each document. Two datasets have comparable graph size.

Training Details. We utilize Adam (Kingma and Ba, 2015) with a gradient clipping of 2.0 and a batch size of 32 for all models. During ML training, a learning rate of 0.001 is used; during RL stage, it is reduced to 0.0001 (Paulus et al., 2018).

We use the base version of BERT model (Devlin et al., 2019) to select candidate answers and we fine-tune the base version of RoBERTa model (Liu et al., 2019) to build our QA model. We take pretrained models from Wolf et al. (2019).

A.2 Human Evaluation Guideline

In our human evaluation, each human annotator is presented with 100 news articles. The annotators are asked to evaluate four summaries (in random order) for each article on two aspects (informativeness and fluency) on a scale of 1 to 5 (1 being very poor and 5 being very good). Furthermore, for unfaithfulness, we define three types of unfaithful errors and ask annotators to label whether summaries contain any type of error. Instructions in Table 5 are given to human judges.

Here are descriptions of the aspects:

- **Informativeness**: Whether the summary provides enough and necessary content coverage from the input article.
- **Fluency**: Whether the summary is free of obvious grammatically incorrect sentences (e.g., fragments, missing components) that make the text difficult to read.
- **Faithfulness**: Whether the summary accords with the facts expressed in the source.

	Article: With a Little Extra Cash.
	What to do with a bonus? The right thing, of course, is to pay off debts or save it for a time when there are not any bonuses. But in Albany, any financial windfall invites hordes of legislators hungrily seeking ways to spend it. This has already started to happen, with lawmakers eyeballing a projected budgetary surplus of just under $1 billion – not all that grand when you consider that the total state budget is in the neighborhood of $120 billion, but a healthy number nonetheless. But one essential part of the equation is different this year: a new governor guarding the state finances. Nobody knows quite yet how Gov. Eliot Spitzer will manage a Legislature that wants to add a lot of its favorite things to his budget before they return it for his approval. One suggestion: Mr. Spitzer should keep his fist as tightly closed as possible, especially on his new school aid formula and his Medicaid adjustments. (....)
	Informativeness:
1	Not relevant to the article
	e.g., *"editorial on gov eliot spitzer 's plan to spend it . of new governor guarding state finances . and to spitzer should keep his fist as tightly closed as possible , especially on new school aid formula and his medicaid adjustments ."*
3	Relevant, but misses the main point of the article
	e.g., *"editorial on new gov eliot spitzer 's new governor guarding state finances . says spitzer should keep his new school aid formula and his medicaid adjustments"*
5	Successfully captures the main point of the article
	e.g., *"Editorial says New York Gov Eliot Spitzer , faced with projected $ 0 billion budget surplus , should be tight-fisted and cautious about overspending"*
	Fluency:
1	Summary is full of garbage fragments and is hard to understand
	e.g., *"of new governor guarding state finances . and to spitzer should keep his fist as tightly closed as possible , to"*
2	Summary contains fragments, missing components but has some fluent segments
	e.g., *"editorial on gov eliot spitzer 's plan to spend it . of new governor guarding state finances . and to spitzer should keep his fist as tightly closed as possible , especially on new school aid formula and his medicaid adjustments."*
3	Summary contains some grammar errors but is in general fluent
	e.g., *"editorial on any financial windfall invites hordes of legislators hungrily seeking ways to spend it . how gov eliot spitzer will manage legislature that wants to add lot of its favorite to his budget before they return it for his approval ."*
4	Summary has relatively minor grammatical errors
	e.g., *"article on in any financial windfall invites hordes of legislators hungrily seeking ways to spend it"*
5	Fluent summary
	e.g., *"editorial says new new jersey gov eliot spitzer guarding state finances . says spitzer should keep his new school aid formula and his medicaid adjustments"*
	Faithfulness:
	We define three types of unfaithful errors. Each type is labeled as "0" or "1" independently. "0" means summary does not make this type of error and "1" suggests this type of error occurs. Three types of errors are :
i	**Hallucination error**: Fabricated content that does not occur in the original article
	e.g., *"correction of dec 0 about new york column on state budget"*
ii	**Out-of-Context error**: Fact occurs in the article, but fails without correct context
	e.g., *"Editorial says one essential part of the equation is different this year: a new governor guarding the tate finances."*
iii	**Deletion or Substitution error**: Summary contains incorrectly edited, missing elements; or summary incorrectly concatenates elements from different sentences.
	e.g., *"editorial says new new jersey gov eliot spitzer guarding state finances, keeping his new school aid formula adjustments."*

Table 5: Sample summaries with explanations on human evaluation aspect scales, and the definition of three types of unfaithful errors.