FACTGRAPH: Evaluating Factuality in Summarization with Semantic Graph Representations

Leonardo F. R. Ribeiro[†*], **Mengwen Liu**[‡], **Iryna Gurevych**[†], **Markus Dreyer**[‡], **Mohit Bansal**[‡,§]

[†]UKP Lab, Technical University of Darmstadt
[‡]Amazon Alexa AI
[§]UNC Chapel Hill

ribeiro@aiphes.tu-darmstadt.de, {mengwliu, mddreyer, mobansal}@amazon.com
gurevych@ukp.informatik.tu-darmstadt.de, mbansal@cs.unc.edu

Abstract

Despite recent improvements in abstractive summarization, most current approaches generate summaries that are not *factually consistent* with the source document, severely restricting their trust and usage in real-world applications. Recent works have shown promising improvements in factuality error identification using text or dependency arc entailments; however, they do not consider the entire semantic graph simultaneously. To this end, we propose FACTGRAPH, a method that decomposes the document and the summary into structured *meaning representations* (MR), which are more suitable for factuality evaluation. MRs describe core semantic concepts and their relations, aggregating the main content in both document and summary in a canonical form, and reducing data sparsity. FACTGRAPH encodes such graphs using a graph encoder augmented with structure-aware adapters to capture interactions among the concepts based on the graph connectivity, along with text representations using an adapter-based text encoder. Experiments on different benchmarks for evaluating factuality show that FACTGRAPH outperforms previous approaches by up to 15%. Furthermore, FACTGRAPH improves performance on identifying content verifiability errors and better captures subsentence-level factual inconsistencies.[1]

1 Introduction

Recent summarization approaches based on pre-trained language models (LM) have established a new level of performance (Zhang et al., 2020; Lewis et al., 2020), generating summaries that are grammatically fluent and capable of combining salient parts of the source document. However, current models suffer from a severe limitation, generating summaries that are *not factually consistent*, that is, the content of the summary does not meet the facts of the source document, an issue also known as *hallucination*. Previous studies (Cao et al., 2018; Falke et al., 2019; Maynez et al., 2020; Dreyer et al., 2021) report rates of hallucinations in generated summaries ranging from 30% to over 70%. In the face of such a challenge, recent works employ promising ideas such as question answering (QA) (Durmus et al., 2020; Nan et al., 2021) and weakly supervised approaches (Kryscinski et al., 2020) to assess factuality. Another line of work explores dependency arc entailment to improve the localization of subsentence-level errors within generated summaries (Goyal and Durrett, 2020).

However, these methods have a reduced correlation with human judgments and may not capture well semantic errors (Pagnoni et al., 2021). One reason for the poor performance is the lack of good quality factuality training data. Second, it is challenging to properly encode core semantic content from the document and summary (Lee et al., 2021)

(a) […] **Mr Mueller was given the role of special counsel by the justice department to lead its investigation into alleged Russian interference in last year's US election** […] The NYT has reported that **Mr Trump has considered firing Mr Mueller** […].

(b) US President Donald Trump has said he will fire special counsel Robert Mueller, who is investigating alleged Russian interference in the US election.

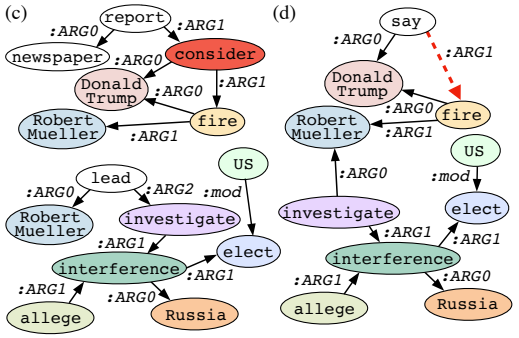

Figure 1: Example of (a) a document, (b) a summary, and (c) the corresponding document and (d) summary graph-based meaning representations. The summary graph does not contain the "consider" node, indicating a factual error (red dashed edge).

[*] Work done as an intern at Amazon Alexa AI.

[1] Our code will be publicly available at https://github.com/amazon-research/fact-graph

and reason over salient pieces of information in order to assess the summary factuality. Third, previous work (DAE, Goyal and Durrett, 2021) treats semantic relations as isolated units, not simultaneously considering the entire semantic structure of *both* document and summary texts.

To mitigate the above issues, we explore *meaning representations* (MR) as a form of content representation for factuality evaluation. We present FACTGRAPH, a novel graph-enhanced approach that incorporates core information from the document and the summary into the factuality model using graph-based MRs, which are more suitable for factuality evaluation: As shown in Figure 1, graph-based MRs capture semantic relations between entities, abstracting away from syntactic structure and producing a canonical representation of meaning.

Different from previous methods (Kryscinski et al., 2020; Goyal and Durrett, 2021), FACTGRAPH is a dual approach which encodes both text and graph modalities, better integrating linguistic knowledge and structured semantic knowledge. As shown in Figure 2, it is composed of parameter-efficient text and graph encoders which share the same pretrained model and differ by their adapter weights (Houlsby et al., 2019). The texts from the document and summary are encoded using the adapter-based text encoder whereas the entire semantic structures that represent document and summary facts are used as input to the graph encoder augmented structure-aware adapters (Ribeiro et al., 2021b). The representations of the two modalities thus are combined to generate the factuality score.

In particular, we explore *Abstract Meaning Representation* (AMR) (Banarescu et al., 2013), a semantic formalism that has received much research interest (Song et al., 2018; Guo et al., 2019; Ribeiro et al., 2019, 2021a; Opitz et al., 2020, 2021; Fu et al., 2021) and has been shown to benefit downstream tasks such as spoken language understanding (Damonte et al., 2019), machine translation (Song et al., 2019), commonsense reasoning (Lim et al., 2020), and question answering (Kapanipathi et al., 2021; Bornea et al., 2021).

Intuitively, AMR provides important benefits: First, it encodes core concepts as it strives for a more logical and less syntactic representation, which has been shown to benefit text summarization (Hardy and Vlachos, 2018; Dohare et al., 2018; Lee et al., 2021). Furthermore, AMR captures semantics at a high level of abstraction explicitly modeling relations in the text and reducing the negative influence of diverse text surface variances with the same meaning. Lastly, recent studies (Dreyer et al., 2021; Ladhak et al., 2021) demonstrate that there is a trade-off between factuality and abstractiveness. Structured semantic representations are potentially beneficial for reducing data sparsity and localizing generation errors in abstractive scenarios. Figure 1 shows examples of (c) document and (d) summary AMRs, where the summary AMR is missing a crucial modifying node present in the document AMR, which indicates a factual error in the summary.

We consolidate a factuality dataset with human annotations derived from previous works (Wang et al., 2020; Kryscinski et al., 2020; Maynez et al., 2020; Pagnoni et al., 2021). This dataset is constructed from the widely-used CNN/DM (Hermann et al., 2015) and XSum (Nallapati et al., 2016) benchmarks. Extensive experimental results demonstrate that FACTGRAPH achieves substantial improvements over previous approaches, improving factuality performance by up to 15% and correlation with human judgments by up to 10%, capturing more content verifiability errors and better classifying factuality in semantic relations.

2 Related Work

Evaluating Factuality. Recently, there has been a surge of new methods for factuality evaluation in text generation, especially for summarization. Falke et al. (2019) propose to rerank summary hypotheses generated via beam search based on entailment scores to the source document. Kryscinski et al. (2020) introduce FACTCC, a model-based approach trained on artificially generated data, to measure if the summary can be entailed by the source document in order to assess the summary factuality. QA-based methods (Wang et al., 2020; Durmus et al., 2020; Honovich et al., 2021; Nan et al., 2021) generate questions from the document and summary, and compare the corresponding answers in order to assess factuality. Xie et al. (2021) formulate causal relationships among the document, summary, and language prior to evaluate the factuality via counterfactual estimation.

Categorizing Factual Errors. A thread of analysis work has focused on identifying different categories of factual errors in summarization. Maynez et al. (2020) show that semantic inference-based automatic measures are better representations of sum-

marization quality, whereas Pagnoni et al. (2021) propose a linguistically grounded typology of factual errors and develop a fine-grained benchmark for factuality evaluation, moving to a fine-grained measure, instead of using a binary evaluation. Fabbri et al. (2021) introduce different resources for summarization evaluation which include a toolkit for evaluating summarization models.

Factuality versus Abstractiveness. Recent works (Dreyer et al., 2021; Ladhak et al., 2021) investigate the trade-off between factuality and abstractiveness of summaries and observe that factuality tends to drop with increased abstractiveness. Semantic graphs are uniquely suitable to detect factual errors in abstractive summaries as they abstract away from the lexical surface forms of documents and summaries, enabling direct comparisons of the underlying semantic concepts and relations of a document-summary pair.

Graph-based Representations for Summarization. A growing body of work focuses on using graph-based representations for improving summarization. Whereas different approaches encode graphs into neural models for multi-document summarization (Fan et al., 2019; Li et al., 2020; Pasunuru et al., 2021; Wu et al., 2021; Chen et al., 2021), AMR structures have been shown to benefit both document representation and summary generation (Liu et al., 2015; Liao et al., 2018; Hardy and Vlachos, 2018; Dohare et al., 2018) and have the potential of improving controllability in summarization (Lee et al., 2021). The above works are related to FACTGRAPH as they use semantic graphs for content representation, but also different because they utilize graphs for the downstream summarization task, whereas FACTGRAPH employ them for factuality evaluation.

Semantic Representations for Factuality Evaluation. More closely to our work, Goodrich et al. (2019) extract tuples from the document and summary and measure the factual consistency by overlapping metrics. Recently, dependency arc entailment (DAE, Goyal and Durrett, 2020) is used to measure subsentence-level factuality by classifying pairs of words defined by dependency arcs which often describe semantic relations. However, FACTGRAPH is considerably different from those approaches, since it explicitly encodes the entire graph semantic structure into the model. Moreover, while DAE considers semantic edge relations of the summary only, FACTGRAPH encodes the semantic structures of both the input document and summary leading to better factuality performance at both sentence and subsentence levels.

3 FACTGRAPH Model

We introduce FACTGRAPH, a method that employs semantic graph representations for factuality evaluation in text summarization, describing its intuition (§3.3) and defining it formally (§3.4).

3.1 Problem Statement

Given a source document D and a *sentence-level* summary S, we aim to check whether S is *factual* with respect to D. For each sentence $d \in D$ we extract a semantic graph $\mathcal{G}_d$. Similarly, for the summary sentence S we extract its semantic graph $\mathcal{G}_s$. We use texts and graphs from both document and summary for factuality evaluation. Sentence-level summary predictions can be aggregated to generate a factuality score for a multi-sentence summary.

3.2 Extracting Semantic Graphs

We select AMR as our MR, but FACTGRAPH can be used with other graph-based semantic representations, such as OpenIE (Banko et al., 2007). AMR is a linguistically-grounded semantic formalism that represents the meaning of a sentence as a rooted graph, where nodes are *concepts* and edges are *semantic relations*. AMR abstracts away from surface text, aiming to produce a more language-neutral representation of meaning. We use a state-of-the-art AMR parser (Bevilacqua et al., 2021) to extract an AMR graph $\mathcal{G}_a = (\mathcal{V}_a, \mathcal{E}_a, \mathcal{R}_a)$ with a node set $\mathcal{V}_a$ and labeled edges $(u, r, v) \in \mathcal{E}_a$, where $u, v \in \mathcal{V}_a$ and $r \in \mathcal{R}_a$ is a relation type. Each $\mathcal{G}_a$ aims to explicitly represent the core concepts in each sentence. Figure 1 shows an example of a (b) sentence and its (d) corresponding AMR graph.

Graph Representation. We convert each $\mathcal{G}_a$ into a bipartite graph $\mathcal{G}_b = (\mathcal{V}_b, \mathcal{E}_b)$, replacing each labeled edge $(u, r, v) \in \mathcal{E}_a$ with two unlabeled edges $\{(u, r), (r, v)\} \in \mathcal{E}_b$. Similar to Beck et al. (2018), this procedure transforms the graph into its unlabeled version. Pretrained models typically use a vocabulary with subword tokens, which makes it complicated to properly represent a graph using subword tokens as nodes. Inspired by Ribeiro et al. (2020, 2021b), we transform each $\mathcal{G}_b$ into a new token graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where each token of a node $v_b \in \mathcal{V}_b$ becomes a node $v \in \mathcal{V}$. We convert

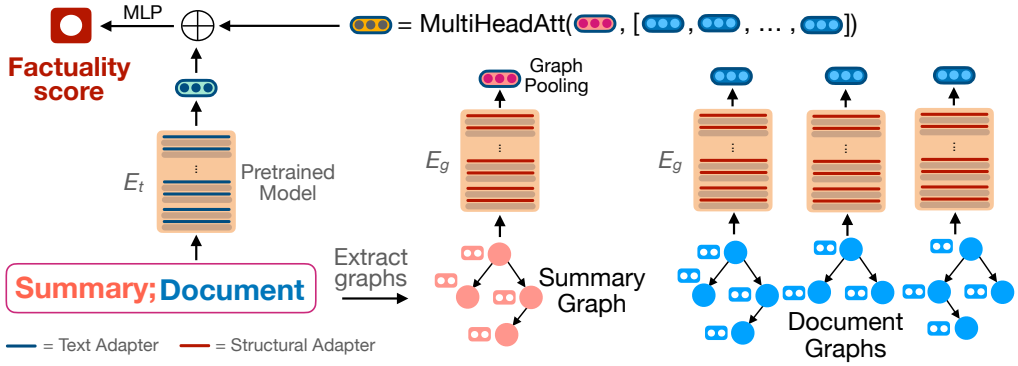

Figure 2: Overview of FACTGRAPH. A sentence-level summary and document graphs are encoded by the graph encoder with structure-aware adapters. Text and graph encoders use the same pretrained model and only the adapters parameters are trained.

each edge $(u_b, v_b) \in \mathcal{E}_b$ into a set of edges and connect every token of u_b to every token of v_b.

3.3 Intuition of Semantic Representation

In order to represent facts to better assess the summary factuality, we draw inspiration from traditional approaches to summarization that condense the source document to a set of "semantic units" (Liu et al., 2015; Liao et al., 2018). Intuitively, the semantic graphs from the source document represent the core factual information, explicitly modeling relations in the text, whereas the semantic summary graph captures the essential content information in a summary (Lee et al., 2021). The document graphs can be compared with the summary graph, measuring the degree of semantic overlap to assess factuality (Cai and Knight, 2013).

Recently, sets of fact triples from summaries were used to estimate factual accuracy (Goodrich et al., 2019). That approach is related to FACTGRAPH as it uses graph-based MRs, but also different because it compares the reference and the generated summary, whereas we compare the generated summary with the input document. Moreover, differently from Goodrich et al. (2019), FACTGRAPH explicitly encodes the semantic structures using a graph encoder and employs AMR as a semantic representation. Finally, in contrast to DAE (Goyal and Durrett, 2021), which focuses only on extracting summary graph representations, FACTGRAPH uses semantic graphs for both document and summary.

3.4 Model

Figure 2 illustrates FACTGRAPH, which is composed of text and graph encoders. The text encoder, denoted by E_t, uses a pretrained encoder E, augmented with adapter modules which receives the summary S and document D and outputs a contextual text representation. Conversely, the graph encoder, denoted by E_g, uses the same E, but is augmented with structure-aware adapters. E_g receives the summary and multiple document semantic graphs corresponding to its sentences, and outputs graph-aware contextual representations that are used to generate the final graph representation. During training, only adapter weights are trained, whereas the weights from E are kept frozen. Finally, both graph and text representations are concatenated and fed to a final classifier, which predicts whether the summary is factual or not.

Text Encoder. We employ an adapter module before and after the feed-forward sub-layer of each layer of the encoder. We modify the adapter architecture from Houlsby et al. (2019). We compute the adapter representation for each token i at each layer l, given the token representation $\boldsymbol{h}_i^l$, as follows:

$$\hat{\boldsymbol{z}}_i^l = \boldsymbol{W}_o^l(\sigma(\boldsymbol{W}_p^l \,\mathsf{LN}(\boldsymbol{h}_i^l))) + \boldsymbol{h}_i^l\,, \qquad (1)$$

where σ is the activation function and $\mathsf{LN}(\cdot)$ denotes layer normalization. $\boldsymbol{W}_o^l \in \mathbb{R}^{d\times m}$ and $\boldsymbol{W}_p^l \in \mathbb{R}^{m\times d}$ are adapter parameters. The representation of the [CLS] token is used as the final textual representation, denoted by $\boldsymbol{t}$.

Graph Encoder. In order to re-purpose the pretrained encoder to structured inputs, we employ a structural adapter (Ribeiro et al., 2021b). In particular, for each node $v \in \mathcal{V}$, given the hidden representation $\boldsymbol{h}_v^l$, the encoder layer l computes:

$$\boldsymbol{g}_v^l = \mathsf{GraphConv}_l(\mathsf{LN}(\boldsymbol{h}_v^l), \{\mathsf{LN}(\boldsymbol{h}_u^l) : u \in \mathcal{N}(v)\})$$
$$\boldsymbol{z}_v^l = \boldsymbol{W}_e^l \sigma(\boldsymbol{g}_v^l) + \boldsymbol{h}_v^l\,, \qquad (2)$$

where $\mathcal{N}(v)$ is the neighborhood of the node v in $\mathcal{G}$ and $\mathbf{W}_e^l \in \mathbb{R}^{d \times m}$ is a adapter parameter. $\mathsf{GraphConv}_l(\cdot)$ is the graph convolution that computes the representation of v based on its *neighbors* in the graph. We employ a Relational Graph Convolutional Network (Schlichtkrull et al., 2018) as graph convolution, which considers differences in the incoming and outgoing relations. Since AMRs are directed graphs, encoding edge directions is beneficial for downstream performance (Ribeiro et al., 2019). The structural adapter is placed before, whereas the normal adapter is kept after the feed-forward sub-layer of each encoder layer.

We calculate the final representation of each graph from the pooling denoted as $\boldsymbol{z}^{\mathrm{G}} = \{\mathbf{z}_v^{(L)} \mid v \in \mathcal{V}\}$, where $\mathbf{z}_v^{(L)}$ is the final representation of v. Thus, we use a multi-head self-attention (Vaswani et al., 2017) layer to estimate to what extent each sentence graph contributes to the document semantic representation based on the summary graph. This mechanism allows encoding a global document representation based on the summary graph. In particular, each attention head computes:

$$\begin{aligned} \alpha_i &= \mathtt{Attn}(\boldsymbol{z}_s^{\mathrm{G}}, \boldsymbol{z}_i^{\mathrm{G}}), \\ \boldsymbol{g} &= \sum\nolimits_{i=1}^{k} \alpha_i \, \boldsymbol{W}_r \, \boldsymbol{z}_i^{\mathrm{G}}, \end{aligned} \tag{3}$$

where $\boldsymbol{z}_s^{\mathrm{G}}$ is the final representation of $\mathcal{G}_s$, k is the number of considered sentence graphs from the input document and $\mathbf{W}_r \in \mathbb{R}^{d \times d}$ is a parameter.

The final representation is derived from the text and graph representations, $\boldsymbol{q} = [\boldsymbol{t}; \boldsymbol{g}]$, and fed into a classification layer that outputs a probability distribution over the labels y = {Factual, Non-Factual}.

3.5 Edge-level Factuality Model

Inspired by Goyal and Durrett (2021), we evaluate the factuality at the edge level. In this setup, we use the same text and graph encoders; however, we encode the semantic graphs differently. In particular, we concatenate $\mathcal{G}_s$ with each $\mathcal{G}_d \in D$ and feed the concatenation to the graph encoder. The representation of a node $v \in \mathcal{G}_s$ is calculated as:

$$\begin{aligned} \boldsymbol{r}_{t_v} &= \sum\nolimits_{w \in A(v)} E_t(D; S)_w \\ \boldsymbol{r}_{g_v} &= \sum\nolimits_{d=1}^{k} E_g(\mathcal{G}_s; \mathcal{G}_d)_v \\ \boldsymbol{r}_v &= [\boldsymbol{r}_{t_v}; \boldsymbol{r}_{g_v}] \end{aligned} \tag{4}$$

where $A(v)$ is the set of all summary words aligned with v, and $\boldsymbol{r}_{t_v}$ and $\boldsymbol{r}_{g_v}$ are the word and node representations, respectively. Edge representations are derived for each AMR edge $(u, v) \in \mathcal{E} : \boldsymbol{r}_e = [\boldsymbol{r}_u; \boldsymbol{r}_v]$. The edge representation $\boldsymbol{r}_e$ is fed into a classification layer that outputs a probability distribution over the output labels (y_e = {Factual, Non-Factual}). We assign the label *non-factual* for an edge in $\mathcal{G}_s$ if one of the nodes in this edge is aligned with a word that belongs to a span annotated as *non-factual*. Otherwise, the edge is assigned the label *factual*. We call this variant FACTGRAPH-E.

Source	# datapoints	Domain
Wang et al. (2020)	953	CNN/DM, XSum
Kryscinski et al. (2020)	1,434	CNN/DM
Maynez et al. (2020)	2,500	XSum
Pagnoni et al. (2021)	4,942	CNN/DM, XSum
Total	9,829	CNN/DM, XSum

Table 1: Consolidated Human annotations.

4 Experimental Setup

4.1 Data

One of the main challenges in developing models for factuality evaluation is the lack of training data. Existing synthetic data generation approaches are not well-suited to factuality evaluation of current summarization models and human-annotated data can improve factuality models (Goyal and Durrett, 2021). In order to have a more effective training signal, we gather human annotations from different sources and consolidate a factuality dataset that can be used to train FACTGRAPH and other models.

The source collections of the dataset are presented in Table 1. The dataset covers two parts, namely CNN/DM (Hermann et al., 2015) and XSum (Nallapati et al., 2016). CNN/DM contains news articles from two providers, CNN and DailyMail; while XSum contains BBC articles. CNN/DM has considerably lower levels of abstraction, and the summary exhibits high overlap with the article; a typical CNN/DM summary consists of several bullet points. In XSum, the first sentence is removed from an article and used as a summary, making it highly abstractive. After we remove duplicated annotations, the total number of datapoints is 9,567, which we divide into train (8,667), dev (300) and test (600) sets. We call this dataset FACTCOLLECT.

4.2 Method Details

Selecting the Document Semantic Graphs. We limit the number of considered document graphs due to efficiency reasons. In particular, we com-

Model	All data		CNN/DM		XSum	
	BACC	F1	BACC	F1	BACC	F1
QAGS (Wang et al., 2020)	79.8	79.7	64.2	76.2	59.3	85.2
QUALS (Nan et al., 2021)	78.3	78.5	60.8	76.2	57.5	82.2
FACTCC (Kryscinski et al., 2020)	76.0	76.3	69.0	77.8	55.9	73.9
FACTCC+	83.9 (0.4)	84.2 (0.4)	68.0 (1.0)	83.7 (0.5)	58.3 (2.2)	84.9 (1.0)
FACTGRAPH	86.3 (1.3)	86.7 (1.1)	73.0 (2.3)	86.8 (0.8)	68.6 (2.3)	86.6 (2.0)
FACTGRAPH (pretrained structural adapters)	86.4 (0.6)	86.8 (0.5)	74.1 (1.0)	87.4 (0.3)	**70.4** (1.9)	85.9 (1.4)
FACTGRAPH (pretrained structural and text adapters)	**87.6** (0.7)	**87.8** (0.7)	**76.0** (2.8)	**87.5** (0.4)	69.9 (2.3)	**88.4** (1.2)

Table 2: BACC and F1 scores for factuality models in the test set of FACTCOLLECT. Mean (±s.d.) over 5 seeds.

pute the pairwise cosine similarity between the embeddings of each sentence $d \in D$ and the summary sentence S, generated by Sentence Transformers (Reimers and Gurevych, 2019). We thus select k sentences from the source document with the highest scores to be used to generate the document semantic graphs.

The model weights are initialized with ELECTRA (electra-base discriminator, 110M parameters, Clark et al., 2020), the structural adapters are pretrained using the release 3.0 of the AMR corpus containing 55,635 gold annotated AMR graphs, and the text adapters are pretrained using synthetic generated data. The adapters' hidden dimension is 32, which corresponds to about 1.4% of the parameters of the original ELECTRA encoders. The number of considered document graphs (k) is 5.[2] We report the test results when the balanced accuracy (BACC) on dev set is optimal. Following previous work (Kryscinski et al., 2020; Goyal and Durrett, 2021), we evaluate our models using BACC and Micro F1 scores.

5 Results and Analysis

We compare FACTGRAPH with different methods for factuality evaluation: two QA-based methods, namely QAGS (Wang et al., 2020) and QUALS (Nan et al., 2021), and FACTCC (Kryscinski et al., 2020). We fine-tune FACTCC using the training set, that is, it is trained on both synthetic data and FACTCOLLECT. We call this approach FACTCC+.

Table 2 presents the results. QA-based approaches perform comparatively worse than FACTCC on CNN/DM, while QAGS has a general better performance than QUALS. FACTCC has a strong performance on CNN/DM, as it was trained on synthetic data derived from this dataset. However, the FACTCC's performance does not transfer to XSum. FACTCC+ has a large increase in performance, especially on XSum, demonstrating the importance of human-annotated data for training improved factuality models.

FACTGRAPH outperforms FACTCC+ by 2.4 BACC points in both subsets and by 10.3 BACC in XSum, even though FACTCC+ was pretrained on millions of synthetic examples. This indicates that considering semantic representations is beneficial for factuality evaluation and FACTGRAPH can be trained on a small number of annotated examples. Pretraining structural adapters improves the performance on CNN/DM and XSum. Finally, FACTGRAPH's performance further improves when both structural and text adapters are pretrained, improving over FACTCC+ by 3.7 BACC points.[3]

5.1 Correlation with Human Judgments

We also evaluate the model performance using correlations with human judgments of factuality (Pagnoni et al., 2021). In this experiment, FACTCC+ and FACTGRAPH are trained with the FACTCOLLECT data without the Pagnoni et al. (2021)'s subset, which is used as dev and test sets, according to its split. For both models, following Pagnoni et al. (2021), we obtain a binary factuality label for each sentence and take the average of these labels as the final summary score. We use the official script to calculate the correlations.[4]

AMR and Factuality. We investigate whether SMATCH (Cai and Knight, 2013), a metric that measures the degree of overlap between two AMRs, correlates with factuality judgments. We calculate the SMATCH score between all the summary sentence graphs and k document sentence graphs, with $k \in \{1, 3, 5\}$. We obtain one score per summary sentence by maxing over its scores with the sentence graphs, then averaging over the summary sentence scores to obtain the summary-level score. We also calculate the SMATCH between the generated summary and the reference summary graphs.

[2] Hyperparameter details and pretraining procedures are described in Appendix A.

[3] FACTGRAPH is significantly better than FACTCC+ with $p<0.05$ on both BACC and F1 scores.

[4] https://github.com/artidoro/frank

	All data				CNN/DM				XSum			
Pearson, Spearman	ρ	p-val	r	p-val	ρ	p-val	r	p-val	ρ	p-val	r	p-val
BLEU	.10	.00	.05	.02	.06	.06	.07	.02	.16	.00	.15	.00
METEOR	.13	.00	.10	.00	.11	.00	.11	.00	.16	.00	.08	.01
ROUGE-L	.13	.00	.09	.00	.09	.00	.10	.00	.17	.00	.09	.01
BERTSCORE	.16	.00	.11	.00	.13	.00	.12	.00	.19	.00	.10	.00
SMATCH-AMR$_1$	.07	.00	-.01	.62	.07	.02	.03	.26	.09	.01	.07	.05
SMATCH-AMR$_3$	.11	.00	.10	.00	.15	.00	.14	.00	.06	.10	.04	.21
SMATCH-AMR$_5$	.13	.00	.13	.00	.17	.00	.16	.00	.05	.17	.04	.28
SMATCH-AMR$_{ref}$	.08	.00	.03	.20	.05	.12	.03	.35	.13	.00	.08	.02
QAGS	.22	.00	.23	.00	.34	.00	.27	.00	.07	.05	.06	.09
QUALS	.22	.00	.19	.00	.31	.00	.27	.00	.14	.00	.07	.03
DAE	.17	.00	.20	.00	.27	.00	.22	.00	.03	.38	.33	.00
FACTCC	.20	.00	.29	.00	.36	.00	.30	.00	.06	.07	.19	.00
FACTCC+	.32	.00	.38	.00	.40	.00	.28	.00	.24	.00	.16	.00
FACTGRAPH	**.35**	.00	**.42**	.00	**.45**	.00	**.34**	.00	**.30**	.00	**.49**	.00

Table 3: Partial Pearson and Spearman correlation coefficients and p-values between human judgements and methods scores for the test split of Pagnoni et al. (2021).

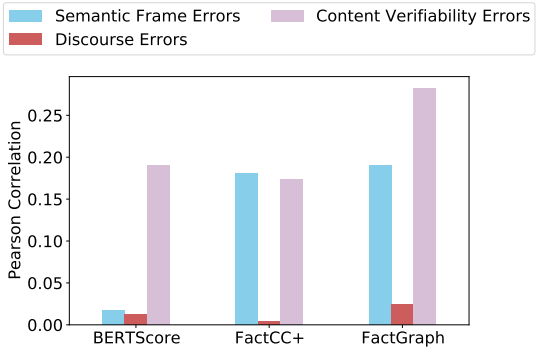

Figure 3: Variation in partial Pearson correlation when omitting error types. Higher variation indicates greater influence of an error type in the overall correlation.

As shown in Table 3, SMATCH approaches have a small but consistent correlation, slightly improving over n-gram based metrics (e.g., METEOR and ROUGE-L) in CNN/DM, suggesting that AMR, which has a higher level of abstraction than plain text, may be a semantic representation alternative to content verification.

QA-based approaches have higher correlation on the CNN/DM dataset than XSum where their correlation is relatively reduced, and DAE shows higher Spearman correlation than FACTCC on XSum. FACTCC+ and FACTGRAPH, which are trained on data from FACTCOLLECT, have a overall higher performance than models trained on synthetic data, such as FACTCC, again demonstrating the importance of the human-annotation signal when training factuality evaluation approaches. Finally, FACTGRAPH has the highest correlations in both datasets, with a large improvement in XSum, suggesting that representing facts as semantic graphs is effective for more abstractive summaries.

Sentence-level models	**BACC**
Sent-Factuality (Goyal and Durrett, 2021)	65.6
FACTGRAPH	74.9
Edge-level models	**BACC**
DAE (Goyal and Durrett, 2021)	78.7
FACTGRAPH-E	**81.1**

Table 4: Sentence-level BACC in human-annotated XSum generated summaries (Maynez et al., 2020).

Types of Errors. Figure 3 shows the influence of the different types of factuality errors (Pagnoni et al., 2021) for each approach. *Semantic Frame Errors* are errors in a frame, core, and non-core frame elements.[5] *Discourse Errors* extend beyond a single semantic frame introducing erroneous links between discourse segments. *Content Verifiability Errors* capture cases when it is not possible to verify the summary against the source document due to the difficulty in aligning it to the source.[6] Note that whereas BERTSCORE strongly correlates with content verifiability errors as it is a token-level similarity metric, the other methods improve in *Semantic Frame Errors*. FACTGRAPH has the highest performance suggesting that graph-based MRs are able to capture different semantic errors well. In particular, FACTGRAPH improves in capturing content verifiability errors by 48.2%, suggesting that representing facts using AMR is helpful.

5.2 Edge-level Factuality Classification

We assess factuality beyond sentence-level with FACTGRAPH-E (§3.5). We train and evaluate the

[5] A semantic frame is a representation of an event, relation, or state (Baker et al., 1998).

[6] Refer to Pagnoni et al. (2021) for a detailed description of the error categories and the correlation computations.

Model	BACC	F1
Only graphs	77.7	78.0
Only text	88.4	88.6
FACTGRAPH	91.2	91.3

Table 5: Ablation study for different components of the model in the FACTCOLLECT's dev set.

# Graphs	BACC	F1
1	90.1	90.3
3	90.9	91.0
5 (**final**)	91.2	91.3
7	89.8	90.0

Table 6: Effect in the FACTCOLLECT's dev set of the number of considered AMR graphs from the document.

model against the sentence-level factuality data from Maynez et al. (2020). In this dataset, human annotations for sentence and span levels are available. We derive the edge labels required for FACTGRAPH-E training as follows: For each edge in the summary graph, if one of the nodes connected to this edge is aligned with a word that belongs to a span labeled as non-factual, the edge is annotated as non-factual.[7] Summary-level labels are obtained from edge-level predictions: if any edge in the summary graph is classified as non-factual, the summary is labeled as non-factual. We use the same splits from Goyal and Durrett (2021).[8] We compare FACTGRAPH-E with DAE and additionally with a sentence-level baseline (Goyal and Durrett, 2021) and FACTGRAPH.

Table 4 shows that the edge-level factuality classification gives better performance than sentence-level classification, and FACTGRAPH performs better in both sentence and edge classification levels. FACTGRAPH-E outperforms DAE, demonstrating that training on subsentence-level factuality annotations enables it to accurately predict edge-level factuality and output summary-level factuality.

Finally, while the semantic representations contribute to overall performance, extracting those representations adds some overhead in preprocessing time (and lightly more in inference time), as shown in Appendix B.

5.3 Model Ablations

In Table 5, we report an ablation study on the impact of distinct FACTGRAPH's components. First, note that only encoding the textual information leads to better performance than just encoding graphs. This is expected since pretrained encoders are known for good performance in NLP textual tasks due to their transfer learning capabilities and the full document text encodes more information than the selected k document graphs. Moreover, AMR representations abstract aspects such as verb tenses, making the graphs agnostic regarding more fine-grained information. However, this is compensated in FACTGRAPH, which captures coarse-grained details from the text modality. Future work can consider incorporating such information into the graph representation in order to improve the factuality assessment.

Ultimately, FACTGRAPH, which uses both document and summary graphs, gives the overall best performance, demonstrating that semantic graph representations complement the text representation and are beneficial for factuality evaluation.

Number of Document Graphs Table 6 shows the influence of the number of considered document graphs measured on FACTCOLLECT's dev set performance. Note that generally more document graphs leads to better performance with a peak in 5. This suggests that using all graph sentences from the source document is not required for better performance. Moreover, the results indicate that our strategy of selecting document graphs using the contextual representations of the document sentences which are compared to the summary performs well in practice.

We additionally present the performance of FACTGRAPH with other semantic representations in Appendix C.

5.4 Comparison to Full Fine-tuning.

FACTGRAPH only trains adapter weights that are placed into each layer of both text and graph encoders. We compare FACTGRAPH with a model with similar architecture, with both text and graph encoders, but without (structural) adapter layers. We then fine-tune all the model parameters. Table 7 shows that FACTGRAPH performs better even though it trains only 1.4% of the parameters of the fully fine-tuned model, suggesting that the structural adapters help to adapt the graph encoder to semantic graph representations.

5.5 Case Study

FACTGRAPH-E computes factuality scores for

[7] We use the JAMR aligner (Flanigan et al., 2014) to obtain node-to-word alignments.

[8] We sample 100 datapoints from the training set as dev set to execute hyperparameter search.

Article: Margaret Fleming, 36, was last seen at her home in Inverkip by her two carers at about 17:40 on Friday 28 October. She is described as about 5ft 5in tall, […]. Police had said they were trying to build a picture of Ms Fleming's life, part of which she kept "quite private". When last seen, she was wearing a green tartan fleece[…]. She also had a satchel-type handbag. A police spokesman said: "There is a specialist search team combing the area around where the missing person was last seen, this includes in the garden of her last known address." […] The detective said that Ms Fleming was a student at James Watt College in Greenock between 1996 and 1997. He said **he was keen to speak to anyone** who remembered her from then, and who might have been in touch with her over the years.

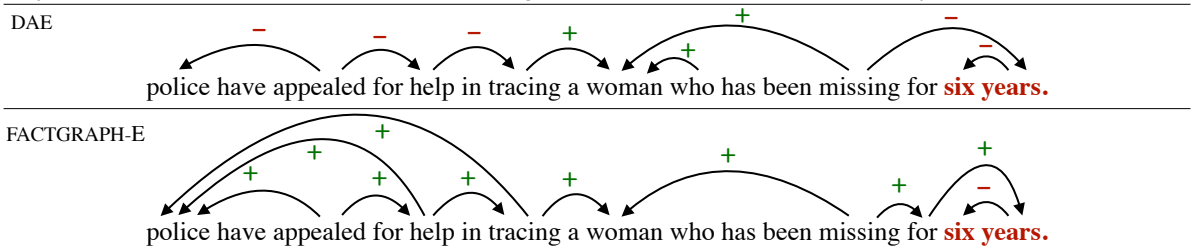

Figure 4: An example of a document, its generated summary and factuality predictions for word pairs, based on the dependency graph (DAE) versus AMR graph (FACTGRAPH-E). $+/-$ means the predicted label for that edge.

	BACC	F1	Parameters
Fully fine-tuned	90.3	90.3	100.0%
FACTGRAPH	91.2	91.3	1.4%

Table 7: Comparison between FACTGRAPH and fully fine-tuning in the dev set of FACTCOLLECT.

each edge of the AMR summary graph and those predictions are aggregated to generate a sentence-level label (§5.2). Alternatively, it is possible to identify specific inconsistencies in the generated summary based on the AMR graph structure. This factuality information at subsentence-level can provide deeper insights on the kinds of factual inconsistencies made by different summarization models (Maynez et al., 2020) and can supply text generation approaches with localized signals for training (Cao et al., 2020; Zhu et al., 2021).

Figure 4 shows a document, its generated summary, and factuality edge predictions by DAE and FACTGRAPH-E.[9] First, note that since DAE uses dependency arcs and FACTGRAPH-E is based on AMR, the sets of edges in both approaches, that is, the relations between nodes and hence words, are different. Second, both methods are able to detect the hallucination *six years*, which was never mentioned in the source document. However, DAE does not consider that *police appealed for help in tracing* is factual whereas FACTGRAPH-E captures it. This piece of information is related to a span in the document with a very different but semantically related form (highlighted in bold in Figure 4). This poses challenges to DAE, since it classifies semantic relations independently and only considers the text surface. On the other hand, FACTGRAPH-E matches the summary against the document not only at text surface level but semantic level.

[9] Appendix D presents the complete AMR and dependency summary graphs.

6 Conclusion

We presented FACTGRAPH, a graph-based approach to explicitly encode facts using meaning representations to identify factual errors in generated text. We provided an extensive evaluation of our approach and showed that it significantly improves results on different factuality benchmarks for summarization, indicating that structured semantic representations are beneficial to factuality evaluation. Future work includes (i) exploring approaches to develop document-level semantic graphs (Naseem et al., 2021), (ii) an explainable graph-based component to highlight hallucinations and (iii) to combine different meaning representations in order to capture distinct semantic aspects.

Acknowledgments

We thank the anonymous reviewers for their valuable feedback. We also thank Shiyue Zhang, Kevin Small, and Yang Liu for their feedback on this work. Leonardo F. R. Ribeiro has been supported by the German Research Foundation (DFG) as part of the Research Training Group "Adaptive Preparation of Information form Heterogeneous Sources" (AIPHES, GRK 1994/1).

Impact Statement

In this paper, we study the problem of detecting factual inconsistencies in summaries generated from

input documents. The proposed models better consider the text internal meaning structure and could benefit general generation applications by evaluating their output regarding factual consistency, which could ensure that these systems are more trustworthy. This work is built using semantic representations extracted using AMR parsers. In this way, the quality of the parser used to generate the semantic representations can significantly impact the results of our models. In our work, we mitigate this risk by employing a state-of-the-art AMR parser.

References

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 1*, ACL '98/COLING '98, page 86–90, USA. Association for Computational Linguistics.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

Michele Banko, Michael J. Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI'07, page 2670–2676, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

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.

Michele Bevilacqua, Rexhina Blloshmi, and Roberto Navigli. 2021. One spring to rule them both: Symmetric AMR semantic parsing and generation without a complex pipeline. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(14):12564–12573.

Mihaela Bornea, Ramon Fernandez Astudillo, Tahira Naseem, Nandana Mihindukulasooriya, Ibrahim Abdelaziz, Pavan Kapanipathi, Radu Florian, and Salim Roukos. 2021. Learning to transpile AMR into SPARQL.

Deng Cai and Wai Lam. 2020. AMR parsing via graph-sequence iterative inference. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1290–1301, Online. Association for Computational Linguistics.

Shu Cai and Kevin Knight. 2013. Smatch: an evaluation metric for semantic feature structures. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 748–752, Sofia, Bulgaria. Association for Computational Linguistics.

Meng Cao, Yue Dong, Jiapeng Wu, and Jackie Chi Kit Cheung. 2020. Factual error correction for abstractive summarization models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6251–6258, Online. Association for Computational Linguistics.

Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. 2018. Faithful to the original: Fact aware neural abstractive summarization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Moye Chen, Wei Li, Jiachen Liu, Xinyan Xiao, Hua Wu, and Haifeng Wang. 2021. SgSum:transforming multi-document summarization into sub-graph selection. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4063–4074, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

Marco Damonte, Rahul Goel, and Tagyoung Chung. 2019. Practical semantic parsing for spoken language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Industry Papers)*, pages 16–23, Minneapolis, Minnesota. Association for Computational Linguistics.

Shibhansh Dohare, Vivek Gupta, and Harish Karnick. 2018. Unsupervised semantic abstractive summarization. In *Proceedings of ACL 2018, Student Research Workshop*, pages 74–83, Melbourne, Australia. Association for Computational Linguistics.

Markus Dreyer, Mengwen Liu, Feng Nan, Sandeep Atluri, and Sujith Ravi. 2021. Analyzing the abstractiveness-factuality tradeoff with nonlinear abstractiveness constraints.

Esin Durmus, He He, and Mona Diab. 2020. FEQA: A question answering evaluation framework for faithfulness assessment in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5055–5070, Online. Association for Computational Linguistics.

Alexander R. Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. SummEval: Re-evaluating Summarization Evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409.

Tobias Falke, Leonardo F. R. Ribeiro, Prasetya Ajie Utama, Ido Dagan, and Iryna Gurevych. 2019. Ranking generated summaries by correctness: An interesting but challenging application for natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2214–2220, Florence, Italy. 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 4186–4196, Hong Kong, China. Association for Computational Linguistics.

Jeffrey Flanigan, Sam Thomson, Jaime Carbonell, Chris Dyer, and Noah A. Smith. 2014. A discriminative graph-based parser for the Abstract Meaning Representation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1426–1436, Baltimore, Maryland. Association for Computational Linguistics.

Qiankun Fu, Linfeng Song, Wenyu Du, and Yue Zhang. 2021. End-to-end AMR coreference resolution. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4204–4214, Online. Association for Computational Linguistics.

Ben Goodrich, Vinay Rao, Peter J. Liu, and Mohammad Saleh. 2019. Assessing the factual accuracy of generated text. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery; Data Mining*, KDD '19, page 166–175, New York, NY, USA. Association for Computing Machinery.

Tanya Goyal and Greg Durrett. 2020. Evaluating factuality in generation with dependency-level entailment. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3592–3603, Online. Association for Computational Linguistics.

Tanya Goyal and Greg Durrett. 2021. Annotating and modeling fine-grained factuality in summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1449–1462, Online. Association for Computational Linguistics.

Zhijiang Guo, Yan Zhang, Zhiyang Teng, and Wei Lu. 2019. Densely connected graph convolutional networks for graph-to-sequence learning. *Transactions of the Association for Computational Linguistics*, 7:297–312.

Hardy Hardy and Andreas Vlachos. 2018. Guided neural language generation for abstractive summarization using Abstract Meaning Representation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 768–773, Brussels, Belgium. Association for Computational Linguistics.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 1693–1701, Cambridge, MA, USA. MIT Press.

Or Honovich, Leshem Choshen, Roee Aharoni, Ella Neeman, Idan Szpektor, and Omri Abend. 2021. Q^2: Evaluating factual consistency in knowledge-grounded dialogues via question generation and question answering. *CoRR*, abs/2104.08202.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

Pavan Kapanipathi, Ibrahim Abdelaziz, Srinivas Ravishankar, Salim Roukos, Alexander Gray, Ramón Fernandez Astudillo, Maria Chang, Cristina Cornelio, Saswati Dana, Achille Fokoue, Dinesh Garg, Alfio Gliozzo, Sairam Gurajada, Hima Karanam, Naweed Khan, Dinesh Khandelwal, Young-Suk Lee, Yunyao Li, Francois Luus, Ndivhuwo Makondo, Nandana Mihindukulasooriya, Tahira Naseem, Sumit Neelam, Lucian Popa, Revanth Gangi Reddy, Ryan Riegel, Gaetano Rossiello, Udit Sharma, G P Shrivatsa Bhargav, and Mo Yu. 2021. Leveraging Abstract Meaning Representation for knowledge base question answering. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3884–3894, Online. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Kevin Knight, Bianca Badarau, Laura Baranescu, Claire Bonial, Madalina Bardocz, Ulf Hermjakob Kira Griffitt, Daniel Marcu, Martha Palmer, Tim O'Gorman, and Nathan Schneider. 2020. Abstract meaning representation (AMR) annotation release 3.0.

Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346, Online. Association for Computational Linguistics.

Faisal Ladhak, Esin Durmus, He He, Claire Cardie, and Kathleen McKeown. 2021. Faithful or extractive? on mitigating the faithfulness-abstractiveness trade-off in abstractive summarization.

Fei-Tzin Lee, Chris Kedzie, Nakul Verma, and Kathleen McKeown. 2021. An analysis of document graph construction methods for AMR summarization.

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*, pages 7871–7880.

Wei Li, Xinyan Xiao, Jiachen Liu, Hua Wu, Haifeng Wang, and Junping Du. 2020. Leveraging graph to improve abstractive multi-document summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6232–6243, Online. Association for Computational Linguistics.

Kexin Liao, Logan Lebanoff, and Fei Liu. 2018. Abstract Meaning Representation for multi-document summarization. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1178–1190, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Jungwoo Lim, Dongsuk Oh, Yoonna Jang, Kisu Yang, and Heuiseok Lim. 2020. I know what you asked: Graph path learning using AMR for commonsense reasoning. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2459–2471, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Fei Liu, Jeffrey Flanigan, Sam Thomson, Norman Sadeh, and Noah A. Smith. 2015. Toward abstractive summarization using semantic representations. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1077–1086, Denver, Colorado. Association for Computational Linguistics.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

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

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Caglar Gulcehre, 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, Berlin, Germany. Association for Computational Linguistics.

Feng Nan, Cicero Nogueira dos Santos, Henghui Zhu, Patrick Ng, Kathleen McKeown, Ramesh Nallapati, Dejiao Zhang, Zhiguo Wang, Andrew O. Arnold, and Bing Xiang. 2021. Improving factual consistency of abstractive summarization via question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6881–6894, Online. Association for Computational Linguistics.

Tahira Naseem, Austin Blodgett, Sadhana Kumaravel, Tim O'Gorman, Young-Suk Lee, Jeffrey Flanigan, Ramón Fernandez Astudillo, Radu Florian, Salim Roukos, and Nathan Schneider. 2021. DocAMR: Multi-sentence AMR representation and evaluation.

Juri Opitz, Angel Daza, and Anette Frank. 2021. Weisfeiler-Leman in the BAMBOO: Novel AMR graph metrics and a benchmark for AMR graph similarity. *Transactions of the Association for Computational Linguistics*.

Juri Opitz, Letitia Parcalabescu, and Anette Frank. 2020. AMR similarity metrics from principles. *Transactions of the Association for Computational Linguistics*, 8:522–538.

Artidoro Pagnoni, Vidhisha Balachandran, and Yulia Tsvetkov. 2021. Understanding factuality in abstractive summarization with FRANK: A benchmark for factuality metrics. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4812–4829, Online. Association for Computational Linguistics.

Ramakanth Pasunuru, Mengwen Liu, Mohit Bansal, Sujith Ravi, and Markus Dreyer. 2021. Efficiently summarizing text and graph encodings of multi-document clusters. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4768–4779, Online. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on*

Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Leonardo F. R. Ribeiro, Claire Gardent, and Iryna Gurevych. 2019. Enhancing AMR-to-text generation with dual graph representations. 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 3183–3194, Hong Kong, China. Association for Computational Linguistics.

Leonardo F. R. Ribeiro, Jonas Pfeiffer, Yue Zhang, and Iryna Gurevych. 2021a. Smelting gold and silver for improved multilingual AMR-to-Text generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 742–750, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Leonardo F. R. Ribeiro, Yue Zhang, Claire Gardent, and Iryna Gurevych. 2020. Modeling global and local node contexts for text generation from knowledge graphs. *Transactions of the Association for Computational Linguistics*, 8:589–604.

Leonardo F. R. Ribeiro, Yue Zhang, and Iryna Gurevych. 2021b. Structural adapters in pretrained language models for AMR-to-text generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Punta Cana, November 7-11, 2021*.

Michael Sejr Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings*, pages 593–607.

Linfeng Song, Daniel Gildea, Yue Zhang, Zhiguo Wang, and Jinsong Su. 2019. Semantic neural machine translation using AMR. *Transactions of the Association for Computational Linguistics*, 7:19–31.

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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. Asking and answering questions to evaluate the factual consistency of summaries. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5008–5020, Online. Association for Computational Linguistics.

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

Wenhao Wu, Wei Li, Xinyan Xiao, Jiachen Liu, Ziqiang Cao, Sujian Li, Hua Wu, and Haifeng Wang. 2021. BASS: Boosting abstractive summarization with unified semantic graph. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6052–6067, Online. Association for Computational Linguistics.

Yuexiang Xie, Fei Sun, Yang Deng, Yaliang Li, and Bolin Ding. 2021. Factual consistency evaluation for text summarization via counterfactual estimation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 100–110, Punta Cana, Dominican Republic. 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 *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 11328–11339. PMLR.

Chenguang Zhu, William Hinthorn, Ruochen Xu, Qingkai Zeng, Michael Zeng, Xuedong Huang, and Meng Jiang. 2021. Enhancing factual consistency of abstractive summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 718–733, Online. Association for Computational Linguistics.

Appendices

In this supplementary material, we detail experiments' settings, additional model evaluations and additional information about semantic graph representations.

A Details of Models and Hyperparameters

The experiments were executed using the version 3.3.1 of the *transformers* library released by Hugging Face (Wolf et al., 2019). In Table 8, we

Computing Infrastructure	32GB NVIDIA V100 GPU
Optimizer	Adam
Optimizer Params	$\beta = (0.9, 0.999), \epsilon = 10^{-8}$
learning rate	1e-4
Learning Rate Decay	Linear
Weight Decay	0
Warmup Steps	0
Maximum Gradient Norm	1
batch size	4
epoch	10
Adapter dimension	32
# document graphs (k)	5

Table 8: Hyperparameter settings for our methods.

report the hyperparameters used to train FACTGRAPH. We use the Adam optimizer (Kingma and Ba, 2015) and employ a linearly decreasing learning rate schedule without warm-up. Mean pooling is used to calculate the final representation of each graph.

Structural Adapters' Pretraining. The structural adapters are pretrained using AMR graphs from the release 3.0 (LDC2020T02) of the AMR annotation corpus (Knight et al., 2020).[10] Similarly to the masked language modeling objective, we execute self-supervised node-level prediction, where we randomly mask and classify AMR nodes. The goal of this pretraining phase is to capture domain specific AMR knowledge by learning the regularities of the node/edge attributes distributed over graph structure.

Text Adapters' Pretraining. The text adapters are pretrained using synthetically created data, which is generated by applying a series of rule-based transformations to the sentences of source documents (Kryscinski et al., 2020). The pretraining task is to classify each summary sentence as factual or non-factual. The goal of this pretraining phase is to learn suitable text representations to better identify whether summary sentences remain factually consistent to the input document after the transformation.

B Speed Comparison

FACTGRAPH encodes the structured semantic representations that encode facts from the document and summary. Despite their effectiveness, extracting semantic graphs, such as AMR, is computationally expensive because current models employ Transformer-based encoder-decoder architectures based on Transformers and pretrained language models.

[10] https://catalog.ldc.upenn.edu/LDC2020T02

	Preprocessing	Inference
DAE	135.8	62.6
FACTGRAPH-E - **Parser1**	427.9	79.0
FACTGRAPH-E - **Parser2**	1332.2	75.4

Table 9: Speed Comparison. Execution time is measured in seconds.

Graph Type	BACC	F1
Only text	88.4	88.6
FACTGRAPH-Dependency	90.2	90.3
FACTGRAPH-OpenIE	90.5	90.7
FACTGRAPH-AMR	91.2	91.3

Table 10: Effect of different graph representations in the factuality model (on the dev set of FACTCOLLECT).

In this experiment, we compare the time execution of FACTGRAPH-E and DAE in a sample of 1000 datapoints extracted from the XSum test set. In order to extract the semantic graphs, we investigate two AMR parsers, **Parser1**: a dual graph-sequence parser that iteratively refines an incrementally constructed graph (Cai and Lam, 2020) , and **Parser2**: a linearized graph model that employs BART (Bevilacqua et al., 2021). The execution of the AMR parsers is parallelized using four Tesla V100 GPUs. We use Parser2 for the experiments in this paper since it is the current state of the art in AMR parsing, although it is slower in preprocessing than Parser1 is.

As shown in Table 9, DAE's preprocessing is much faster compared to this phase in FACTGRAPH-E, since DAE employs a fast enhanced dependency model from the Stanford CoreNLP tool (Manning et al., 2014). This model builds a parse by performing a linear-time scan over the words of a sentence. Finally, note that FACTGRAPH is slower than DAE in inference because it employs adapters and encodes both graphs and texts from the document and summary, whereas the DAE model encodes only the texts.

C Comparing Semantic Representations for Factuality Evaluation

OpenIE graph-based structures were used in order to improve factuality in abstractive summarization (Cao et al., 2018), whereas dependency arcs were shown to be beneficial for evaluating factuality (Goyal and Durrett, 2020). We thus investigate different graph-based meaning representations

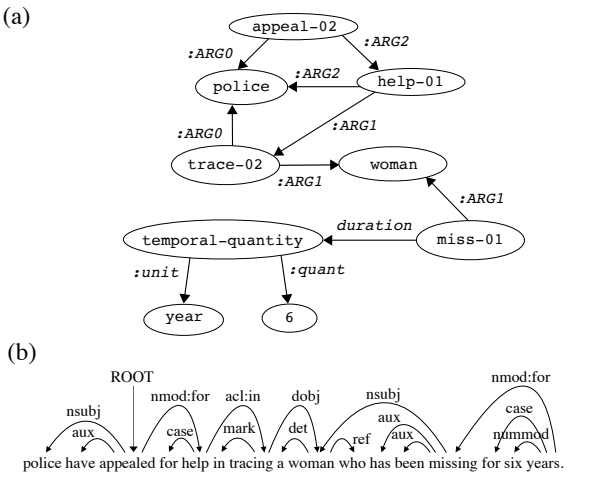

Figure 5: (a) AMR and (b) dependency representations for the summary *"police have appealed for help in tracing a woman who has been missing for six years."*

using FACTGRAPH. AMR is a more logical representation that models relations between core concepts, and has a rough alignment between nodes and spans in the text. Conversely, dependencies capture more fine-grained relations between words, and all words are mapped into nodes in the dependency graph. OpenIE constructs a graph with node descriptions similar to the original text and uses open-domain relations, leading to relations that are hard to compare.

As shown in Table 10, whereas OpenIE performs slightly better than dependency graphs, AMR gives the best results according to the two metrics, highlighting the potential use of AMRs in representing salient pieces of information. Different from our work, Lee et al. (2021) and Naseem et al. (2021) propose a graph construction approach which generates a single document-level graph created using the individual sentences' AMR graphs by merging identical concepts – this is orthogonal to our sentence-level AMR representation and can be incorporated in future work.

D Semantic Representations

In Figure 5 we show AMR and dependency representations for the summary sentence *"police have appealed for help in tracing a woman who has been missing for six years."*. In §5.5 those semantic representations are used to predict subsentence-level factuality using edge-level information. In particular, FACTGRAPH-E employs AMR (Figure 5a) whereas DAE uses dependencies (Figure 5b).