Jointly Learning Guidance Induction and Faithful Summary Generation via Conditional Variational Autoencoders

Wang Xu and Tiejun Zhao*
Harbin Institute of Technology, Harbin, China
xuwang@hit-mtlab.net, tjzhao@hit.edu.cn

Abstract

Abstractive summarization can generate high quality results with the development of the neural network. However, generating factual consistency summaries is a challenging task for abstractive summarization. Recent studies extract the additional information with off-the-shelf tools from the source document as a clue to guide the summary generation, which shows effectiveness to improve the faithfulness. Unlike these work, we present a novel framework based on conditional variational autoencoders, which induces the guidance information and generates the summary equipped with the guidance synchronously. Experiments on XSUM and CNNDM dataset show that our approach can generate relevant and fluent summaries which is more faithful than the existing state-of-the-art approaches, according to multiple factual consistency metrics.

1 Introduction

Document summarization aims to produce the shorter version of a document while preserving salient information, which helps people out of the information explosion (Mihalcea and Tarau, 2004; Daumé III and Marcu, 2006; Allahyari et al., 2017). Compared with extractive summarization that retrieves essential sentences from the source document, abstractive summarization has no constraint on the words and phrases, which has attracted more attention. With the development of neural network and the large pre-trained language models, systems can generate summarizes with a high level fluency and coherence (Devlin et al., 2019; Dong et al., 2019; Lewis et al., 2020; Zhang et al., 2020a).

Generating faithful summaries is a challenging task for abstractive summarization (Kryscinski et al., 2020; Maynez et al., 2020; Gabriel et al., 2021; Zhou et al., 2021). Previous studies have shown that the generated summaries distort or fabricate the facts of the source document, which also refers to the hallucination phenomenon (Huang et al., 2021). It statistics that most models produce 80% summaries with factual errors in XSUM dataset (Narayan et al., 2018) which limits the usage of summarization system (Pagnoni et al., 2021).

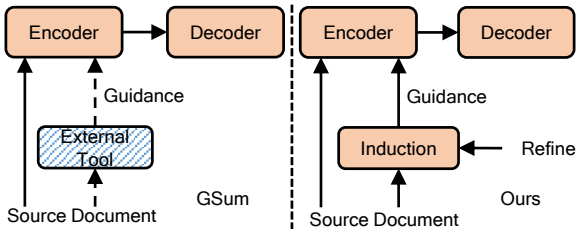

Figure 1: Our framework trains guidance induction and summary generation jointly. It avoids the domain mismatch of the external tools and the guidance extraction is refined during training.

Recent studies provide different guidance information as input to enhance the factual consistency of the summary (Cao et al., 2018; Zhu et al., 2021). Generally, these models act a separate two-stage processing, the guidance extracting by off-the-shelf tools and summary generation conditioned on source document and guidance. Typically, Dou et al. (2021) propose an extensible guided summarization framework GSum, which has achieved impressive results. It uses an oracle to select guidance during training and extracts the keywords by out-of-box tools (Li et al., 2018) at test time. Then two transformers (Vaswani et al., 2017) are used to encode the source document and guidance.

However, the performances of separate two-stage processing models are limited by the external tools which may suffer from domain mismatch. In fact, the experiments of GSum have shown that the performance would have a significant gain when

*Corresponding author

Findings of the Association for Computational Linguistics: NAACL 2022, pages 2340 - 2350

the model uses an oracle to select guidance in testing, rather than the external tools. Moreover, the inaccuracy of the guidance extraction leads to the unfaithfulness of the summary.

In this paper, we present a novel framework which trains **G**uidance **I**nduction and **S**ummary **G**eneration (**GISG**) jointly via conditional variational autoencoder. Specifically, we use phrases as the information granularity of our guidance and we induce the *keyphrases* of the source document, which appear in the summary semantically. First, we extract all phrases from the source document by part-of-speech tagger as candidates and we use latent variables to indicate the keyphrases. Then we learn to induce the latent variables and generate the summary jointly. Our approach avoids the domain mismatch of the external tools while the guidance extraction is refined during training. Then the faithful summaries are generated conditioned on the accurate guidance information.

Experiments on XSUM (Narayan et al., 2018) and CNNDM (Hermann et al., 2015) datasets show that our approach can generate relevant and fluent summaries which is more faithful to the source document than existing state-of-the-art approaches, according to multiple factual consistency metrics.

2 Related Work

2.1 Abstractive Summarization

Abstractive Summarization is prone to generate factual inconsistency text with the source document (Durmus et al., 2020; Gabriel et al., 2021). Recent studies divide factual inconsistency error into two categories, intrinsic error and extrinsic error separately (Zhou et al., 2021). The intrinsic error refers to the error which is contradicted to the source document. And the extrinsic error refers to the error which is neither supported nor contradicted by the source document. Recent efforts for improving factual consistency are mainly categorized into factual guidance methods, contrastive learning methods and post-edit-based methods.

Factual guidance methods provide the models with additional information for the encoder, including the relation triples, keywords and important sentences, which guide summarization systems to pay attention to the facts and to reduce consistent error (Cao et al., 2018; Xu et al., 2021b,a; Dou et al., 2021). Zhu et al. (2021) explore using knowledge graphs to model the facts to the source document. Dou et al. (2021) design a unified framework to introduce different information by an additional transformer encoder.

Contrastive learning methods encourage models to distinguish between positive and negative examples (Nan et al., 2021; Cao and Wang, 2021; Liu et al., 2021; Xu et al., 2022). Nan et al. (2021) generate multiple summaries candidates by sampling from the pre-trained models and selecting positive and negative examples according to the question answer based metric. Cao and Wang (2021) construct positive and negative examples by the heuristic rules, for example, replacing the entity in the references or paraphrasing the references.

Post-edit based method aims to apply a correction over the generated results to obtain more factual-consistent summarization (Dong et al., 2020; Cao et al., 2020; Chen et al., 2021a). Dong et al. (2020) leverages the question answering models to correct the factual error iteratively via span selection over the generated summaries. Cao et al. (2020) propose a corrector model to identify and correct factual errors in generated summaries. The model is trained on the synthesis data which is transformed from the reference summaries.

2.2 Conditional Variational Autoencoder

The variational auto-encoder (VAE) is a directed graphical model with certain types of latent variables, such as Gaussian latent variables (Kingma and Welling, 2014; Sohn et al., 2015; Rezende et al., 2014). A generative process of the VAE contains two stages; a set of latent variables are generated from the prior distribution and the data is generated by the generative distribution conditioned on latent variables.

Conditional VAE (CVAE) (Sohn et al., 2015; Zhao et al., 2017; Chen et al., 2021b) is a recent modification of VAE to generate diverse example conditioned on additional constrained information. Instead of providing additional information in the output, CVAE models introduce latent variables to represent the information. Inspired by CVAE, we view the keyphrases as the conditional attributes and adapt CVAE to train keyphrases induction and faithful summarization generation jointly.

3 Background

Given the source input document $X = \{X_1, X_2 \cdots X_N\}$, of length N. The task of abstractive summarization is to generate a short

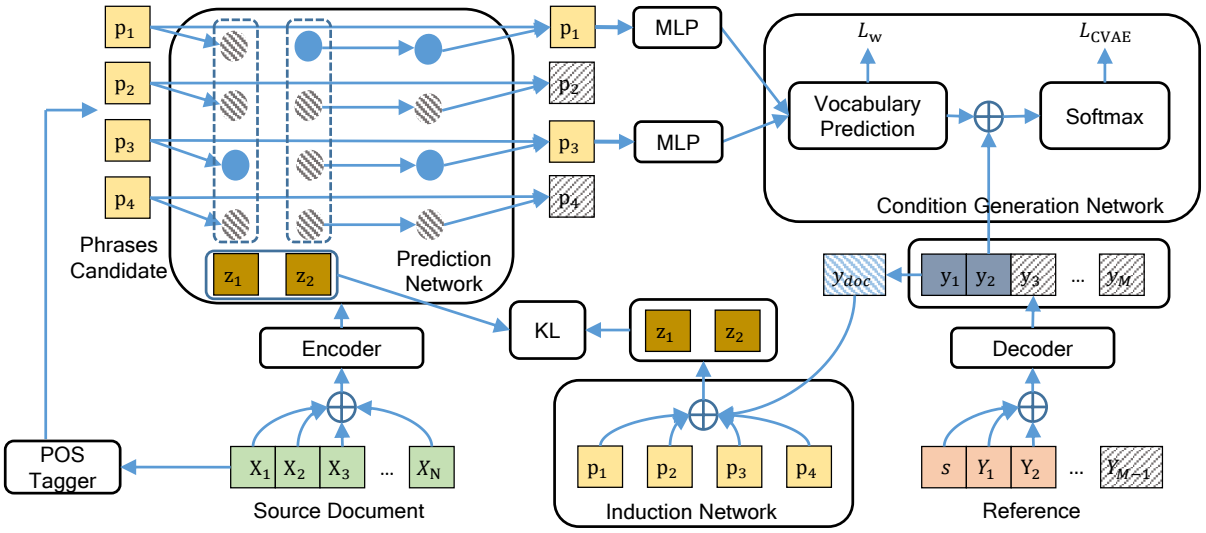

Figure 2: General framework of our model. There are mainly three parts, keyphrases prediction network, induction network and condition generation network.

version of the source document, i.e. $Y = \{Y_1, Y_2 \cdots, Y_M\}$, where M is the length of summary. Each token X_n, Y_m takes one value from a vocabulary $\mathcal{V}$.

Abstractive summarization is generally formulated as $P(Y|X) = \sum_{t=1}^{M} P(Y_t \mid Y_{<t}, X)$, which is a typical sequence to sequence generation problem. We use BART (Lewis et al., 2020) which is based on Transformer-based encoder and decoder architectures (Vaswani et al., 2017) as our backbone. Transformer layers use multi-heads self-attention to capture the dependency between the input (Vaswani et al., 2017). Concretely, the input X is converted into a vector sequence $\boldsymbol{X} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_N\}$ by the encoder, where $\boldsymbol{x}_n \in \mathbb{R}^h$ and h is the size of hidden representation. In decoding step t, the decoder generates the t word representation $\boldsymbol{y}_t$ by attending to the input contextual representation $\boldsymbol{X}$ and the prefix words $\{Y_1, \cdots, Y_{t-1}\}$ through the encoder-decoder attention. The probability of predicting the next token Y_t from the vocabulary $\mathcal{V}$ is

$$P(Y_t|Y_{<t}, X) = \text{softmax}(\boldsymbol{E}\boldsymbol{y}_t) \quad (1)$$

where $\boldsymbol{E} \in \mathbb{R}^{|\mathcal{V}| \times h}$ is the embedding matrix of the vocabulary.

4 Methodology

4.1 Summarization with Conditional Variational Autoencoders

Previous work uses external tools to extract the guidance (e.g. keyphrases, important sentences or relation triplets) and generate the summaries conditioned on the source document and the guidance. Our idea is to induce the guidance and generate the summary jointly. Phrases are the meaning semantic information unit of the document, which is important to express the facts of the document. Compared with a single word or a sentence of the document, a phrase contains more abundant and accurate information and is refined without lots of useless information. We will use the phrases as the information granularity of our guidance and our framework can easily be generalized to the sentence or the relation triplets.

We extract all phrases from the source document as the candidates, since the keyphrases are the subset of the phrases of the document. Then we assume a latent variable Z to indicate the keyphrases set.

Based on CVAE, we introduce an induction network $Q(Z|X, Y)$ to approximate the true posterior distribution $P(Z|X, Y)$. Sohn et al. (2015) have shown that the variational lower bound can be written as:

$$\begin{aligned} \mathcal{L}_{\text{CVAE}} &= \text{KL}(Q(Z|X, Y)||P(Z|X)) \\ -\mathbb{E}_Q(\log P(Y|X, Z)) &\geq -\log P(Y|X) \end{aligned} \quad (2)$$

Thus, we jointly learn the keyphrases prediction $P(Z|X)$ and summary generation $P(Y|X, Z)$. Intuitively, the term $\mathbb{E}_Q(\log P(Y|X, Z))$ ensures the model generates the summary conditioned on X and Z, while the KL diversity term tries to guide the keyphrases prediction $P(Z|X)$ approximate the induction $Q(Z|X, Y)$.

When the model is evaluated, a latent variable Z is first predicted from $P(Z|X)$. Then the decoder $P(Y|X,Z)$ generates the summaries conditioned on X and Z.

We will describe our approach in detail in the following sections. The overview of our framework is in Figure 2. First, we describe candidate phrases extraction in Section 4.2. In Section 4.3, we present the prediction network and keyphrases induction network. Section 4.4 further presents the conditional summary generation network.

4.2 Candidate Phrases Extraction

We extract the phrases from the source document including the noun and verb phrases. Following Wu et al. (2021)'s work, we use the rule-based matchers to extract noun and verb phrases by the part-of-speech[1] (POS). Concretely, we use SpaCy (Montani et al., 2020) to obtain the POS tag of each word. The noun phrases are extracted by the built-in function of Spacy. And a phrase will be treated as the verb phrase if any of the cases are satisfied. 1). `[AUX] VERB`. The words with the verb POS tag are extracted besides the auxiliary verb. 2). `VERB [RP]`. A verb phrase may be followed by the particle including prepositions or adverbs (e.g., walk down). 3). `AUX not VERB [RP]`. "not" is considered to handle negation (e.g., would not find). And we filter out the phrases that contain less than three words. The extracted context phrases of the source document are treated as the phrases candidates.

Although, we also use the external tools POS tagger to extract the phrases of the source document, we do not directly use the phrases to guide the summary generation. Generally, the keyphrases are only a small subset of the candidates extracted by the POS tagger. We believe that our approach is robust even with an inaccurate POS tagger.

4.3 Keyphrases Prediction and Induction

We use the output of the encoder to obtain the phrase representation by averaging the representation of the corresponding words. Specially, suppose a phrase is $X_{s:t}$. The representation of the phrase is $\boldsymbol{q} = \frac{1}{t-s}\sum_{k=s}^{t}\boldsymbol{x}_k$. Thus, we get the representation of the candidates $\boldsymbol{Q} = \{\boldsymbol{q}_1, \boldsymbol{q}_2, \cdots, \boldsymbol{q}_I\}$, where $\boldsymbol{Q} \in \mathbb{R}^{I\times h}$ and I is the number of the phrases candidates.

[1] https://spacy.io/usage/linguistic-features

Generally, every phrase candidate is assigned a latent variable to indicate whether the phrase is the keyphrase and the selection of each phrase is a binary classification problem. However, we find that the models tend to select redundancy phrases or even all the candidates. We argue that it is because the candidates contain similar phrases and the binary classification would lead to repetition without being constrained with the number of the phrases.

To solve the problem, we use the latent variable to select the keyphrases from the candidates. Formally, we assume the maximum of keyphrases in a document is B. We define $\boldsymbol{Z} = \{z_1, z_2, \cdots, z_B\}$ as a latent indicator variable, where $\boldsymbol{Z} \in \mathbb{R}^{B\times I}$ and z_i is one-hot vector. $z_i^j = 1$ means the phrase j is the ith keyphrase. The model can select less than B keyphrases by having the repetition latent value in $\boldsymbol{Z}$.

Then we have the prediction network and keyphrases induction network as follows:

$$\begin{aligned} Q(z_i|X,Y) &= \text{softmax}(\text{MLP}_1^i(\boldsymbol{Q})\boldsymbol{y}_{\text{doc}}^T) \\ P(z_i|X) &= \text{softmax}(\text{MLP}_2^i(\boldsymbol{Q})) \end{aligned} \quad (3)$$

where $\boldsymbol{y}_{doc}$ is the representation of the summary. $\boldsymbol{y}_{doc}$ is obtained by averaging $\{\boldsymbol{y}_1 \cdots \boldsymbol{y}_M\}$.

4.4 KeyPhrases Guide Summary Generation

We calculate the distribution of the word by attending to the source contextual representation and the keyphrases representation for the generation network $P(Y|X,Z)$. Similar to Aralikatte et al. (2021)'s work, we introduce a bias in Eqn. (1) to help the model focus on the keyphrases.

Formally, the generation probability of Y_t is formulated as:

$$P(Y_t|Y_{<t}, X, Z) = \text{softmax}(\boldsymbol{y}_t\boldsymbol{E} + \boldsymbol{f}_t\boldsymbol{E}) \quad (4)$$

where $\boldsymbol{f}_t = \boldsymbol{S}\boldsymbol{A}\text{MLP}_3(\boldsymbol{Q})$ and $\boldsymbol{S} \in \mathbb{R}^{1\times I}$ is the selection vector. $\boldsymbol{S}_i = 1$ means the ith candidate is selected as the keyphrases. $\boldsymbol{A}$ is the attention score over the selected keyphrases and $\boldsymbol{A} = \text{softmax}(\text{MLP}_3(\boldsymbol{Q})\boldsymbol{y}_t^T)$.

Basically, $\boldsymbol{S}$ is obtained from the $Q(Z|X,Y)$ during training. As Eqn. (2) indicated, we need to calculate the expectation of $P(Y|X,Z)$ over the distribution $Q(Z|X,Y)$. We use the Gumbel-Softmax trick (Jang et al., 2017) to sample from $Q(Z|X,Y)$ and obtain low-variance gradients. Concretely, the sample probability $\hat{Q}_i$ is

Dataset	Pairs			Tokens	
	Train	Valid	Test	Doc	Sum
XSUM	203028	11273	11332	430.2	23.3
CNNDM	287084	13367	11489	766.1	58.2

Table 1: Statistics of the dataset with respect to corpus size of training, validation and test set, average document (source) and summary (reference) length (in terms of tokens).

as following:

$$\hat{Q}_i = \text{softmax}(\frac{\text{MLP}_1^i(\boldsymbol{Q})\boldsymbol{y}_{doc}^T + \epsilon}{\tau}) \quad (5)$$

where ϵ is Gumbel noise and τ is temperature. Thus $\boldsymbol{S} = \sum_{i=1}^{B} \hat{Q}_i$. To avoid repetition selection among latent variables, we normalize the $\hat{S} = \frac{\boldsymbol{S}}{max(\boldsymbol{S})}$. In this way, the model is encouraged to extract different keyphrases, otherwise only one keyphrase is selected.

During testing, we obtain $\boldsymbol{S}$ from $P(z_i|X)$. $\boldsymbol{S} = \sum_{i=1}^{B} \text{one\_hot}(\text{argmax}(P(z_i|X)))$. The upper value of $\boldsymbol{S}$ is clipped into 1 to avoid repetition selection.

The vanishing latent variable problem (Bowman et al., 2016; Lucas et al., 2019) exists when training with VAE. There are multiple techniques to address the problem (Zhao et al., 2017; Zhu et al., 2020). Following Zhao et al. (2017)'s work, we introduce an auxiliary loss encouraging the keyphrases to predict the words of the summary. The auxiliary loss would guide the selected phrase representation to better represent the content of the summary. Then additional loss is following:

$$\begin{aligned}\mathcal{L}_{\text{w}} = \frac{1}{|\mathcal{V}|}\sum_{i=1}^{|\mathcal{V}|}\{|v_i \in \hat{Y}|\log(\sigma\frac{e^{\boldsymbol{f}_{key}\boldsymbol{E}_i}}{\sum e^{\boldsymbol{f}_{key}\boldsymbol{E}_i}}) \\ +|v_i \notin \hat{Y}|\log(1-\sigma\frac{e^{\boldsymbol{f}_{key}\boldsymbol{E}_i}}{\sum e^{\boldsymbol{f}_{key}\boldsymbol{E}_i}})\}\end{aligned} \quad (6)$$

where $\boldsymbol{f}_{\text{key}} = \boldsymbol{S}\text{MLP}_3(\boldsymbol{Q})$. $\hat{Y}$ is the target summary and σ is Sigmoid function. Then our final loss function is:

$$\mathcal{L} = \mathcal{L}_{\text{CVAE}} + \lambda\mathcal{L}_{\text{w}} \quad (7)$$

5 Experiments

5.1 Setup

Datasets. We evaluate our models on extreme document summarization (XSUM) (Narayan et al., 2018) and CNN/Daily Mail (CNNDM) (Hermann et al., 2015). Both of the datasets are extracted from the news and the detailed statistics of the datasets are listed in Table 1. In XSUM dataset, the documents are summarized into single-sentence summaries. These summaries demonstrate a high level of abstraction which requires document-level inference, abstraction, and paraphrasing. CNNDM is a high quality summarization dataset consisting of news articles and human annotation summaries.

Implementation Details. We introduce our framework into **BART** (Lewis et al., 2020) which is a strong abstractive summarization model pretrained with a denoising autoencoding objective. We use the FairSeq[2] as the implementation of our baseline and model. We inherit their provided hyper-parameters of XSUM and CNNDM. Concretely, the total number of the updates is 1.5w in XSUM and 2w in CNNDM. The maximum number of tokens in a batch is 4096 with gradient accumulation steps of 4. We use Adam optimizer and the learning rate is set to 3e-5. The ϵ is 1e-8 and β is (0.9, 0.999). The maximum of the keyphrases B is set to 8. And the temperature τ is set to 0.1 for Gumbel-Softmax during training. We use mixed-precision to speed up model training and the warm-up is set to 500 steps. All the experiments are done on 2 and 4 NVIDIA 3090 in XSUM and CNNDM. For the beam search, the minimum summary length is 11 and 56 for XSUM and CNNDM, respectively. The number of beams is 4 for XSUM and 6 for CNNDM. And the ROUGE-L score on the validation set is used to pick the best model.

Evaluation Metrics. ROUGE[3] (Lin and Hovy, 2003) considers lexical overlap against the reference summaries, which is widely used to evaluate the informativeness and fluency of the summary. We report on ROUGE-1, ROUGE-2 and ROUGE-L to measure summary qualities.

We also use **BERTScore**[4] (Zhang et al., 2020b) to evaluate the semantic similarity between a hypothesis and the reference summary by contextual representation.

However ROUGE and BERTScore perform poorly in capturing factual consistency with the source document. Recent studies have developed different categories to evaluate the faithfulness of a generated summary given its

[2] https://github.com/pytorch/fairseq
[3] https://github.com/pltrdy/files2rouge
[4] https://github.com/Tiiiger/bert_score

Models	Lexical Overlap			Semantic	Relation Triplet		QA-based	
	R1	R2	RL	BERTScore	Close	Open	QAGS	QuesEval
XSUM								
EXTORA* (Narayan et al., 2018)	29.82	8.83	22.68	85.74	18.57	72.46	69.20	45.84
FASum* (Zhu et al., 2021)	30.28	10.03	23.76	88.03	1.63	0.36	11.13	31.18
GSum* (Dou et al., 2021)	44.93	21.19	35.96	90.41	1.92	1.75	13.58	36.90
BART* (Lewis et al., 2020)	45.49	21.82	36.69	90.83	1.89	2.02	13.76	36.91
BART†	44.65	21.28	36.09	90.62	2.03	2.11	13.76	37.07
GISG (ours)	**45.54**	**21.99**	**36.82**	**92.11**	**2.14**	**2.18**	**14.31**	**37.09**
CNNDM								
MATCH* (Zhong et al., 2020)	44.47	20.92	40.05	87.32	50.92	89.21	77.70	59.93
FASum* (Zhu et al., 2021)	40.53	17.84	37.40	87.86	38.56	67.82	71.33	57.58
GSum* (Dou et al., 2021)	**45.89**	**22.27**	**42.68**	**88.64**	41.23	70.69	71.11	57.54
BART* (Lewis et al., 2020)	44.25	21.11	41.16	88.33	41.40	70.79	71.89	57.65
BART†	44.11	21.16	40.55	88.13	39.16	69.69	71.25	57.80
GISG (ours)	44.50	21.45	41.05	88.56	**42.07**	**70.83**	**72.34**	**58.52**

Table 2: Main results. MATCHSum (denoted by MATCH) and EXT-ORACLE (denoted by EXTORA) are extractive summarization models. The results with * are computed based on the output files in the EXPLAINABOARD. The results with † are our reimplement of the baseline models. Bold indicates the best performance in the abstractive summarization models groups.

source document (Zhang et al., 2020b; Dong et al., 2020; Liu et al., 2021). Our approach is evaluated with four factual consistency metrics, including relation triplets based and question answering (QA) based metrics.

Relation triplets based metrics evaluate factual consistency by comparing structured data of factual information extracted from the summary and the source document. **Close Scheme Fact Triple** (Dong et al., 2020) extract (Subject, Relation, Object) by named entity recognition and relation extraction models and then calculate the precision between the triples extracted from the summary and source document. **Open Scheme Fact Triple** (Dong et al., 2020) extract the text spans to indicate the relation and calculate the precision similar to close scheme fact triple.

Question answering based metrics use the pretrained QA model to evaluate the faithfulness. **QAGS** (Wang et al., 2020) extracts text spans from the predicted summary and generates questions conditioned on the predicted summary by a trained question generation model. A pretrained QA model answers the questions from the document. Then the matching score is calculated between the answer from the document and the summary. **QuestEval** (Rebuffel et al., 2021) not only generate (question, answer) pairs from the summary, but also from the source document, which considers to measure the recall performance.

We use factsumm[5] and OpenIE[6] to calculate close scheme fact triple and open scheme fact triple. And we use the repository to calculate QAGS[5] and QuesEval[7]. We only calculate factual consistency metrics of 1k (document, reference, summary) for the computing efficiency.

Competing Methods. We compare our model with some competing methods, including extractive and abstractive summarization models. **EXT-ORACLE** (Narayan et al., 2018) and **MATCHSUM** (Zhong et al., 2020) are extractive models. EXT-ORACLE selects a single best sentence of the document by referring to the target. MATCHSUM reranks the candidate summaries produced by BertExt (Liu and Lapata, 2019) and achieves state-of-the-art extractive results on various summarization datasets. For abstractive summarization models, **FASum** (Zhu et al., 2021) and **GSum** (Dou et al., 2021) are models designed for faithful summarization. FASum extracts the relation triplets and uses a knowledge graph to synthesize information. Then the graph information is fed into the Transformer architecture. GSum is a general framework for guided neural summarization, which investigates four types of guidance signals and achieves state-of-the-art performance on various popular datasets.

[5] https://github.com/Huffon/factsumm

[6] https://nlp.stanford.edu/software/openie.html

[7] https://github.com/ThomasScialom/QuestEval

We use the prediction files of the competing models provided in EXPLAINABOARD [8] other than running the models. It is noted that the difference between the performance of EXPLAINABOARD and the results in the original paper is below 1 point in terms of ROUGE.

5.2 Main Results and Analysis

Main results. Table 2 presents the detailed results on the test set of the datasets including traditional metrics and factual consistency metrics. Compared with the results published in EXPLAINABOARD, our reimplement of BART is inferior by about 0.6 points in terms of ROUGE-L. It is noted that the performance of BART on XSUM dataset has been discussed in fairseq repository[9]. The results on base models implicate that our implementation is fair for our study.

We apply GISG on XSUM and CNNDM with BART as the backbone. As seen, GISG achieves higher performance for lexical overlay on both datasets compared to BART. It achieves 0.8 and 0.5 points improvement in terms of ROUGE-L on XSUM and CNNDM datasets, which is a considerable improvement over strong baselines for summarization. It is noted that GSum in Table 2 use the key sentences as the guidance. Although there is the version that GSum uses keywords as the guidance in (Dou et al., 2021), which is more relevant to our work. EXPLAINABOARD does not provide the output files and we report the results using key sentences as the guidance.

For the factual consistency metrics, GISG outperforms BART on all factual consistency metrics which indicates that jointly training keyphrases induction and summary generation benefit the faithful consistency. Compared with a strong factual guidance baseline FASum and GSum, our approach consistently outperforms FASum and GSum.

Compared with extractive summarization baseline MATCHSUM and EXT-ORACLE, the abstractive summarization models have a large margin in terms of factual consistency, even if these models achieve much higher performance on the lexical overlap. It indicates that extractive summarization models can get better factual consistency at the cost of being relevant and fluent.

[8] http://explainaboard.nlpedia.ai/leaderboard/task-summ/index.php

[9] https://github.com/pytorch/fairseq/issues/1971

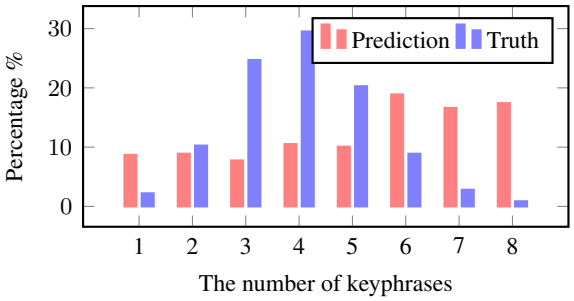

Figure 3: Distribution of the number of keyphrases.

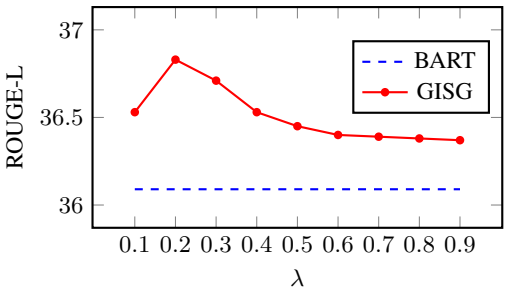

Figure 4: Fine tuning on the hyper-parameter λ.

For the results between XSUM and CNNDM, all factual consistency metrics on XSUM are much lower than CNNDM. This is consistent with the conclusion that summaries in XSUM are much more abstractive. It is more difficult for the model to generate consistent results on XSUM.

Due to the extreme abstractive nature of XSUM dataset, it is ideal to evaluate the models' ability to capture the facts of the document. In the rest of this section, we present in-depth analyses to better understand our model with XSUM as the testbed.

Distribution of the number of keyphrases. We assume the maximum number of the keyphrases is B in Section 4.3 and the model selects fewer keyphrases by selecting one candidate repeatedly. In this section, We investigate the distribution of the number of keyphrases for the test set and the model prediction in Table 3.

As seen, most of the reference summaries have about 4 keyphrases while most of the reference summaries have less than eight keyphrases. Thus B is set to eight according to the ground truth distribution.

Moreover, the number of keyphrases in the model prediction is larger than the ground truth. We argue that it is because the model tends to take advantage of all the latent variables and selects redundancy and similar candidates.

Fine tuning on hyper-parameter λ. In Eqn. (7),

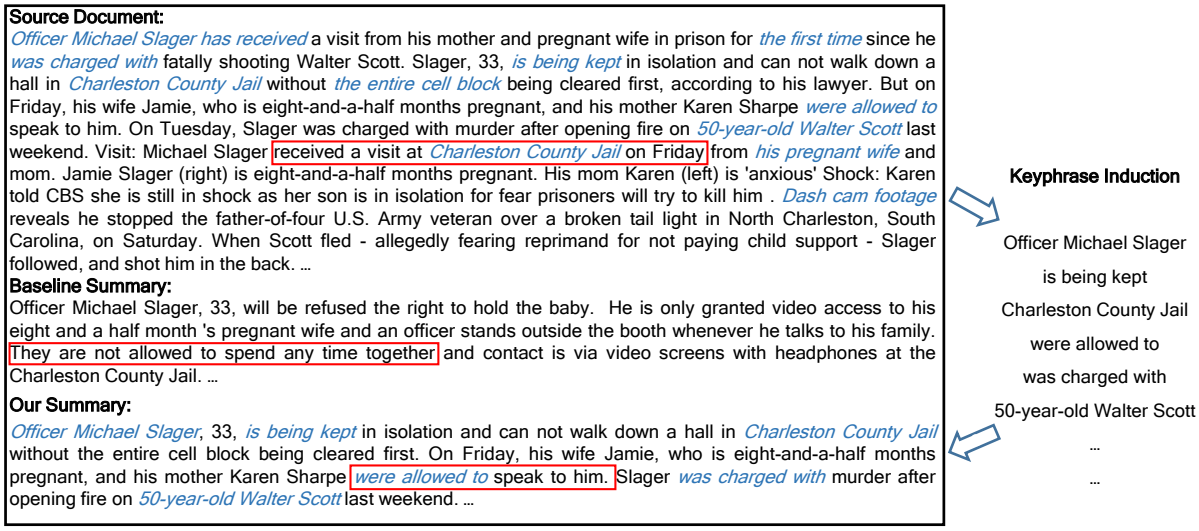

Figure 5: Case study.

we use λ to keep a trade-off between $\mathcal{L}_{\text{CVAE}}$ and $\mathcal{L}_{\text{w}}$. We analyze the effect of λ in Figure 4.

As seen, ROUGE-L is boosted with the increment of λ until 0.2, showing that predicting the words of the summary by the keyphrases contributes to the performance.

Subsequently, a larger value of λ reduces the ROUGE-L and the performance is even lower than without $\mathcal{L}_{\text{w}}$. We argue that it is because $\boldsymbol{f}_{\text{key}}$ is constrained to predict the words of the summary. A larger value of λ would disturb the word prediction item $\boldsymbol{y}_t$, which would hurt the performance. Therefore, we set the hyper-parameter λ to 0.2 to control the effect.

Ablation on the keyphrases prediction network. We first predict the keyphrases and generate the summaries conditioned on the source document and the keyphrases. We investigate the influence of the keyphrases prediction network and replace the module with a random selection of B keyphrases. The results are shown in Table 3.

The results show that both ROUGE, BERTScore and factual consistency metrics have a descend without the keyphrases prediction module, which indicates the effectiveness of the guidance prediction module.

Ablation on the number of keyphrases in testing. To investigate the effectiveness of the keyphrases prediction network, we make ablation of the keyphrases in testing in Figure 6, where we increase the number of the attending keyphrases gradually.

As shown in the figure, the performance increases as more keyphrases are used to generate the summaries. Without attending to any keyphrases of the module prediction, the performance drops about 1 point in terms of ROUGE-L. It indicates that the keyphrase prediction filters the information and helps the decoder to generate a more accurate summarization.

Models	Prediction	Rand
R-L	36.82	36.25
BERTScore	92.11	91.07
Close	2.14	1.96
Open	2.18	2.07
QAGS	14.31	13.94
QuesEval	37.09	37.02

Table 3: Ablation on the keyphrases prediction.

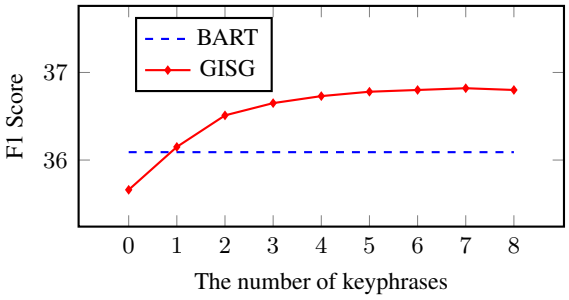

Figure 6: Ablation on the number of keyphrases in testing.

Case Study To further demonstrate the effectiveness of our method, we give a case study in Figure 5. We compare the summary generated based on our approach and baseline which is based

on BART. As shown in Figure 5, the baseline model generated hallucination, “They are not allowed to spend any time together", which is inconsistent with the source document, “They visit at Charleston County Jail". Our model first predicts the keyphrases from the source document and generates the summary conditioned on the source document. As shown in the figure, our result is more faithful, which confirms the effectiveness of our approach.

6 Conclusion

In this paper, we propose to learn guidance induction and summary generation jointly via conditional variational autoencoders. We use phrases as the information granularity of our guidance and we induce the keyphrases of the source document. These summaries are generated conditioned on the source document and the keyphrases, ensuring the important information is consistent with the source document. The experiments show that our approach can generate more faithful summaries than the existing state-of-the-art approaches, according to multiple factual consistency metrics.

Acknowledgments

We are grateful to the anonymous reviewers and meta reviews for their insightful comments and suggestions. This work is funded by the National Natural Science Foundation of China (Grant No. U1908216).

References

Mehdi Allahyari, Seyed Amin Pouriyeh, Mehdi Assefi, Saeid Safaei, Elizabeth D. Trippe, Juan B. Gutierrez, and Krys J. Kochut. 2017. Text summarization techniques: A brief survey. *CoRR*, abs/1707.02268.

Rahul Aralikatte, Shashi Narayan, Joshua Maynez, Sascha Rothe, and Ryan Thomas Mcdonald. 2021. Focus attention: Promoting faithfulness and diversity in summarization. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *Proceedings of The Conference on Computational Natural Language Learning*, pages 10–21.

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

Shuyang Cao and Lu Wang. 2021. CLIFF: Contrastive learning for improving faithfulness and factuality in abstractive summarization. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 6633–6649.

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).

Sihao Chen, Fan Zhang, Kazoo Sone, and Dan Roth. 2021a. Improving faithfulness in abstractive summarization with contrast candidate generation and selection. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, pages 5935–5941.

Wenqing Chen, Jidong Tian, Yitian Li, Hao He, and Yaohui Jin. 2021b. De-confounded variational encoder-decoder for logical table-to-text generation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing*, pages 5532–5542.

Hal Daumé III and Daniel Marcu. 2006. Bayesian query-focused summarization. In *Proceedings of the International Conference on Computational Linguistics and Annual Meeting of the Association for Computational Linguistics*, pages 305–312. Association for Computational Linguistics.

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*, pages 4171–4186.

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 *Advances in Neural Information Processing Systems*.

Yue Dong, Shuohang Wang, Zhe Gan, Yu Cheng, Jackie Chi Kit Cheung, and Jingjing Liu. 2020. Multi-fact correction in abstractive text summarization. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 9320–9331.

Zi-Yi Dou, Pengfei Liu, Hiroaki Hayashi, Zhengbao Jiang, and Graham Neubig. 2021. Gsum: A general framework for guided neural abstractive summarization. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, pages 4830–4842.

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

Saadia Gabriel, Asli Celikyilmaz, Rahul Jha, Yejin Choi, and Jianfeng Gao. 2021. GO FIGURE: A meta evaluation of factuality in summarization. In *Findings of the Association for Computational Linguistics*, pages 478–487.

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*, volume 28, page 1693–1701.

Yichong Huang, Xiachong Feng, Xiaocheng Feng, and Bing Qin. 2021. The factual inconsistency problem in abstractive text summarization: A survey.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*.

Diederik P. Kingma and Max Welling. 2014. Auto-encoding variational bayes. In *International Conference on Learning Representations*.

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

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 Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Haoran Li, Junnan Zhu, Jiajun Zhang, and Chengqing Zong. 2018. Ensure the correctness of the summary: Incorporate entailment knowledge into abstractive sentence summarization. In *Proceedings of the International Conference on Computational Linguistics*, pages 1430–1441.

Chin-Yew Lin and Eduard Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, page 71–78.

Wei Liu, Huanqin Wu, Wenjing Mu, Zhen Li, Tao Chen, and Dan Nie. 2021. Co2sum:contrastive learning for factual-consistent abstractive summarization.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing*, pages 3730–3740.

James Lucas, George Tucker, Roger Grosse, and Mohammad Norouzi. 2019. Don't blame the elbo! a linear vae perspective on posterior collapse. In *Proceedings of the International Conference on Neural Information Processing Systems*.

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

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, Barcelona, Spain. Association for Computational Linguistics.

Ines Montani, Matthew Honnibal, Matthew Honnibal, Sofie Van Landeghem, Adriane Boyd, Henning Peters, Maxim Samsonov, Jim Geovedi, Jim Regan, György Orosz, Paul O'Leary McCann, Søren Lind Kristiansen, Duygu Altinok, Roman, Leander Fiedler, Grégory Howard, Wannaphong Phatthiyaphaibun, Explosion Bot, Sam Bozek, Mark Amery, Yohei Tamura, Björn Böing, Pradeep Kumar Tippa, Leif Uwe Vogelsang, Ramanan Balakrishnan, Vadim Mazaev, GregDubbin, jeannefukumaru, Jens Dahl Møllerhøj, and Avadh Patel. spaCy: v3.0.0rc: Transformer-based pipelines, new training system, project templates, custom models, improved component API, type hints & lots more [online]. 2020.

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 Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing*, pages 6881–6894.

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 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807.

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

Clement Rebuffel, Thomas Scialom, Laure Soulier, Benjamin Piwowarski, Sylvain Lamprier, Jacopo Staiano, Geoffrey Scoutheeten, and Patrick Gallinari. 2021. Data-QuestEval: A referenceless metric for data-to-text semantic evaluation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 8029–8036.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 2014. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the International Conference on Machine Learning*, volume 32, pages 1278–1286.

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. 2015. Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems*, volume 28.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

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

Zijun Wu, Atharva Naik, Zi Xuan Zhang, and Lili Mou. 2021. Weakly supervised explainable phrasal reasoning with neural fuzzy logic. *arXiv:2109.08927*.

Shusheng Xu, Xingxing Zhang, Yi Wu, and Furu Wei. 2022. Sequence level contrastive learning for text summarization. In *Proceedings of the Association for the Advance of Artificial Intelligence*.

Wang Xu, Kehai Chen, and Tiejun Zhao. 2021a. Discriminative reasoning for document-level relation extraction. In *Findings of the Annual Meeting of the Association for Computational Linguistics*, pages 1653–1663.

Wang Xu, Kehai Chen, and Tiejun Zhao. 2021b. Document-level relation extraction with reconstruction. In *Proceedings of the Association for the Advance of Artificial Intelligence*, 16, pages 14167–14175.

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

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020b. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. 2017. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 654–664.

Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2020. Extractive summarization as text matching. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 6197–6208.

Chunting Zhou, Graham Neubig, Jiatao Gu, Mona Diab, Francisco Guzmán, Luke Zettlemoyer, and Marjan Ghazvininejad. 2021. Detecting hallucinated content in conditional neural sequence generation. In *Findings of the Association for Computational Linguistics*, pages 1393–1404.

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 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 718–733.

Qile Zhu, Wei Bi, Xiaojiang Liu, Xiyao Ma, Xiaolin Li, and Dapeng Wu. 2020. A batch normalized inference network keeps the kl vanishing away. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 2636–2649.