BASS: Boosting Abstractive Summarization with Unified Semantic Graph

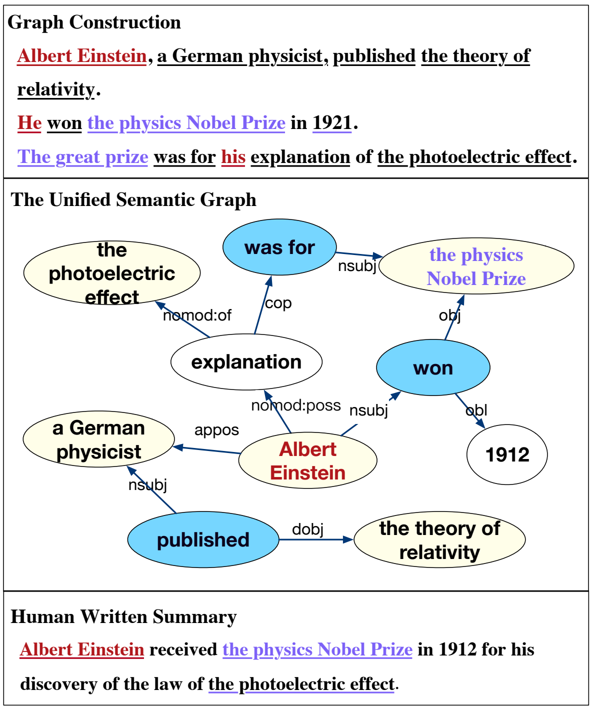
Wenhao Wu[1]*, **Wei Li**[2], **Xinyan Xiao**[2], **Jiachen Liu**[2],**Ziqiang Cao**[3], **Sujian Li**[1]†, **Hua Wu**[2], **Haifeng Wang**[2]
[1]Key Laboratory of Computational Linguistics, MOE, Peking University
[2]Baidu Inc., Beijing, China
[3]Institute of Artificial Intelligence, Soochow University, China
{waynewu,lisujian}@pku.edu.cn
{liwei85,xiaoxinyan,liujiachen,wu_hua,wanghaifeng}@baidu.com
{zqcao}@suda.edu.cn

Abstract

Abstractive summarization for long-document or multi-document remains challenging for the Seq2Seq architecture, as Seq2Seq is not good at analyzing long-distance relations in text. In this paper, we present BASS, a novel framework for Boosting Abstractive Summarization based on a unified Semantic graph, which aggregates co-referent phrases distributing across a long range of context and conveys rich relations between phrases. Further, a graph-based encoder-decoder model is proposed to improve both the document representation and summary generation process by leveraging the graph structure. Specifically, several graph augmentation methods are designed to encode both the explicit and implicit relations in the text while the graph-propagation attention mechanism is developed in the decoder to select salient content into the summary. Empirical results show that the proposed architecture brings substantial improvements for both long-document and multi-document summarization tasks.

1 Introduction

Nowadays, the sequence-to-sequence (Seq2Seq) based summarization models have gained unprecedented popularity (Rush et al., 2015; See et al., 2017; Lewis et al., 2020). However, complex summarization scenarios such as long-document or multi-document summarization (MDS), still bring great challenges to Seq2Seq models (Cohan et al., 2018; Liu et al., 2018). In a long document numerous details and salient content may distribute evenly (Sharma et al., 2019) while multiple documents may contain repeated, redundant or contradictory information (Radev, 2000). These problems make Seq2Seq models struggle with content selection and organization which mainly depend on the long source sequence (Shao et al., 2017). Thus, how to exploit deep semantic structure in the complex text input is a key to further promote summarization performance.



Graph Construction

Albert Einstein, a German physicist, published the theory of relativity.

He won the physics Nobel Prize in 1921.

The great prize was for his explanation of the photoelectric effect.

The Unified Semantic Graph

Human Written Summary

Albert Einstein received the physics Nobel Prize in 1912 for his discovery of the law of the photoelectric effect.

Figure 1: Illustration of a unified semantic graph and its construction procedure for a document containing three sentences. In **Graph Construction**, underlined tokens represent phrases., co-referent phrases are represented in the same color. In **The Unified Semantic Graph**, nodes of different colors indicate different types, according to section 3.1.

Compared with sequence, graph can aggregate relevant disjoint context by uniformly representing them as nodes and their relations as edges. This greatly benefits global structure learning and long-distance relation modeling. Several previous works have attempted to leverage sentence-relation graph to improve long sequence summarization, where nodes are sentences and edges are similarity or dis-

[*] Work is done during an internship at Baidu Inc.

[†] Corresponding author.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pages 6052–6067
August 1–6, 2021. ©2021 Association for Computational Linguistics

course relations between sentences (Li et al., 2020). However, the sentence-relation graph is not flexible for fine-grained (such as entities) information aggregation and relation modeling. Some other works also proposed to construct local knowledge graph by OpenIE to improve Seq2Seq models (Fan et al., 2019; Huang et al., 2020). However, the OpenIE-based graph only contains sparse relations between partially extracted phrases, which cannot reflect the global structure and rich relations of the overall sequence.

For better modeling the long-distance relations and global structure of a long sequence, we propose to apply a phrase-level unified semantic graph to facilitate content selection and organization. Based on fine-grained phrases extracted from dependency parsing, our graph is suitable for information aggregation with the help of coreference resolution that substantially compresses the input and benefits content selection. Furthermore, relations between phrases play an important role in organizing the salient content when generating summaries. For example, in Figure 1 the phrases "Albert Einstein", "the great prize" and "explanation of the of the photoelectric" which distribute in different sentences are easily aggregated through their semantic relations to compose the final summary sentence.

We further propose a graph-based encoder-decoder model based on the unified semantic graph. The graph-encoder effectively encodes long sequences by explicitly modeling the relations between phrases and capturing the global structure based on the semantic graph. Besides, several graph augmentation methods are also applied during graph encoding to tap the potential semantic relations. For the decoding procedure, the graph decoder incorporates the graph structure by graph propagate attention to guide the summary generation process, which can help select salient content and organize them into a coherent summary.

We conduct extensive experiments on both the long-document summarization dataset BIGPATENT and MDS dataset WikiSUM to validate the effectiveness of our model. Experiment results demonstrate that our graph-based model significantly improves the performance of both long-document and multi-document summarization over several strong baselines. Our main contributions are summarized as follows:

- We present the unified semantic graph which aggregates co-referent phrases distributed in context for better modeling the long-distance relations and global structure in long-document summarization and MDS.
- We propose a graph-based encoder-decoder model to improve both the document representation and summary generation process of the Seq2Seq architecture by leveraging the graph structure.
- Automatic and human evaluation on both long-document summarization and MDS outperform several strong baselines and validate the effectiveness of our graph-based model.

2 Related Works

2.1 Abstractive Summarization

Abstractive summarization aims to generate a fluent and concise summary for the given input document (Rush et al., 2015). Most works apply Seq2Seq architecture to implicitly learn the summarization procedure (See et al., 2017; Gehrmann et al., 2018; Paulus et al., 2017; Celikyilmaz et al., 2018). More recently, significant improvements have been achieved by applying pre-trained language models as encoder (Liu and Lapata, 2019b; Rothe et al., 2020) or pre-training the generation process leveraging a large-scale of unlabeled corpus (Dong et al., 2019; Lewis et al., 2020; Qi et al., 2020; Zhang et al., 2020a). In MDS, most of the previous models apply extractive methods (Erkan and Radev, 2004; Cho et al., 2019). Due to the lack of large-scale datasets, some attempts on abstractive methods transfer single document summarization (SDS) models to MDS (Lebanoff et al., 2018; Yang et al., 2019) or unsupervised methods based on auto-encoder (Chu and Liu, 2019; Bražinskas et al., 2020; Amplayo and Lapata, 2020). After the release of several large MDS datasets (Liu et al., 2018; Fabbri et al., 2019), some supervised abstractive models for MDS appear (Liu and Lapata, 2019a; Li et al., 2020). Their works also emphasize the importance of modeling cross-document relations in MDS.

2.2 Structure Enhanced Summarization

Explicit structures play an important role in recent deep learning-based extractive and abstractive summarization methods (Li et al., 2018a,b; Liu et al., 2019a). Different structures benefit summarization models from different aspects. Constituency parsing greatly benefits content selection

Input Length	800	1600	2400	3000
#Nodes	140	291	467	579
#Edges	154	332	568	703

Table 1: Illustration of how the average number of nodes and edges in the graph changes when the input sequence becomes longer on WikiSUM.

and compression for extractive models. Cao et al. (2015) propose to extract salient sentences based on their constituency parsing trees. Xu and Durrett (2019) and Desai et al. (2020) jointly select and compress salient content based on syntax structure and syntax rules. Dependency parsing helps summarization models in semantic understanding. Jin et al. (2020) incorporate semantic dependency graphs of input sentences to help the summarization models generate sentences with better semantic relevance . Besides sentence-level structures, document-level structures also attract a lot of attention. Fernandes et al. (2019) build a simple graph consisting of sentences, tokens and POS for summary generation. By incorporating RST trees, Xu et al. (2020) propose a discourse-aware model to extract sentences. Similarly, structures from semantic analysis also help. Liu et al. (2015) and Liao et al. (2018) propose to guide summarization with Abstract Meaning Representation (AMR) for a better comprehension of the input context. (Li and Zhuge, 2019) propose semantic link networks based MDS but without graph neural networks. Recently, the local knowledge graph by OpenIE attracts great attention. Leveraging OpenIE extracted tuples, Fan et al. (2019) compress and reduce redundancy in multi-document inputs in MDS. Their work mainly focus on the efficiency in processing long sequences. Huang et al. (2020) utilize OpenIE-based graph for boosting the faithfulness of the generated summaries. Compared with their work, our phrase-level semantic graph focus on modeling long-distance relations and semantic structures.

3 Unified Semantic Graph

In this section, we introduce the definition and construction of the unified semantic graph.

3.1 Graph Definition

The unified semantic graph is a heterogeneous graph defined as $G = (V, E)$, where V and E are the set of nodes and edges. Every node in V represents a concept merged from co-referent phrases. For example, in Figure 1 the node "*Albert Einstein*" is merged from phases "*Albert Einstein*" and "*his*" which indicate the same person by coreference resolution. Defined as a heterogeneous graph G, every node $v \in V$ and every edge $e_{ij} \in E$ in our graph belongs to a type of phrase and dependency parsing relation, respectively. Determined by the type of phrases merged from, nodes are categorized into three different types: Noun phrase (N), Verb phrase (V), Other phrase (O). We neglect dependency relations in edges as they mainly indicate sentence syntax. Instead, the meta-paths (Sun et al., 2011) in the unified semantic graph convey various semantic relations. Notice that most O such as adjective phrases, adverb phrases function as modifiers, and the meta-path **O-N** indicates modification relation. The meta-path **N-N** between Noun phrases represents appositive relation or appositional relation. Furthermore, two-hop meta-path represents more complex semantic relations in graph. For example, **N-V-N** like [*Albert Einstein*]-[*won*]-[*the physics Nobel Prize*] indicates SVO (subject–verb–object) relation. It is essential to effectively model the two-hop meta-path for complex semantic relation modeling.

3.2 Graph Construction

To construct the semantic graph, we extract phrases and their relations from sentences by first merging tokens into phrases and then merging co-referent phrases into nodes. We employ CoreNLP (Manning et al., 2014) to obtain coreference chains of the input sequence and the dependency parsing tree of each sentence. Based on the dependency parsing tree, we merge consecutive tokens that form a complete semantic unit into a phrase. Afterwards, we merge the same phrases from different positions and phrases in the same coreference chain to form the nodes in the semantic graph.

The final statistics of the unified semantic graph on WikiSUM are illustrated in table 1, which indicates that the scale of the graph expands moderately with the inputs. This also demonstrates how the unified semantic graph compresses long-text information.

4 Summarization Model

In this section, we introduce our graph-based abstractive summarization model, which mainly consists of a graph encoder and a graph decoder, as shown in Figure 2. In the encoding stage, our

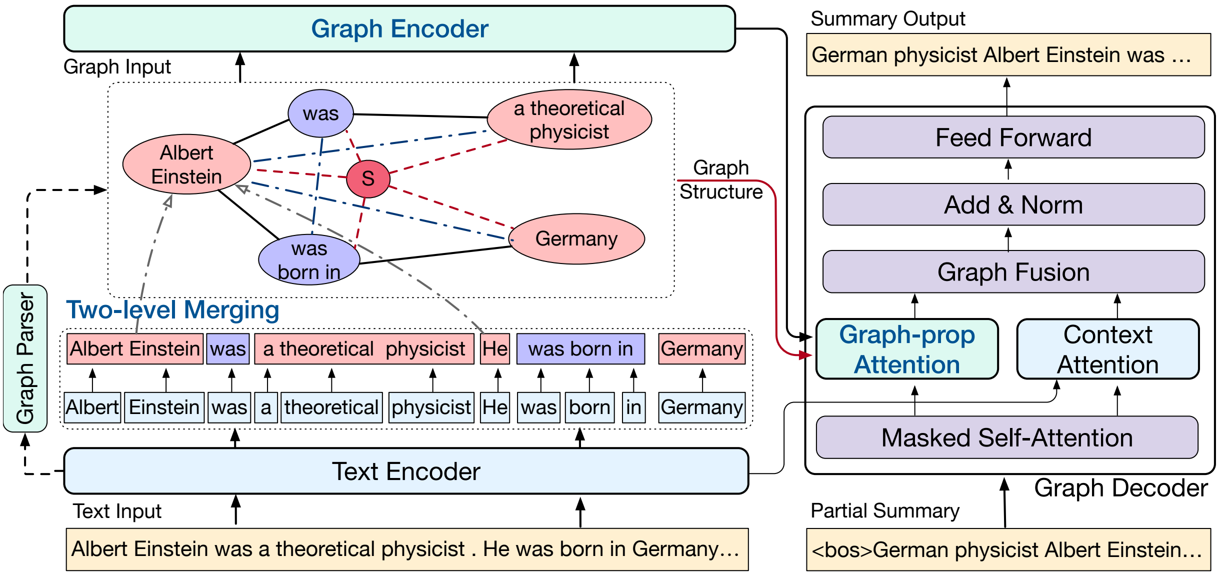


Figure 2: Illustration of our graph-based summarization model. The graph node representation is initialized from merging token representations in two-level. The graph encoder models the augmented graph structure. The decoder attends to both token and node representations and utilizes graph structure by graph-propagation attention.

model takes a document or the concatenation of a set of documents as text input (represented as $x = \{x_k\}$), and encodes it by a text encoder to obtain a sequence of local token representations. The graph encoder further takes the unified semantic graph as graph input (represented as $G = (V, E)$ in section 3.1), and explicitly model the semantic relations in graph to obtain global graph representations. Based on several novel graph-augmentation methods, the graph encoder also effectively taps the implicit semantic relations across the text input. In the decoding stage, the graph decoder leverages the graph structure to guide the summary generation process by a novel graph-propagate attention, which facilitates salient content selection and organization for generating more informative and coherent summaries.

4.1 Text Encoder

To better represent local features in sequence, we apply the pre-trained language model RoBERTa (Liu et al., 2019b) as our text encoder. As the maximum positional embedding length of RoBERTa is 512, we extend the positional embedding length and randomly initialize the extended part. To be specific, in every layer, the representation of every node is only updated by it's neighbors by self attention.

4.2 Graph Encoder

After we obtain token representations by the text encoder, we further model the graph structure to obtain node representations. We initialize node representations in the graph based on token representations and the token-to-node alignment information from graph construction. After initialization, we apply graph encoding layers to model the explicit semantic relations features and additionally apply several graph augmentation methods to learn the implicit structure conveyed by the graph.

Node Initialization Similar to graph construction in section 3.2, we initialize graph representations following the two-level merging, token merging and phrase merging. The token merging compresses and abstracts local token features into higher-level phrase representations. The phrase merging aggregates co-referent phrases in a wide context, which captures long-distance and cross-document relations. To be simple, these two merging steps are implemented by average pooling.

Graph Encoding Layer Following previous works in graph-to-sequence learning (Koncel-Kedziorski et al., 2019; Yao et al., 2020), we apply Transformer layers for graph modeling by applying the graph adjacent matrix as self-attention mask.

Graph Augmentation Following previous works (Bastings et al., 2017; Koncel-Kedziorski et al., 2019), we add reverse edges and self-loop edges in graph as the original directed edges are not enough for learning backward information. For better utilizing the properties of the united semantic graph, we further propose two novel graph augmentation methods.

Supernode As the graph becomes larger, noises introduced by imperfect graph construction also increase, which may cause disconnected sub-graphs. To strengthen the robustness of graph modeling and learn better global representations, we add a special supernode connected with every other node in the graph to increase the connectivity.

Shortcut Edges Indicated by previous works, graph neural networks are weak at modeling multi-hop relations (Abu-El-Haija et al., 2019). However, as mentioned in section 3.1, the meta-paths of length two represent rich semantic structures that require further modeling the two-hop relations between nodes. As illustrated in Figure 2, in a **N-V-N** meta-path [*Albert Einstein*]-[*was*]-[*a theoretical physicist*], the relations [*Albert Einstein*]-[*was*] and [*was*]-[*a theoretical physicist*] are obviously less important than the two-hop relation [*Albert Einstein*]- [*a theoretical physicist*]. Therefore we add shortcut edges between every node and its two-hop relation neighbors, represented as blue edges in Figure 2. We have also attempted other complex methods such as MixHop (Abu-El-Haija et al., 2019), but we find shortcut edges are more efficient and effective. The effectiveness of these graph augmentation methods has also been validated in section 6.2.

4.3 Graph Decoding Layer

Token and node representations benefit summary generation in different aspects. Token representations are better at capturing local features while graph representations provide global and abstract features. For leveraging both representations, we apply a stack of Transformer-based graph decoding layers as the decoder which attends to both representations and fuse them for generating summaries. Let y_t^{l-1} denotes the representation of t-th summary token output by $(l-1)$-th graph decoding layer. For the graph attention, we apply multi-head attention using y_t^{l-1} as query and node representations $V = \{v_j\}$ as keys and values:

$$\alpha_{t,j} = \frac{(y_t^{l-1}W_Q)(v_j W_K)^T}{\sqrt{d_{head}}} \tag{1}$$

where $W_Q, W_K \in \mathbb{R}^{d\times d}$ are parameter weights, $\alpha_{t,j}$ denote the salient score for node j to y_t^{l-1}. We then calculate the global graph vector g_t as weighted sum over values of nodes: $g_t = \sum_j Softmax(\alpha_{t,j})(v_j W_V)$ where $W_V \in \mathbb{R}^{d\times d}$ is a learnable parameter. We also obtain contextualized text vector c_t similar to the procedure above by calculating multi-head attention between y_t^{l-1} and token representations. Afterwards, we use a graph fusion layer which is a feed-forward neural network to fuse the concatenation of the two features: $d_t^l = W_d^T([g_t, c_t])$, where $W_d \in \mathbb{R}^{2d\times d}$ is the linear transformation parameter and d_t^l is the hybrid representation of tokens and graph. After layer-norm and feed-forward layer, the l-th graph decoding layer output y_t^l is used as the input of the next layer and also used for generating the t_{th} token in the final layer.

Graph-propagate Attention When applying multi-head attention to graph, it only attends to node representations linearly, neglecting the graph structure. Inspired by Klicpera et al. (2019), we propose the graph-propagate attention to leverage the graph structure to guide the summary generation process. By further utilizing semantic structure, the decoder is more efficient in selecting and organizing salient content. Without extra parameters, the graph-propagation attention can be conveniently applied to the conventional multi-head attention for structure-aware learning.

Graph-propagate attention consists of two steps: salient score prediction and score propagation. In the first step, we predict the salient score for every node linearly. We apply the output of multi-head attention $\alpha_t \in \mathbb{R}^{|v|\times C}$ in Equation 1 as salient scores, where $|v|$ is the number of nodes in the graph and C is the number of attention heads. C is regarded as C digits or channels of the salient score for every node. We then make the salient score structure-aware through score propagation. Though PageRank can propagate salient scores over the entire graph, it leads to over-smoothed scores, as in every summary decoding step only parts of the content are salient. Therefore, for each node we only propagate its salient score p times in the graph, aggregating at most p-hop relations. Let $\beta_t^0 = \alpha_t$ denotes the initial salient score predicted in previous step, the salient score after p-th propagation is:

$$\beta_t^p = \omega\hat{A}\beta_t^{p-1} + (1-\omega)\beta_t^0 \tag{2}$$

where $\hat{A} = AD^{-1}$ is a degree-normalized adjacent matrix of the graph[1], and $\omega \in (0,1]$ is the teleport

[1] Adjacent matrix A contains self-loop and reverse edges.

probability which defines the salient score has the probability ω to propagate towards the neighbor nodes and $1-\omega$ to restart from initial. The graph-propagation procedure can also be formulated as:

$$\beta_t^p = (\omega^p \hat{A}^p + (1-\omega)(\sum_{i=0}^{p-1} \omega^i \hat{A}^i))\alpha_t \quad (3)$$

After p steps of salient score propagation, the graph vector is then calculated by weighted sum of node values:

$$g_t^{'} = \sum_j Softmax(\beta_{t,j}^p)(v_j W^V) \quad (4)$$

where for the convenience of expression, the concatenation of multi-head is omitted. The output of fusing $g_t^{'}$ and c_t is then applied to generate the t_{th} summary token as mentioned before.

5 Experiment Setup

In this section, we describe the datasets of our experiments and various implementation details.

5.1 Summarization Datasets

We evaluate our model on a SDS dataset and an MDS dataset, namely BIGPATENT (Sharma et al., 2019) and WikiSUM (Liu et al., 2018).

BIGPATENT is a large-scale patent document summarization dataset with an average input of 3572.8 words and a reference with average length of 116.5 words. BIGPATENT is a highly abstractive summarization dataset with salient content evenly distributed in the input. We follow the standard splits of Sharma et al. (2019) for training, validation, and testing (1,207,222/67,068/67,072).

WikiSUM is a large-scale MDS dataset. Following Liu and Lapata (2019a), we treat the generation of lead Wikipedia sections as an MDS task. To be specific, we directly utilize the preprocessed results from Liu and Lapata (2019a), which split source documents into multiple paragraphs and rank the paragraphs based on their titles to select top-40 paragraphs as source input. The average length of each paragraph and the target summary are 70.1 tokens and 139.4 tokens, respectively. We concatenate all the paragraphs as the input sequence. We use the standard splits of Liu and Lapata (2019a) for training, validation, and testing (1,579,360/38,144/38,205).

Model	R-1	R-2	R-L	BS
Lead	38.22	16.85	26.89	-
LexRank	36.12	11.67	22.52	-
TransS2S	40.56	25.35	34.73	25.43
T-DMCA	40.77	25.60	34.90	-
HT	41.53	26.52	35.76	25.62
BERTS2S	41.49	25.73	35.59	-
RoBERTaS2S	42.05	27.00	36.56	29.13
GraphSum	42.99	27.83	37.36	29.69
BASS(2400)	43.65	**28.55**	37.85	**31.91**
BASS(3000)	**44.33**	28.38	**37.87**	31.71

Table 2: Evaluation results on the test set of WikiSUM. Rouge-1, Rouge-2, Rouge-L and BERTScore are abbreviated as R-1,R-2,R-L and BS, respectively.

5.2 Implementation Details

We train all the abstractive models by max likelihood estimation with label smoothing (label smoothing factor 0.1). As we fine-tune the pre-trained language model RoBERTa as text encoder, we apply two different Adam optimizers (Kingma and Ba, 2015) with $\beta_1 = 0.9$ and $\beta_2 = 0.998$ to train the pre-trained part and other parts of the model (Liu and Lapata, 2019b). The learning rate and warmup steps are 2e-3 and 20,000 for the pre-trained part and 0.1 and 10,000 for other parts. As noticed from experiments, when the learning rate is high, graph-based models suffer from unstable training caused by the gradient explosion in the text encoder. Gradient clipping with a very small maximum gradient norm (0.2 in our work) solves this problem. All the models are trained for 300,000 steps on BIGPATENT and WikiSUM with 8 GPUs (NVIDIA Tesla V100). We apply dropout (with the probability of 0.1) before all linear layers. In our model, the number of graph-encoder layers and graph-decoder layers are set as 2 and 6, respectively. The hidden size of both graph encoding and graph decoding layers is 768 in alignment with RoBERTa, and the feed-forward size is 2048 for parameter efficiency. For graph-propagation attention, the parameter ω is 0.9, and the propagation steps p is 2. During decoding, we apply beam search with beam size 5 and length penalty with factor 0.9. Trigram blocking is used to reduce repetitions.

6 Results

6.1 Automatic Evaluation

We evaluate the quality of generated summaries using ROUGE F_1(Lin, 2004) and BERTScore (Zhang

Model	R-1	R-2	R-L	BS
Lead	31.27	8.75	26.18	-
ORACLE	43.56	16.91	36.52	-
LexRank	35.99	11.14	29.60	-
Seq2Seq	28.74	7.87	24.66	-
Pointer	30.59	10.01	25.65	-
Pointer+cov	33.14	11.63	28.55	-
FastAbs	37.12	11.87	32.45	-
TLM	36.41	11.38	30.88	-
TransS2S	34.93	9.86	29.92	9.42
RoBERTaS2S	43.62	18.62	37.86	18.18
BART	**45.83**	19.53	-	-
Pegasus-base	43.55	**20.43**	-	-
BASS	45.04	20.32	**39.21**	**20.13**

Table 3: Evaluation results on the test set of BIGPATENT where the length input of BASS is 1024.

Model	R-1	R-2	R-L	BS
Full model	**42.29**	**27.19**	**36.46**	**30.62**
w/o structure	41.86	27.06	36.43	29.84
+w/o merging	41.56	26.61	35.93	29.15

Table 4: Graph Structure analysis on WikiSUM test set where the input length is 800. **w/o structure** and **+w/o merging** refer to remove relations between phrases and further remove phrase merging in graph construction, respectively.

Model	R-1	R-2	R-L	BS
Full model	**43.40**	**28.50**	**37.71**	**31.64**
w/o shortcut	42.50	27.97	37.23	31.10
w/o supernode	42.93	28.08	37.42	31.15
w/o graph-prop	42.84	28.14	37.42	31.33
w/o graph	42.05	27.00	36.56	29.13

Table 5: Ablation study on WikiSUM test set where the input length is 1600. **graph-prop** is the abbreviation of graph-propagation.

et al., 2020b). For ROUGE, we report unigram and bigram overlap between system summaries and reference summaries (ROUGE-1, ROUGE-2). We report sentence-level ROUGE-L for the BIGPATENT dataset and summary-level ROUGE-L for the WikiSUM for a fair comparison with previous works. We also report BERTScore [2] F_1, a better metric at evaluating semantic similarity between system summaries and reference summaries.

Results on MDS Table 2 summarizes the evaluation results on the WikiSUM dataset. We compare our model with several strong abstractive and extractive baselines. As listed in the top block, Lead and LexRank (Erkan and Radev, 2004) are two classic extractive methods. The second block shows the results of several different abstractive methods. TransS2S is the Transformer-based encoder-decoder model. By replacing the Transformer encoder in TransS2S with BERT (Devlin et al., 2019) or RoBERTa and training with two optimizers (Liu and Lapata, 2019b), we obtain two strong baselines BERTS2S and RoBERTaS2S. T-DMCA is the best model presented by Liu et al. (2018) for summarizing long sequence. HT is the best model presented by Liu and Lapata (2019a) with the hierarchical Transformer encoder and a flat Transformer decoder. GraphSum, presented by Li et al. (2020), leverages paragraph-level explicit graph by the graph encoder and decoder, which gives the current best performance on WikiSUM. We report the best results of GraphSum with RoBERTa and the input length is about 2400 tokens. The last block reports the results of our model BASS with the input lengths of 2400 and 3000. Compared with all the baselines, our model BASS achieves great improvements on all the four metrics. The results demonstrates the effectiveness of our phrase-level semantic graph comparing with other RoBERTa based models, RoBERTaS2S (without graph) and GraphSum (sentence-relation graph). Furthermore, the phrase-level semantic graph improves the semantic relevance of the generated summaries and references, as the BERTScore improvements of BASS is obvious.

Results on SDS Table 3 shows our experiment results along with other SDS baselines. Similar to WikiSUM, we also report LexRank, TransS2S, and RoBERTaS2S. Besides, we report the performance of several other baselines. ORACLE is the upper-bound of current extrative models. Seq2seq is based on LSTM encoder-decoder with attention mechanism (Bahdanau et al., 2015). Pointer and Pointer+cov are pointer-generation (See et al., 2017) with and without coverage mechanism, respectively. FastAbs (Chen and Bansal, 2018) is an abstractive method by jointly training sentence extraction and compression. TLM (Pilault et al., 2020) is a recent long-document summarization method based on language model. We also report the performances of recent pretrianing-based SOTA

[2] We apply roberta-large_L17_no-idf_version as the metric model and rescale_with_baseline setting according to suggestions on https://github.com/Tiiiger/bert_score.

text generation models BART (large) and Peaguasus (base) on BIGPATENT, which both contain a parameter size of 406M . The last block shows the results of our model, which contains a parameter size of 201M . The results show that BASS consistently outperforms RoBERTaS2S, and comparable with current large SOTA models with only half of the parameter size. This further demonstrates the effectiveness of our graph-augmented model on long-document summarization.

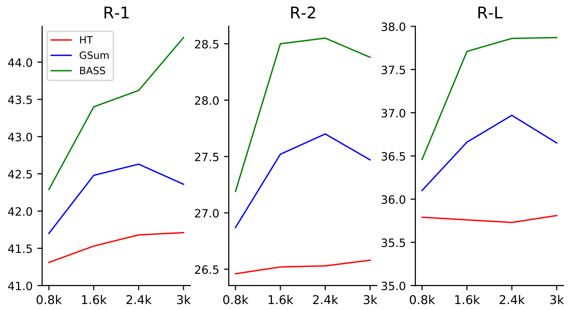


Figure 3: Comparison of HT, GraphSum (GSum in figure), BASS under various length of input tokens.

6.2 Model Analysis

For a thorough understanding of BASS, we conduct several experiments on the WikiSUM test set, including the effects of the graph structure and input length. We also validate the effectiveness of the graph-augmentation methods in graph encoder and the graph-propagation attention in graph decoder by ablation studies.

Graph Structure Analysis To analyze how the unified semantic graph benefits summarization learning, we conduct ablation studies on the graph structures. Illustrated in Table 4, after removing explicit relations between phrases by fully connecting all the nodes, the R-1 metric drops obviously which indicates the relations between phrases improve the informativeness of generated summaries. After further removing phrase merging, we observe a performance decrease in all the metrics, which indicates the long-distance relations benefit both the informativeness and fluency of summary.

Ablation Study The experimental results of removing supernode and shortcut edges from the unified semantic graph prove the effectiveness of graph augmentation methods in the graph encoder. Experimental results without the gaph-propagation attention confirms that the structure of the unified semantic graph is also beneficial for decoding. Overall, the performance of the model drops the most when removing shortcut edges which indicates the rich potential information is beneficial for summarization. Finally, after removing all the graph-relevant components, performance dramatically drops on all the metrics.

Length Comparison According to Liu et al. (2018), input length affects the summarization performance seriously for Seq2Seq models as most of them are not efficient at handling longer sequences. The basic TransS2S achieves its best performance at the input length of 800, while longer input hurts performance. Several previous models achieve better performance when utilizing longer sequences. As illustrated in Figure 3, the performance of HT remains stable when the input length is longer than 800. Leveraging the power of sentence-level graph, GraphSum achieves the best performance at 2,400 but its performance begins to decrease when the input length reaches 3000. Unlike previous methods, ROUGE-1 of BASS significantly increased in 3000 indicates that the unified semantic graph benefits salient information selection even though the input length is extreme.

Abastractiveness Analysis We also study the abstractiveness of BASS and other summarization systems on WikiSUM. We calculate the average novel n-grams to the source input, which reflects the abstractiveness of a summarization system (See et al., 2017). Illustrated in Figure 4, BASS generates more abstract summaries comparing to recent models, GraphSum, HT, and weaker than RoBERTaS2S. Summarized from observation, we draw to a conclusion that RoBERTaS2S usually generates context irrelevant contents due to the strong pretrained RoBERTa encoder but a randomly initialized decoder that relays on the long-text input poorly. Graph-based decoders of BASS and GraphSum alleviate this phenomenon.

6.3 Human Evaluation

In addition to the above automatic evaluations, we also conduct human evaluations to assess the performance of systems. Because the patent dataset BIGPATENT contains lots of terminologies and requires professional background knowledge for annotators, we select WikiSUM as the dataset for evaluations. As Wikipedia entries can be summarized in many different aspects, annotators will naturally favor systems with longer outputs. Thus we first filter instances that the summaries of different systems are significantly different in lengths

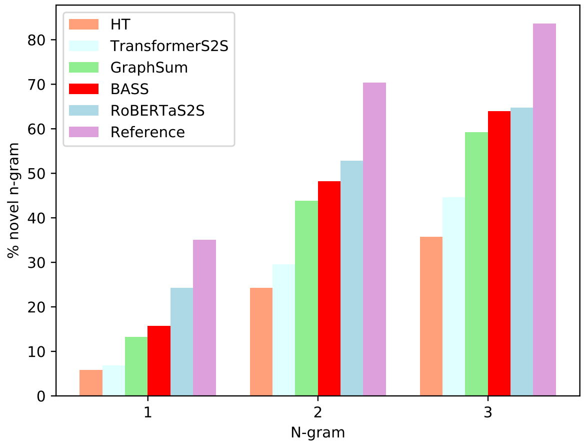


Figure 4: Illustration of novel n-grams in generated summaries form different systems.

and then randomly select 100 instances. We invite 2 annotators to assess the summaries of different models independently.

Annotators evaluate the overall quality of summaries by ranking them taking into account the following criterias: (1) *Informativeness*: whether the summary conveys important and faithful facts of the input? (2) *Fluency*: whether the summary is fluent, grammatical, and coherent? (3) *Succinctness*: whether the summary is concise and dose not describe too many details? Summaries with the same quality get the same order. All systems get score 2,1,-1,2 for ranking 1,2,3,4 respectively. The rating of each system is averaged by the scores of all test instances.

The results of our system and the other three strong baselines are shown in Table 6. The percentage of rankings and the overall scores are both reported. Summarized from the results, our model BASS is able to generate higher quality summaries. Some examples are also shown in the appendix. Specifically, BASS generates fluent and concise summaries containing more salient content compared with other systems. The human evaluation results further validate the effectiveness of our semantic graph-based model.

7 Conclusion and Future Work

In this paper, we propose to leverage the unified semantic graph to improve the performance of neural abstractive models for long-document summarization and MDS. We further present a graph-based encoder-decoder model to improve both the document representation and summary generation process by leveraging the graph structure. Experiments on both long-document summarization and MDS show that our model outperforms several strong baselines, which demonstrates the effectiveness of our graph-based model and the superiority of the unified semantic graph for long-input abstractive summarization. Though remarkable achievements have been made by neural network-based summarization systems, they still do not actually understand languages and semantics. Incorporating language structures in deep neural networks as prior knowledge is a straightforward and effective way to help summarization systems, as proved by this work and previous works.

Model	1	2	3	4	Rating
TransS2S	0.32	0.14	0.09	0.45	-0.21^*
R.B.	0.39	0.22	0.26	0.13	0.48^*
G.S.	0.31	0.38	0.20	0.11	0.58^*
BASS	0.64	0.16	0.14	0.06	**1.18**

Table 6: Ranking results of system summaries by human evaluation. 1 is the best and 4 is the worst. The larger rating denotes better summary quality. R.B. and G.S. are the abbreviations of RoBERTaS2S and GraphSum. * indicates the overall ratings of the corresponding model are significantly (by Welchs t-test with p <0.01) outperformed by BASS.

Acknowledgments

This work was partially supported by National Key R&D Program of China (No. 2020YFB1406701) and National Natural Science Foundation of China (No. 61876009).

References

Sami Abu-El-Haija, Bryan Perozzi, Amol Kapoor, Nazanin Alipourfard, Kristina Lerman, Hrayr Harutyunyan, Greg Ver Steeg, and Aram Galstyan. 2019. MixHop: Higher-order graph convolutional architectures via sparsified neighborhood mixing. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 21–29, Long Beach, California, USA. PMLR.

Reinald Kim Amplayo and Mirella Lapata. 2020. Unsupervised opinion summarization with noising and denoising. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1934–1945, Online. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Jasmijn Bastings, Ivan Titov, Wilker Aziz, Diego Marcheggiani, and Khalil Sima'an. 2017. Graph convolutional encoders for syntax-aware neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1957–1967, Copenhagen, Denmark. Association for Computational Linguistics.

Arthur Bražinskas, Mirella Lapata, and Ivan Titov. 2020. Unsupervised opinion summarization as copycat-review generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5151–5169, Online. Association for Computational Linguistics.

Ziqiang Cao, Furu Wei, Li Dong, Sujian Li, and Ming Zhou. 2015. Ranking with recursive neural networks and its application to multi-document summarization. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, page 2153–2159. AAAI Press.

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

Yen-Chun Chen and Mohit Bansal. 2018. Fast abstractive summarization with reinforce-selected sentence rewriting. *CoRR*, abs/1805.11080.

Sangwoo Cho, Chen Li, Dong Yu, Hassan Foroosh, and Fei Liu. 2019. Multi-document summarization with determinantal point processes and contextualized representations. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 98–103, Hong Kong, China. Association for Computational Linguistics.

Eric Chu and Peter J. Liu. 2019. Meansum: A neural model for unsupervised multi-document abstractive summarization. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 1223–1232. PMLR.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 615–621, New Orleans, Louisiana. Association for Computational Linguistics.

Shrey Desai, Jiacheng Xu, and Greg Durrett. 2020. Compressive summarization with plausibility and salience modeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6259–6274, Online. 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: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 13042–13054.

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

Alexander Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. 2019. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1074–1084, 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.

Patrick Fernandes, Miltiadis Allamanis, and Marc Brockschmidt. 2019. Structured neural summarization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

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

Luyang Huang, Lingfei Wu, and Lu Wang. 2020. Knowledge graph-augmented abstractive summarization with semantic-driven cloze reward. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5094–5107, Online. Association for Computational Linguistics.

Hanqi Jin, Tianming Wang, and Xiaojun Wan. 2020. Semsum: Semantic dependency guided neural abstractive summarization. In *AAAI*, pages 8026–8033.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. 2019. Predict then propagate: Graph neural networks meet personalized pagerank. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

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

Logan Lebanoff, Kaiqiang Song, and Fei Liu. 2018. Adapting the neural encoder-decoder framework from single to multi-document summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4131–4141, Brussels, Belgium. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pretraining for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

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.

Wei Li, Xinyan Xiao, Yajuan Lyu, and Yuanzhuo Wang. 2018a. Improving neural abstractive document summarization with explicit information selection modeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 1787–1796. Association for Computational Linguistics.

Wei Li, Xinyan Xiao, Yajuan Lyu, and Yuanzhuo Wang. 2018b. Improving neural abstractive document summarization with structural regularization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 4078–4087. Association for Computational Linguistics.

Wei Li and Hai Zhuge. 2019. Abstractive multi-document summarization based on semantic link network. *IEEE Transactions on Knowledge and Data Engineering*.

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.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

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.

Peter J. Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. 2018. Generating wikipedia by summarizing long sequences. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Yang Liu and Mirella Lapata. 2019a. Hierarchical transformers for multi-document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5070–5081, Florence, Italy. Association for Computational Linguistics.

Yang Liu and Mirella Lapata. 2019b. Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3730–3740, Hong Kong, China. Association for Computational Linguistics.

Yang Liu, Ivan Titov, and Mirella Lapata. 2019a. Single document summarization as tree induction. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,* *NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 1745–1755. Association for Computational Linguistics.

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

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.

Romain Paulus, Caiming Xiong, and Richard Socher. 2017. A deep reinforced model for abstractive summarization. *CoRR*, abs/1705.04304.

Jonathan Pilault, Raymond Li, Sandeep Subramanian, and Chris Pal. 2020. On extractive and abstractive neural document summarization with transformer language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9308–9319, Online. Association for Computational Linguistics.

Weizhen Qi, Yu Yan, Yeyun Gong, Dayiheng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang, and Ming Zhou. 2020. ProphetNet: Predicting future n-gram for sequence-to-SequencePre-training. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2401–2410, Online. Association for Computational Linguistics.

Dragomir Radev. 2000. A common theory of information fusion from multiple text sources step one: Cross-document structure. In *1st SIGdial Workshop on Discourse and Dialogue*, pages 74–83, Hong Kong, China. Association for Computational Linguistics.

Sascha Rothe, Shashi Narayan, and Aliaksei Severyn. 2020. Leveraging pre-trained checkpoints for sequence generation tasks. *Transactions of the Association for Computational Linguistics*, 8:264–280.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, Lisbon, Portugal. Association for Computational Linguistics.

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

Louis Shao, Stephan Gouws, Denny Britz, Anna Goldie, Brian Strope, and Ray Kurzweil. 2017. Generating long and diverse responses with neural conversation models. *CoRR*, abs/1701.03185.

Eva Sharma, Chen Li, and Lu Wang. 2019. BIGPATENT: A large-scale dataset for abstractive and coherent summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2204–2213, Florence, Italy. Association for Computational Linguistics.

Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S Yu, and Tianyi Wu. 2011. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *Proceedings of the VLDB Endowment*, 4(11):992–1003.

Jiacheng Xu and Greg Durrett. 2019. Neural extractive text summarization with syntactic compression. 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 3292–3303, Hong Kong, China. Association for Computational Linguistics.

Jiacheng Xu, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. Discourse-aware neural extractive text summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5021–5031, Online. Association for Computational Linguistics.

Wenmian Yang, Weijia Jia, Wenyuan Gao, Xiaojie Zhou, and Yutao Luo. 2019. Interactive variance attention based online spoiler detection for time-sync comments. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19, page 1241–1250, New York, NY, USA. Association for Computing Machinery.

Shaowei Yao, Tianming Wang, and Xiaojun Wan. 2020. Heterogeneous graph transformer for graph-to-sequence learning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7145–7154. Association for Computational Linguistics.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020a. PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 11328–11339. PMLR.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020b. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

A Graph Construction

Given a document set with n documents $D = \{d_1, ...d_n\}$ and each document $d_i \in D$ contains k_i sentences. Algorithm 1 gives the details of constructing the unified semantic graph based on dependency parsing.

We apply CoreNLP for both coreference resolution and dependency parsing. We first extract coreference chains from every document and merge coreference chains with overlap phrases. We memorize all the coreference chains in set C, where each chain $c = \{p_1, ..., p_{k_c}\} \in C$ contains a set of co-referent phrases. We then parse every sentence in every document into a dependency parsing tree T_s. Afterwords we refines the tree by following operations:

Algorithm 1: Construct Unified Semantic Graphs

```
Input: Documents set $\mathcal{D} = \{d_1, ..., d_n\}$, document
       $d_i \in \mathcal{D}$, $d_i = \{s_1, ..., s_{k_i}\}$
Output: The unified semantic graph $\mathcal{G}$
                                        ▷ Coreference Resolution
$\mathcal{C} \leftarrow \emptyset$
foreach $d \in \mathcal{D}$ do
    $c_d \leftarrow$ COREFERNCE_RESOLUTION($d$)
    $\mathcal{C} \leftarrow$ COREFERNCE_MERGE($\mathcal{C}, c_d$)
end
                                        ▷ Dependency Parsing
$\mathcal{T} \leftarrow \emptyset$
foreach $d \in \mathcal{D}$ do
    foreach $s \in d$ do
        $T_s \leftarrow$ DEPENDENCY_PARSE($s$)
        $T_s \leftarrow$ IDENTIFY_NODE_TYPES($T_s$)
        $T_s \leftarrow$ REMOVE_PUNCTUATION($T_s$)
        $T_s \leftarrow$ MERGE_COREF_PHRASE($T_s, \mathcal{C}$)
        $T_s \leftarrow$ MERGE_NODES($T_s$)
        $\mathcal{T} \leftarrow \mathcal{T} \bigcup \{T_s\}$
    end
end
                                        ▷ Initialize Graph
$\mathcal{G} = (\mathcal{V}, \mathcal{E}), \mathcal{V} \leftarrow \emptyset, \mathcal{E} \leftarrow \emptyset$
foreach tree $T = (V_T, E_T) \in \mathcal{T}$ do
    $\mathcal{V} \leftarrow \mathcal{V} \bigcup \{V_T\}$
    $\mathcal{E} \leftarrow \mathcal{E} \bigcup \{E_T\}$
end
                                        ▷ Merge Co-referent Nodes
foreach corefernce chain $c \in \mathcal{C}$ do
    $(\mathcal{V}, \mathcal{E}) \leftarrow$ MERGE_PHRASE($c, \mathcal{V}, \mathcal{E}$)
end
$\mathcal{G} \leftarrow (\mathcal{V}, \mathcal{E})$
return $\mathcal{G}$
```

- IDENTIFY_NODE_TYPES: after dependency parsing, each node in the tree is attached with a POS tag. We associate every node with its POS tag for future merging operations.
- PRUNE_PUNCTUATION: we remove all the punctuation nodes and their edges.
- MERGE_COREFE_PHRASE: since a coreference chain contains a set of phrases but a dependency parsing tree is based on tokens, we first obtain phrases in coreference chains for the future convenience in merging coreferent phrases. For every phrase p_i in a co-reference chain c, we merge the corresponding tokens of p_i to form the target phrase p_i in the tree. The merging operation is carried out by removing edges between the nodes and represent the tokens as a unified node.
- NODE_MERGE: after obtaining phrases in coreference chains, we merge other token nodes into concise phrases. This procedure is carried out by traveling every dependency graph in depth-first, and merge the tokens into a phrase if they satisfy the merging conditions. Overall, we merge consecutive tokens that form a complete semantic unit into a phrases.

After we extract all the phrases, we merge all the same phrases and phrases in the same coreference chain by MERGE_PHRASE and return the final semantic graph.

B Case Study

We select several cases from human evaluation and demonstrate them to show the overall quality of systems. In each table, there are four blocks present the input article (**Article**), the reference summary (**Reference Summary**), the output summary of a strong baseline GraphSum (**Baseline**) and the output summary of our model BASS (**BASS**), separately. The original input article is the concatenation of several document paragraphs by the "||" symbol containing 1600 tokens in maximum. We only show the salient part of the input article due to the paragraph constraints. Spans in highlight indicate the salient contents. Spans in red indicate the unfaithful content, irrelevant content or repeats a system generated. The case in Table 7 describes an American ice hockey player "Colleen Coyne". The important fact, "won a gold medal at the 1998 winter Olympics", is well captured by BASS, however, the baseline model only mentions she "was a member" neglecting the substantial achievement. The case in Table 8 introduces the play "Colleen Coyne" which based on the four novels of “Leonardo Padura” is difficult to summarize, as the relation between “Colleen Coyne”, “Leonardo Padura” and the name list of the four novels cross different documents and a long-span. The baseline model confuses with the name of stars and fails to list the names of four books. The **Reference Summary** in Table 9 is not informative enough to give a precise description of what is “Cetacean Intelligence”. Though BASS does not introduce the definition of “Cetacean”, it clearly describes the categories of “Cetacean Intelligence” which is more essential to the topic. In Table 10, BASS and Baseline generate summaries with similar content, but BASS provides more details such as, “right-handed”, distributed in different documents. In the case describing Broadcast, in Table 11, while the Baseline generates irrelevant titles of editors, BASS describes essential characters of the magazine. Though all the models apply trigram-block to avoid repeats, Table 12 shows that sometimes the Baseline still generates repeated n-grams while this seldomly happens on BASS.

Article: colleen coyne is a graduate of the university of new hampshire. an ice hockey player , she represented the united states , as a defenseman, on 6 nat ... || ... colleen coyne was one of the trailblazing women who won gold at the 1998 olympic winter games . ...||...history and heroes:george nagobads, mike ilitch's pizza & colleen coyne of the 1998 u.s. women's olympic team ...
Reference Summary: colleen m. coyne (born september 19, 1971) is an american ice hockey player. she won a gold medal at the 1998 winter olympics.
Baseline: colleen coyne (born november 3, 1974) is an american former ice hockey player. she was a member of the united states women 's national ice hockey team at the 1998 winter olympics.
BASS: colleen coyne is an american ice hockey player. she won a gold meda at the 1998 winter olympics. 1998 winter olympics.

Table 7: Colleen Coyne

Article: "havana quartet" , based on the popular book series by cuban novelist leonardo padura , follows a hard-drinking, romantic cuban police detective mario conde... || "havana quartet" follows hard-drinking, romantic cuban police detective mario conde... || known cuban playwright eduardo machado -lrb- starz 's magic city -rrb-, who lives in the u.s., is the writer on the project, based on the popular four-book series of detective novels havana blue, havana gold, havana red and havana black by another cuban writer that is well known internationally, novelist leonardo padura .
Reference Summary: havana quartet is an upcoming american television drama series starring antonio banderas. the series is based on four detective novels by cuban author leonardo padura , havana blue , havana gold , havana red and havana black .
Baseline: havana quartet is an upcoming american television drama series based on the novel series of the same name by cuban novelist leonardo padura the series stars antonio banderas, eduardo machado and eduardo machad.
BASS: havana quartet -lrb- spanish : la gazeta de havana -rrb- is an upcoming american television series baed on leonardo padura 's novels havana blue , havana gold , havana red and havana black.

Table 8: Havana Quartet

Article:in the past three decades, new research has revealed that dolphin brains are not only large but are extremely complex in their organization... dolphin brains are larger than expected for their body size and contain numerous features associated with sophisticated perceptual and cognitive abilities, echolocation, communication, cooperation, and dynamic social networks , cultural transmission of learned behaviors , and self-awareness , to name a few.
Reference Summary: cetacean intelligence refers to the cognitive capabilities of the cetacea order of mammals. this order includes whales, porpoises, and dolphins.
Baseline: cetacean intelligence (cetacean intelligence) refers to the ability of cetaceans to communicate with other animals. cetacean intelligence is an object of fascination among scientists.
BASS:cetacean intelligence refers to the ability of cetaceans to communicate with other animals. cetacans have a wide range of social and cognitive abilities, including echolocation, communication, cooperation and dynamic social networks, cultural transmission of learned behaviors,and self-awareness .

Table 9: Cetacean Intelligence

<table>
<tr><td><b>Article:</b>|| james tolbert hearn was born on april 11 , 1921 , in atlanta , georgia , the sixth and final child born to raymond and pearl (adams) hearn || st. louis cardinals , new york giants , and philadelphia phillies from 1947 through 1959 .|| jim hearn a || right-handed pitcher who helped propel the new york giants to victory ...</td></tr>
<tr><td><b>Reference Summary:</b> james tolbert hearn (april 11 , 1921 – june 10 , 1998) was an american right-handed pitcher in major league baseball for 13 seasons (1947 – 59) . a right-hander , he stood 6 feet 3 inches (1.91 m) tall and weighed 205 pounds (93 kg) .</td></tr>
<tr><td><b>Baseline:</b> james tolbert hearn (april 11 , 1921 – june 10 , 1998) was an american professional baseball pitcher . he played 13 seasons in major league baseball (mlb) for the st. louis cardinals , new york giants , and philadelphia phillies .</td></tr>
<tr><td><b>BASS:</b>james tolbert hearn (april 11, 1921 – june 10, 1998) was an american right-handed pitcher in major league baseball who played for the st. louis cardinals, new york giants , and philadelphia phillies from 1947 through 1959 .</td></tr>
</table>

Table 10: James Tolbert Hearn

<table>
<tr><td><b>Article:</b> broadcast (magazine) content s.v.p. chief content officer derek t. dingle s.v.p. / executive editor-at-large alfred a. edmond jr. . v.p. director of ...|| broadcast , a weekly magazine for the uk television and radio industry , ...|| broadcast is the ideal supplement for those working in the uk broadcasting industry , offering up-to-date industry news from the most experienced and authoritative voices in broadcasting ||as a broadcast subscriber , you will benefit from the uk tv industry’s most knowledgeable source of industry news and analysis on the successful people , programmes and companies .</td></tr>
<tr><td><b>Reference Summary:</b> broadcast is a weekly magazine for the united kingdom television and radio industry . it covers a wide range of news and issues affecting the professional broadcast market in the uk . broadcast has regular weekly sections covering news ...</td></tr>
<tr><td><b>Baseline:</b> broadcast is a british monthly magazine covering the uk television industry . it is published by content .v.p. . the magazine was established in 1931 and is edited by alfred a. edmond , jr. . v.p.</td></tr>
<tr><td><b>BASS:</b>broadcast is a monthly magazine covering the uk television industry . the magazine was first published in 1931 . it is the uk ’s most authoritative voices of industry news and analysis on the successful people , programmes and companies .</td></tr>
</table>

Table 11: Broadcast

<table>
<tr><td><b>Article:</b>dams building (sault ste. marie , michigan) npgallery allows you to search the national register information system a database of over 90,000 historic buildings ,... (added 2010 - - # 10000218) also known as central savings bank building 418 ashmun st. , sault ste. marie || for those of you who are interested in working with data in a gis environment...</td></tr>
<tr><td><b>Reference Summary:</b> the adams building , also known as the central savings bank building , was built as a commercial and office building located at 418 ashmun street in sault ste. marie , michigan . ... .it was listed on the national register of historic places in 2010 .</td></tr>
<tr><td><b>Baseline:</b> the adams building , also known as the central savings bank building building , is a building located at 418 ashmun street in sault ste. marie , michigan . it was listed on the national register of historic places in 2010 .</td></tr>
<tr><td><b>BASS:</b>the adams building , also known as the central savings bank building , is a commercial building located at 418 ashmun street in sault ste. marie , michigan . it was listed on the national register of historic places in 2010 .</td></tr>
</table>

Table 12: The Adams Building