Knowledge-Based Systems 221 (2021) 106973

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Frame Semantics guided network for Abstractive Sentence Summarization

Yong Guan [a,*], Shaoru Guo [a], Ru Li [a,b,*], Xiaoli Li [c], Hu Zhang [a,b]

[a] School of Computer & Information Technology, Shanxi University, China
[b] Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, Shanxi University, China
[c] Institute for Infocomm Research, A*Star, Singapore

ARTICLE INFO

Article history:
Received 21 November 2020
Received in revised form 16 March 2021
Accepted 17 March 2021
Available online 19 March 2021

Keywords:
Abstractive Sentence Summarization
Frame semantics selection
Frame semantics integration
Neural network

ABSTRACT

Abstractive Text Summarization is an important and practical task, aiming to rephrase the input text into a short version summary, while preserving its same and important semantics. In this paper, we propose a novel Frame Semantics guided network for Abstractive Sentence Summarization (FSum), which is able to learn a better text semantic representation by *selecting more relevant Frame semantics* from text, and integrating *Frame semantic representation* with *text representation* effectively. Extensive experiments demonstrate that our proposed FSum model performs significantly better than existing state-of-the-art techniques on both Gigaword and DUC 2004 benchmark datasets.

1. Introduction

Text Summarization is a critical task in NLP domain, aiming at condensing a text into a short version while preserving its essential semantic information [1]. It is particularly important in the big data era, given that there has been an information explosion in the amount of text data, as well as consumers will need to digest large amount of information in short period of time. Clearly, it can be applied to wide real-world applications, such as news summarization, question answering, headline generation, technical paper abstraction etc.

Text Summarization can be categorized into two tasks, namely, *extractive* summarization and *abstractive* summarization. While extractive summarization directly *copies* most relevant words and phrases from source text, abstractive summarization *shortens* and *paraphrases* the given source text, potentially with new words or different yet semantically equivalent phrases. As such, abstractive summarization could be more flexible to condense the content of given text. Summarization contains *sentence* summarization [2] and *document* summarization [3,4]. The input of the sentence summarization is a sentence, while the input of standard document summarization is a document. In this paper, we focus on sentence summarization, which is different from standard document summarization since it is hard to apply existing techniques in extractive methods, such as extracting sentence features and ranking sentences [5]. As a sentence is typically shorter than a document, sentence summarization needs more fine-grained textual representation to generate high quality summaries.

Abstractive summarization needs human beings or learning models to first understand the overall meaning of the given text, and subsequently rephrase its content, by keeping the same and important semantics while ignoring non-critical details. We observe abstractive summarization has three challenges, namely, (1) distill semantic information of given source sentence, (2) *select important semantic information*, and (3) *integrate semantic information to the source sentence representation* to generate a summary. Clearly, obtaining semantic information or representation is critical to tackle the three challenges.

Recently, neural network models have been proposed, which focus on designing sophisticated model structures. For example, [5,6] designed the selective gate network to reweight the source text representation. [7] integrated reinforcement learning, generative adversarial networks, and recurrent neural networks to improve text generalization representation. [8] applied capsule networks with an adaptive optimizer to enhance the generalization capability from few data points. [9,10] used an extractive technique to weight the copy probability and guided the pointer network to copy important words from the source input. To distinguish salient information, [11] applied a focus-attention mechanism and an independent saliency-selection network in the source encoder. On the other hand, some work

The code (and data) in this article has been certified as Reproducible by Code Ocean: (https://codeocean.com/). More information on the Reproducibility Badge Initiative is available at https://www.elsevier.com/physical-sciences-and-engineering/computer-science/journals.

[*] Corresponding authors at: School of Computer & Information Technology, Shanxi University, China.

E-mail addresses: guanyong0130@163.com (Y. Guan), guoshaoru0928@163.com (S. Guo), liru@sxu.edu.cn (R. Li), xlli@i2r.a-star.edu.sg (X. Li), zhanghu@sxu.edu.cn (H. Zhang).

https://doi.org/10.1016/j.knosys.2021.106973
0950-7051/© 2021 Elsevier B.V. All rights reserved.

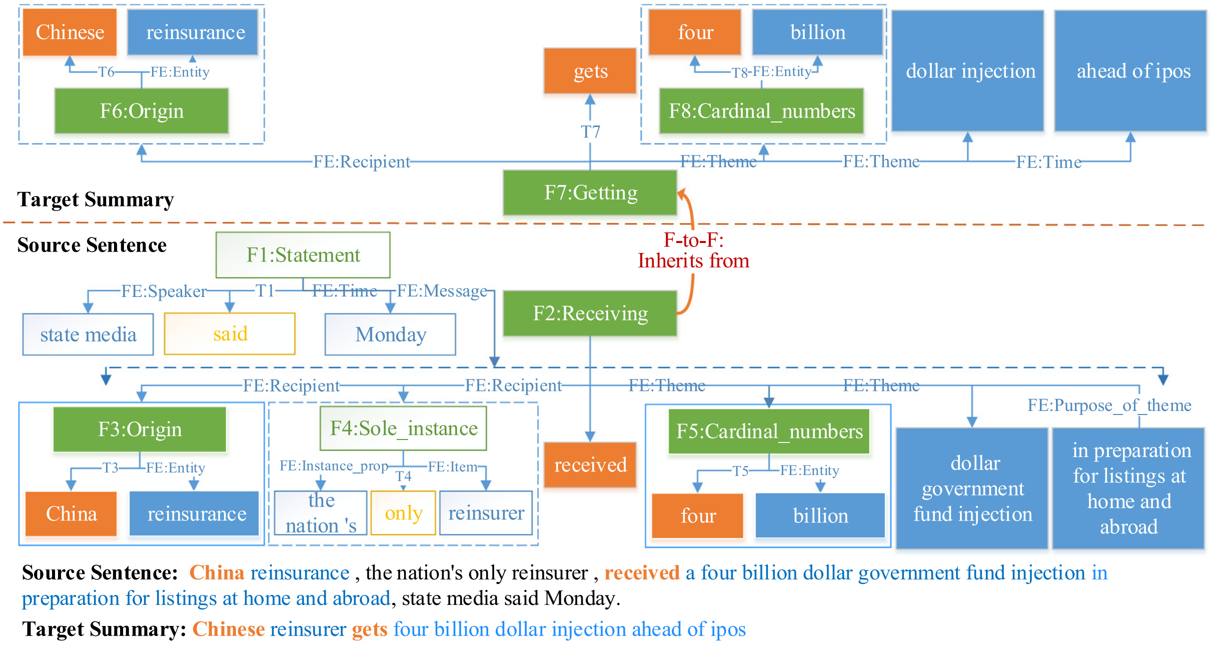

Fig. 1. FrameNet-style parsing of the source sentence and its corresponding summary. Where F [number] indicates the Frame, FE [number] denotes the Frame Element, T [number] refers to the Target word. The color filled modules refer to the related parts between source sentence and target summary.

extracted and integrated entity information into models, or retrieved summary templates to guide summary generation. For instance, [12] proposed a fact aware neural model, which leveraged open information extraction and dependency parse technologies to extract actual fact descriptions as external entity relation knowledge, to guide summary generation. [6] leverages template discovered from training data to softly select key information from each source article to guide its summarization process. [13] extracted entity from the Wikidata knowledge graph and incorporated entity-level knowledge into the encoder–decoder architecture.

Note that the above work always leverage rule-based or triple style (*subject*, *predicate*, *object*) methods to extract important words from source text as important information to guide the summary generation. In practice, however, the word in *source text* may not always appear in *target summary*; instead target summary may use some other semantically related words to better represent the entire semantics of the source sentence. That is, source text is not consistent with the target summary *at the word level*, making existing methods generate lower quality summary. As target summary represents the key information of the source text, they should be consistent *at semantic level*. As such, we propose to first distill/ extract important semantic information from source text, and subsequently use them to guide the summary generation.

We observe that FrameNet [14,15] provides *schematic scenario representation* that could be potentially leveraged to distill the semantics from source sentences. So we propose a novel **FSum** model (Frame Semantics guided model for Abstractive Sentence Summarization), which systematically selects important semantic information and incorporates Frame semantic scenario information into the encoder–decoder architecture to guide the generation of summarization. In particular, *Frame* (F) is defined as a composition of *Lexical Units* (LUs) and a set of *Frame Elements* (FEs). Given a sentence, if its certain word evokes a Frame by matching a LU, then it is called *Target* (T) [16,17]. Taking Frame **Receiving** in Fig. 1 as an example, the word *received* in source sentence evokes the Frame, which contains four FEs, i.e., two *Recipient*, *Theme*, *Purpose_of_theme*. The FE *Theme* is filled by phrase *a four billion dollar government fund injection*. It is worth mentioning that FrameNet connects different relevant Frames into *a Frame network* by defining **Frame-to-Frame** (F-to-F) relations, which facilitate providing natural and effective ways to model the semantic relations between a sentence and its associated summary, as the summary contains semantically relevant rephrased words or phrases. In Fig. 1, given the source sentence "*China reinsurance, the nation's only reinsurer, received a four billion-dollar government fund injection in preparation for listings at home and abroad, state media said Monday*", the Frame semantic parser SEMAFOR [18] distills five semantic scenarios or Frames, namely, F1:**Statement**, F2:**Receiving**, F3:**Origin**, F4: **Sole_instance** and F5: **Cardinal_numbers**. Clearly, not all these semantic scenarios are essential for generating summaries. Hence, we further employ the Frame semantic information in target summary, to help select most important semantic information by leveraging F-to-F relations. For example, F2:**Receiving** (in source sentence) is considered as an important Frame, as it inherits from F7:**Getting** (in target summary). Clearly, F-to-F relations not only help connect important semantics between source text and associated summary, but also can be used for learning what semantics in text are important for summary.

Specifically, we first annotate the given source sentence and target summary into several Frames using the automatic Frame-semantic parser respectively. Then, we utilize additional associated Frame information in target summary to select important Frames (and ignore unimportant Frames) from given sentence. Note, in the process, we take full advantage of F-to-F relations to model the semantic relationship between given sentence and its summary. Finally, an interaction mechanism is proposed to integrate the selected Frame representation and text representation into a better comprehensive representation, which will be fed into decoder to generate accurate summary. To verify the effectiveness of FSum, we conducted extensive experiments on two benchmark data, namely, Gigaword data [19] and DUC 2004 data [20]. Overall, this paper makes the following contributions:

1. We propose a novel FSum model, which, to the best of our knowledge, is the first attempt to utilize Frame semantics to guide the generation of abstractive summarization.

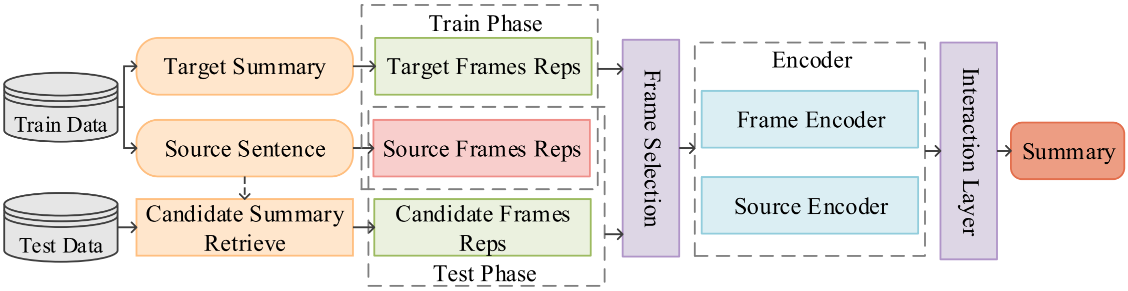

Fig. 2. The flowchart of our proposed FSum model. The dashed line boxes refer to the input of Frame Selection Module, which obtained by different ways during training and testing phase, respectively.

2. We design a new Frame Selection module that selects important and relevant Frames from source sentence to guide the summary generation by leveraging summary Frames and F-to-F relations.
3. We design the interaction mechanism between source sentence representation and Frame representation, which further facilitates learning a better semantic representation.
4. Experiments on Gigaword dataset and DUC 2004 dataset show that our proposed FSum model achieves significantly better results than existing state-of-the-art methods.

2. Related work

Abstractive summarization is a task to generate a short summary that contains the key information of text. Early studies in this task focused on the feature-based traditional machine learning methods, such as syntactic tree pruning [21], statistical machine translation [22] and template methods [23].

Recently, neural network techniques have been adopted to various Natural Language Processing (NLP) tasks, such as combining Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) for Multi-label Text Categorization [24], utilizing deep recurrent belief network to learn Word Dependency [25], applying Long Short-Term Memory (LSTM) networks with the external knowledge for Sentiment Analysis [26]. In summarization task, most neural models employed the encoder–decoder architecture [27]. For instance, [2] applied an attention-based sequence-to-sequence model for abstractive summarization; [28] proposed to use copy-generate mechanism to control when to copy from the source article and when to generate from the vocabulary. [29] avoided repetition of the same words in the summary by proposing a model of a convolutional gated unit that performed global encoding to improve the representation of the input data. [30] introduced Determinantal Point Processes methods to generate comprehensive summaries. [31] defined a pre-training scheme for summarization and produced a zero-shot abstractive summarization model.

2.1. Incorporating external knowledge

Encoder–decoder architecture has attracted attentions and achieved state-of-the-art performance [5,32,33]. Some work have been proposed to incorporate additional knowledge into encoder for better text representation. For instance, [34] proposed to enrich their encoder with handcrafted features, such as named entities and part-of-speech tags, in the embedding lookup layer. [35] encoded the structural information from abstract meaning representation as an additional feature to the source encoder to enhance the source representation. [36] applied a hidden semi-markov model decoder, which learned latent, discrete templates jointly with learning to generate. [37] extracted factual relations from the article to build a knowledge graph and integrated it into the decoding process via neural graph computation.

2.2. Pre-trained language model

Pre-trained transformers with self-supervised objectives on large text corpora have shown great success in text summarization task. For example, BERTShare [38] learned contextualized text representations by predicting *words* based on their context using large amounts of text data. UniLM [33] also predicted a masked word based on its context by jointly pre-trained with multiple language modeling objectives. Some other works attempted to make the models have the foresight to generate a *span* at each step rather than a word. MASS [39] adopted the encoder–decoder Framework and took a sentence with randomly masked fragment (several consecutive tokens) as input, and its decoder predicted this masked fragment. ERNIE-GEN [40] further introduced a span-by-span generation flow that trained the model to predict semantically-complete spans consecutively. In contrast to word-level mask and span-level mask, PEGASUS [41] masked multiple whole sentences from an input document and generated together as one output sequence from the remaining sentences. Different from the above pre-trained models, ProphetNet [42], a sequence-to-sequence pre-trained model, learned to predict future n-gram at each time step.

2.3. FrameNet work on NLP

Recently FrameNet has been adopted for different NLP tasks. For automatic event detection, [43] detected events in FrameNet, and then analyzed possible mappings from Frames to event-types. [44] identified duplicate questions by integrating FrameNet with neural networks. [16] integrated multi-frame semantic information to facilitate sentence modeling for Machine Reading Comprehension (MRC) task. To the best of our knowledge, we have not seen any work integrating it to abstractive sentence summarization so far. Hence, we propose a novel FSum model to utilize Frame Semantics to guide the generation of abstractive summarization.

3. Our proposed method

The overall flowchart and architecture of our proposed FSum model are illustrated in Figs. 2 and 3 respectively, including six key modules:

(1) **Frame Representation** distills frame semantic information F^S from source sentence;

(2) **Frame Selection** selects important Frames F^{sle} in given sentence based on its summary;

(3) **Encoder** represents the selected Frames and given sentence through Frame Encoder $\mathcal{H}^f$ and Source Encoder $\mathcal{H}^c$ respectively;

(4) **Interaction Layer** integrates the Frame representation $\mathcal{H}^f$ and sentence representation $\mathcal{H}^c$ into an overall semantic representation $\mathcal{C}$;

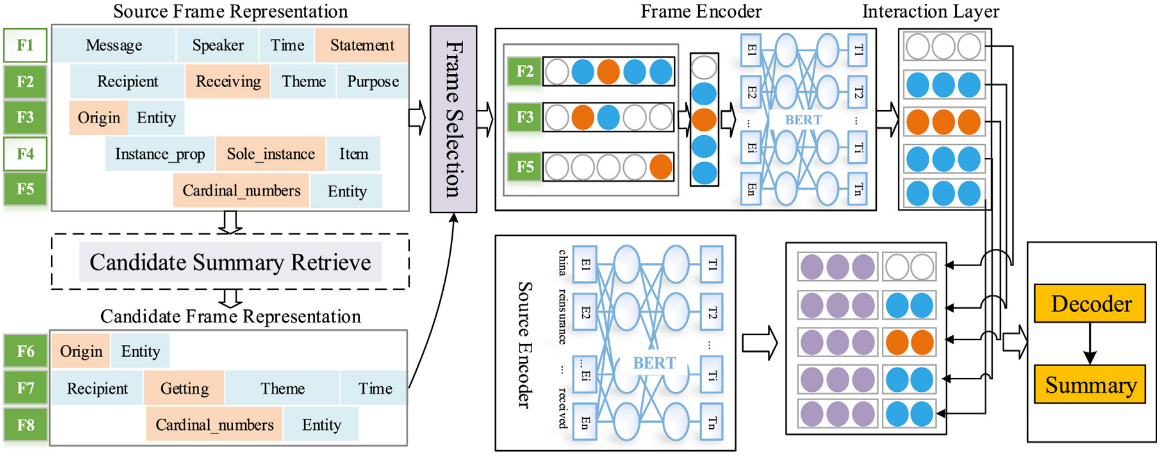

Fig. 3. The overall architecture of FSum model for testing phase. There is no Candidate Summary Retrieve module (dashed line box) in training phase.

(5) **Summary Generation** employs the overall presentation $\mathcal{C}$ to generate its summary y^* word-by-word;

(6) **Candidate Summary Retrieval** retrieves candidate summary X^{sen} from the training corpus during test phase.

Next, we introduce each of the six modules one by one.

3.1. Frame representation

Frame Representation aims to distill Frame semantic information from the given source sentence by a formalized representation of the *Frame semantic structure* F^S. During data preprocessing stage, each sentence is first annotated into several Frame semantic sequences using SEMAFOR 3.0 [18], which is an automatic Frame-semantic parser. More specifically, let F represents a set of Frames in the sentence, FE_i is corresponding Frame Element set of each Frame F_i:

$$F = \{F_1, F_2, \dots, F_i, \dots\} \tag{1}$$

$$FE_i = \{FE_{i1}, FE_{i2}, \dots, FE_{ij}, \dots\} \tag{2}$$

Correspondingly, we can get its multi-Frame semantic structure $F^S = \{F_1^S, F_2^S, \dots, F_i^S, \dots\}$, where $F_i^S = \{F_i, FE_{i1}, FE_{i2}, \dots, FE_{ij}, \dots, \}$ represents the ith Frame semantic structure.

Continuing with the example in Fig. 1, the left part of Fig. 3 shows five source sentence Frames: **Statement**, **Receiving**, **Origin**, **Sole_instance** and **Cardinal_numbers** and three summary Frames **Origin**, **Getting**, **Cardinal_numbers** extracted from source sentence and summary respectively. Each Frame has a Frame semantic structure. For example, the Frame semantic structure of Frame **Statement** is shown as $F^s_{Statement}$ = *{Statement, Message, Speaker, Time}*, where *{ Message, Speaker, Time}* are three Frame Elements of Frame **Statement**.

Next, the Frame semantic structure F_i^S is mapped to an embedding matrix $E(F_i^S)$, i.e. $E(F_i^S) = [e(F_i), e(FE_{i1}), e(FE_{i2}), \dots, e(FE_{ij}), \dots]$, where $e(F_i)$ is the embedding of Frame F_i and $e(FE_{ij})$ is the embedding of Frame Element FE_{ij}. In particular, The Frame and Frame Elements embedding $e(F_i)$ and $e(FE_{ij})$ are initialized by BERT [45], which encode the *Frame definition* F_d and *Frame Element definition* FE_d in FrameNet, e.g., *Statement* is Frame and *Speaker* is a Frame Element of the Frame $F^s_{Statement}$. In FrameNet, *Statement* is defined as *the Frame communicate the act of a Speaker to address a Message to some Addressee using language*, and *Speaker* is *the sentient entity that produces the Message.*[1] Subsequently, we employ the first input token ([CLS]) representation of the last layer as their representation vectors $e(F_i)$, $e(FE_{ij})$ respectively.

$$e(F_i) = BERT(F_d) \tag{3}$$

$$e(FE_{ij}) = BERT(FE_d) \tag{4}$$

3.2. Frame selection

Frame Selection is used to select important semantic information from the source sentence according to target summary. Given the source sentence Frames $F^X = \{F_1^X, F_2^X, \dots, F_M^X\}$ and the target summary Frames $F^{X^{sum}} = \{F_1^{X^{sum}}, F_2^{X^{sum}}, \dots, F_N^{X^{sum}}\}$, where M and N are the number of Frames in the source sentence X and target summary X^{sum} respectively, we will need to train a model that is capable to select most important and relevant Frames F^r in F^X based on $F^{X^{sum}}$. This is because $F^{X^{sum}}$ represents critical Frames that occur in target summary X^{sum}, which should be able to help identify which Frames in source sentence X are important (thus are kept in condensed summary). In this paper, we design two methods for source sentence Frame selection.

3.2.1. Frame path-based method (FPM)

Frame Path-based Method (FPM) aims to take advantage of Frame-to-Frame relations in FrameNet to select important Frames in X. For each target summary Frame, we choose the most relevant sentence Frame if the path length ≤ 2 [46], e.g. in Fig. 1, Frame **Receiving** *Inherits from* the Frame **Getting** (path length is 1). Similarly, Frame **Activity_start** *inherits from* the Frame **Process_start**, which further *inherits from* the **Event** Frame (path length is 2).

3.2.2. Frame similarity-based method (FSM)

Frame Similarity-based Method (FSM) deploys cosine similarity to measure the similarity score between each target summary Frame $F_i^{X^{sum}}$ and source Frames F_m^X. Then, for each $F_i^{X^{sum}}$, we choose the Frame with highest similarity from all source sentence Frames F^X.

$$Sim(i, m) = cosine(c(F_i^{X^{sum}}), c(F_m^X)) \tag{5}$$

$$Sim(i) = [Sim(i, 1), Sim(i, 2), \dots, Sim(i, m)] \tag{6}$$

$$Max\_Sim(i) = Max(Sim(i)) \tag{7}$$

where $Sim(i, m)$ is the score between ith target summary Frame $F_i^{X^{sum}}$ and the mth source sentence Frame F_m^X, and $Sim(i)$ is the

[1] https://framenet.icsi.berkeley.edu/fndrupal/frameIndex.

score set between $F_i^{X^{sum}}$ and F^X. More specifically, we obtain the selected Frames $F^{sle} = \{F_1^{sle}, F_2^{sle}, \ldots, F_N^{sle}\}$ by the similarity score $Max\_Sim(i)$. Note the number of F^{sle} is equal to the number of target summary Frames N. $c(F_i^{X^{sum}})$ is the Frame representation of $F_i^{X^{sum}}$, and is calculated by averaging all elements in its Frame semantic structure $\{e(F_i^{X^{sum}}), e(FE_{i1}^{X^{sum}}), e(FE_{i2}^{X^{sum}}), \ldots, e(FE_{ij}^{X^{sum}}), \ldots\}$.

$$c(F_i^{X^{sum}}) = e(F_i^{X^{sum}}) + \frac{1}{|FE_{ij}^{X^{sum}}|} \sum_{j \in |FE_{ij}^{X^{sum}}|} e(FE_{ij}^{X^{sum}}) \tag{8}$$

where $|FE_{ij}^{X^{sum}}|$ is the Frame Elements number of $F_i^{X^{sum}}$.

3.3. Encoder

Source Encoder computes deep and context-aware representation for the source sentence X. We employ the pre-trained BERT [45] to construct its contextual information for each token via self-attention and produce a sequence of contextual representation $\mathcal{H}^{bx}$.

$$\mathcal{H}^{bx} = BERT(X) \tag{9}$$

During BERT encoding, the input sequence will be segmented to subwords (if any) by BERT word-piece tokenizer. According to [47], we also apply a CNN [48] to reconstruct the contextual word representations $\mathcal{H}^c$.

$$\mathcal{H}^c(t) = \tanh[(\sum_{i=1}^{w-1} \mathcal{H}^{bx}(t+i)^T \eta(i)) + b] \tag{10}$$

where w is the window size, and η is the convolutional filter.

Frame Encoder aims to encoder the selected Frames F^{sle} into a vector representation $\mathcal{H}^f$ to emphasize the important Frames. As shown in Fig. 4, we first compress the selected Frames into one Frame sequence. Specifically, the larger Frame coverage degree[2] means the more information it contains, so we sort the selected Frames according to descending order of the Frame coverage degree. Then, we iteratively replace words with Frame and Frame Elements according to the coverage of every Frame [16], if the words not replaced by the previous Frames. At last, we get the final Frame sequence F^f. As some words (elements) may not be fully covered by the whole selected Frames in F^f, and the words are not replaced. In addition, we apply the pre-trained BERT to construct its contextual representation $\mathcal{H}^{bf}$ and use a CNN to reconstruct the Frame representations $\mathcal{H}^f$. The computational process is as described in Eqs. (9) and (10).

3.4. Interaction mechanism

Interaction Mechanism has been designed, aiming to construct the overall context vector $\mathcal{C}$ by integrating selected Frame representation $\mathcal{H}^f$ to the source sentence representation $\mathcal{H}^c$. We explore three combination approaches.

The first one is called $FSum_{att}$, which adopts attention mechanism like [49] to reweight $\mathcal{H}^c$ based on $\mathcal{H}^f$.

$$\alpha_{tj} = \frac{exp(\mathcal{H}^c(t) \cdot \mathcal{H}^f(j))}{\sum_{j'} exp(\mathcal{H}^c(t) \cdot \mathcal{H}^f(j'))} \tag{11}$$

$$\mathcal{C}(t) = \sum_{j \in |\mathcal{H}^f(j)|} \alpha_{tj} \mathcal{H}^f(j) \tag{12}$$

where α_{tj} is the attention weight for Frame representation attending to source sentence token at time step t, and $\mathcal{H}^f(j)$ is length of the hidden representation $\mathcal{H}^f$.

The second method is denoted as $FSum_{gate}$, where we design a gated network to fuse the source sentence $\mathcal{H}^c$ and selected Frame $\mathcal{H}^f$ [12]. We compute both the Frame-to-Source (F2S) attention $\mathcal{C}^{f2s}$ and Source-to-Frame (S2F) attention $\mathcal{C}^{s2f}$. The attention computation is shown in Eqs. (11) and (12). Then we compute the gate:

$$\mathcal{H}_t^{gate} = \vartheta(\mathcal{C}_t^{f2s}, \mathcal{C}_t^{s2f}) \tag{13}$$

$$\mathcal{C}(t) = \mathcal{H}_t^{gate} \mathcal{C}_t^{f2s} + (1 - \mathcal{H}_t^{gate}) \mathcal{C}_t^{s2f} \tag{14}$$

where ϑ stands for a nonlinear function.

The third approach is called $FSum_{cat}$, which directly concatenates the source sentence representation $\mathcal{H}_t^c$ and the Frame vector $\mathcal{H}_t^f$.

$$\mathcal{C}(t) = [\mathcal{H}_t^c; \mathcal{H}_t^f] \tag{15}$$

3.5. Summary generation

Summary Generation applies the overall presentation $\mathcal{C}$ to generate its summary with a transformer structure [37,50]. As shown in Fig. 5, it utilizes the previous output $y_1, y_2, \ldots, y_{t-1}$, context vector $\mathcal{C}_t$ and Frame vector $\mathcal{H}_t^f$ with attention mechanism to construct the output state Z. For clarity, we first use a multi-head attention to represent the context vector $\mathcal{C}_t$ and the partial generated summary $y_1, y_2, \ldots, y_{t-1}$.

$$Z_{l-1} = concat(head_1, \ldots, head_i, \ldots)W_l^C \tag{16}$$

$$head_i = att(Q_l W_{l,i}^Q, K_l W_{l,i}^K, V_l W_{l,i}^V) \tag{17}$$

where Z_l is the lth layer input for the sub-layer and $l \in L$, W_l^C, $W_{l,i}^Q$, $W_{l,i}^K$ and $W_{l,i}^V$ are four learnable weight matrices, $head_i$ is the ith sub-head of multi-head attention, att is the Scaled Dot-Product Attention. Specifically, query Q_l is obtained by self-attention from the partial generated summary $y_1, y_2, \ldots, y_{t-1}$; key K_l and value V_l are the context vector $\mathcal{C}_t$.

To make the summarization model semantic-aware, we also insert another multi-head attention sub-layer for Frame representation $\mathcal{H}_t^f$. The calculation of integrating frame representation is similar to the Eqs. (16) and (17). The final output Z_l of this sub-layer is obtained with residual connection and layer normalization. The output of last layer Z_L is considered as the final decoder output Z.

Finally, for the tth decoding step, we compute a distribution over the V_{vocab} for the vocabulary distribution $\mathcal{P}(y_t)$ by a linear-softmax operation on z_t.

$$\mathcal{P}(y_t) = softmax(W^o z_t + b^o) \tag{18}$$

where W^o and b^o stand for learnable parameters; $\mathcal{P}(y_t)$ is a probability distribution over all words in the vocabulary V_{vocab}.

3.6. Candidate summary retrieval

During training phase, we directly use the target summary to guide the Frame Selection. For testing purpose, however, the ground truth summary is not available, and we are only given a source sentence X. So we design a **Candidate Summary Retrieval** module to find out the most similar sentences X^{sen} to the source sentence X in the training data X^{tra}, and use the summary of X^{sen} as the candidate summary X^{can}, which is utilized to guide the Frame selection. Specifically, we leverage the widely-used

[2] Frame coverage degree refers to the ratio of words contained in Frame roles (e.g., Target, Frame Element) to the words in the whole sentence. Taking source sentence Frame **Origin** in Fig. 1 as an example, the Frame coverage degree is 2/26, as the total number of words in the source sentence is 26 (length of the source sentence) and the number of Frame annotation words is 2 (China, reinsurance).

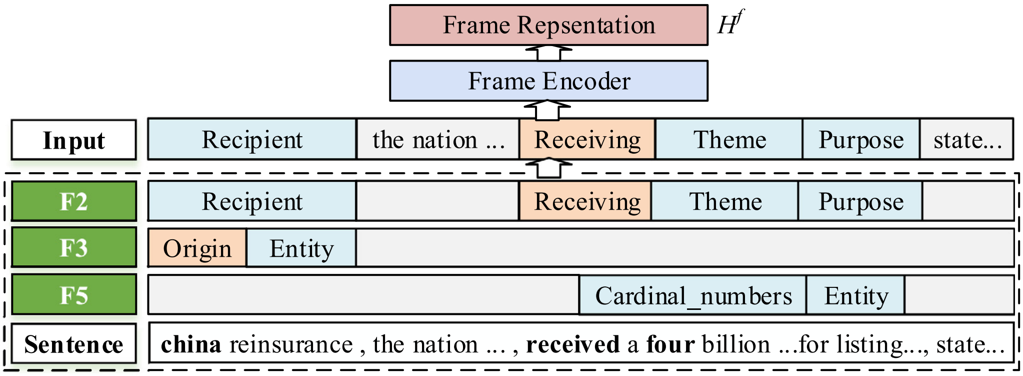

Fig. 4. The frame encoder of FSum model. The light blue refers to Frame Element, and the orange denotes Frame.

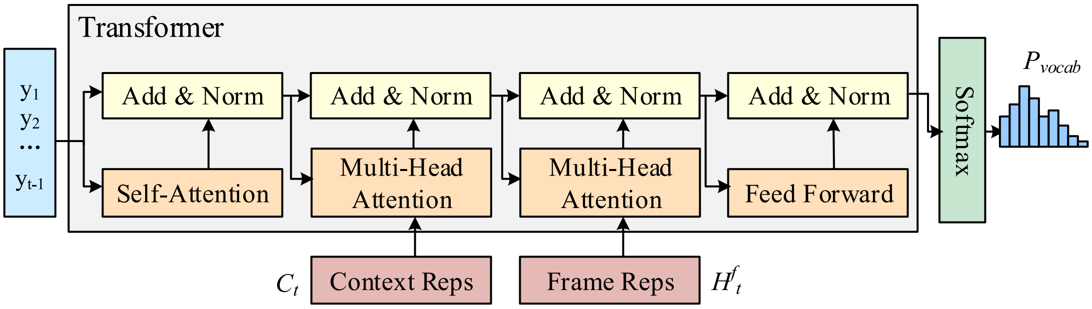

Fig. 5. The decoder of FSum model.

Information Retrieval system Lucene[3] to index and search the similar sentences X^{sen} efficiently.

$$X^{sen} = Top10(Lucene(X, X^{tra})) \tag{19}$$

For each source sentence X, we first select the top 10 sentences as candidate sentences X^{sen}. However, X^{sen} are selected according to Lucene, which only considers the BM25 similarity between the source sentence and the sentences in the training data *at the word level*. However, we want to retrieve sentences that are similar with X *at the semantic level*. So we further utilize the Frame semantic relation to compute the sentence semantic relevant score between X and X^{sen} selected by Lucene.

Given the Frame set F^X of source sentence X, and candidate sentences X^{sen}'s Frame set $F^{X^{sen}}$, we define a function $f(F^X_m, F^{X^{sen}}_{ij})$ to compute Frame semantic similarity between Frame F^X_m and $F^{X^{sen}}_{ij}$, where F^X_m represents the mth Frame of X and $F^{X^{sen}}_{ij}$ ($i = 1, 2,\dots,10$) denotes the jth Frame of ith sentence in X^{sen}. Particularly, the Frame semantic similarity between F^X_m and $F^{X^{sen}}_{ij}$ can be calculated as:

$$f(F^X_m, F^{X^{sen}}_{ij}) = \begin{cases} 1, & \text{if } d(F^X_m, F^{X^{sen}}_{ij}) \leq 2; \\ 0, & other. \end{cases} \tag{20}$$

where $d(F^X_m, F^{X^{sen}}_{ij})$ is the frame relation path length between frame F^X_m and $F^{X^{sen}}_{ij}$. Then, we compute score $Score(F^X, F^{X^{sen}}_i)$ between source sentence Frame set F^x and ith candidate sentence Frame set $F^{X^{sen}}_i$ as:

$$Score(F^X, F^{X^{sen}}_i) = \sum_{m \in |F^X_m|} \sum_{j \in |F^{X^{sen}}_{ij}|} f(F^X_m, F^{X^{sen}}_{ij}) \tag{21}$$

where $|F^X_m|$ and $|F^{X^{sen}}_{ij}|$ are the Frame number of F^X and $F^{X^{sen}}_i$ respectively. Finally, we pick up the most similar sentence, which has the highest relation score $Score(F^X, F^{X^{sen}}_i)$, and use its summary as the candidate summary X^{can}. If $Max(Score(F^X, F^{X^{sen}}_i)) = 0$, we select the most similar sentence by IR system as the candidate summary X^{can}.

Table 1
Data statistics for English Gigaword. AvgSourceLen is the average source sentence length and AvgTargetLen is the average summary length.

Dateset	Train	Dev.	Test
Count	3.8M	189k	1951
AvgSourceLen	31.4	31.7	29.7
AvgTargetLen	8.3	8.3	8.8

3.7. Training objective

Our training objective is to maximize the probability of output summary y^* given the input sentence. Therefore, we adopt the negative log-likelihood as loss function:

$$\mathcal{L} = -\frac{1}{|\mathcal{D}|} \sum_{(x,y^*) \in \mathcal{D}} \log p(y^*|x; \theta) \tag{22}$$

where $\mathcal{D}$ denotes the training dataset and θ stands for the model parameters.

4. Experiments

In this section, we conduct extensive experiments to compare FSum model with state-of-the-art methods on two benchmark datasets.

4.1. Datasets and evaluation metrics

We performed experiments on two widely used benchmark datasets in text summarization, i.e. Annotated English Gigaword dataset [19] and DUC 2004 dataset [20].

[3] https://lucene.apache.org/.

Table 2
F-measures ROUGE socres on Gigaword. We compare our FSum model (the bottom block) with both pre-trained (the top block) and non-pre-trained (the middle block) models.

Method	ROUGE-1	ROUGE-2	ROUGE-L
ProphetNet [42]	39.55	20.27	36.57
ERNIE-GEN [40]	39.25	20.25	36.53
PEGASUS [41]	39.12	19.86	36.24
MASS [39]	38.73	19.71	35.96
UniLM [33]	38.45	19.45	35.75
BERTShare [38]	38.13	19.81	35.62
ABS [2]	30.88	12.22	27.77
ABS+ [2]	31.00	12.65	28.34
ABS+AMR [35]	31.64	12.94	28.54
Featseq2seq [34]	32.67	15.59	30.64
ConvS2S [57]	35.88	17.48	33.29
SEASS [5]	36.15	17.54	33.63
Open-NMT [56]	36.73	17.86	33.68
FTSum [12]	37.27	17.65	34.24
Re3Sum [32]	37.04	19.03	34.46
BiSET [6]	39.11	19.78	36.87
FSum (*FSum*$_{cat}$)	41.56	23.43	38.71

Gigaword pairs the first sentence in the news article and its headline as the summary with heuristic rules [2].[4] As shown in Table 1, it comprises about 3.8M sentence-headline pairs as the training set, 189k pairs as the development set, and 2000 pairs as the test set. For a fair comparison, we use the same vision of test set as [32,33,42], which removes 49 sentence-headline pairs which consist of empty titles in the test set.

DUC 2004 for summarization task (task 1) consists of 500 news articles from the New York Times and Associated Press Wire services, and each article has 4 different human-generated reference summaries.[5] For abstractive sentence summarization, we only use the first sentence of article as input text, following the previous work [2,51–53]. As it contains very small number of instances, we will not train our model on it directly (otherwise the results could be biased as we do not have sufficient training examples to learn a model, as well as sufficient test examples to accurately evaluate the model performance). As such, we directly use the model trained on the Gigaword to test on the DUC 2004 dataset which can also evaluate models' generalization capabilities.

We employ three standard ROUGE metrics [54], including ROUGE-1, ROUGE-2 and ROUGE-L, to measure summary qualities by computing overlaps between the generated summaries and ground truth summaries, in terms of unigram, bigram and longest common subsequence. Existing systems were participated and evaluated using several variants of the F-score-based ROUGE metric on Gigaword data [2,5,55,56] and recall-based ROUGE metric on DUC 2004 data [2,51–53]. In order to keep consistent with the existing work, we apply the three F-score-based ROUGE to evaluate Gigaword data with full length. DUC 2004 data is evaluated by the three recall-based ROUGE scores, and the generated summary is cut-off after 75-characters to make recall-only evaluation unbiased to length.

4.2. Experiment setup

We implement our FSum model in PyTorch.[6] In particular, we use BERT for encoder, whose implementation is based on the PyTorch version.[7] We select Adam [58] as the optimizer, with a learning rate and the dropout probability [59] setting as 5e-5 and 0.3 respectively. Finally, for decoder, we use the beam search of size 3 to generate summaries, and set 12 heads for multi-head attention. Note we conduct our experiment on a machine with a NVIDIA V100 GPU.

4.3. Baselines

We compare our FSum model with the following strong baselines on Gigaword dataset, including both non-pre-trained models and pre-trained models. Here, we only list some models that do not describe in Sections 1 and 2.

ABS [2] uses an attentive CNN encoder and a NNLM decoder to summarize the sentence.

Luong-NMT [55] applies a conditional RNN based on ABS.

ConvS2S [57] introduces an architecture based entirely on CNN.

SEASS [5] applies a selective gate network to control the information flow from encoder to decoder.

Open-NMT [56] implements the standard Seq2Seq model with attention mechanism.

Besides, we compare our model with several pre-training based strong baselines: **PEGASUS** [41], **MASS** [39], **UniLM** [33], **ProphetNet** [42], **ERNIE-GEN** [40], **BERTShare** [38].

In addition, we report the performance of six approaches on the DUC 2004 dataset, including **ABS** [2] and **ABS+** [51], **Featseq2seq** [34], **SEASS** [5], **ERAML** [52], **GLEAM** [53].

4.4. Experimental results

This section reports the experimental results with detailed analysis on two standard benchmark datasets.

4.4.1. Experiments on Gigaword

Table 2 illustrates the comparison results in terms of the ROUGE scores on Gigaword dataset. The results show that FSum model outperforms the baseline models and achieves the best results, in terms of all the three evaluation metrics, e.g., 41.56%, 23.43% and 38.71%, consistently. In addition, we have the following three observations:

(1) Our FSum achieves the best performance, comparing with other 16 state-of-the-art models, indicating frame semantic information is valuable in summary generation.

(2) The results of pre-trained models (top block) are slightly better than the non-pre-trained models (middle block), verifying it is reasonable to choose pre-trained model as our backbone model.

(3) FSum model significantly outperforms pre-trained methods that have leveraged either hand-crafted features (**Featseq2seq** [34]) or summary templates (**Re3Sum,cao-etal-2018-retrieve** and **BiSET,wang-etal-2019-biset**), signifying the importance of choosing relevant and critical sentence Frames based on summary Frames, and learning a better representation by designing interaction mechanisms based on both sentence and summary representations. To affiliate other researchers for performing comparison, we also provide the recall-measures ROUGE scores on Gigaword dataset in Table 3. The results also demonstrate the superiority of our proposed FSum method.

4.4.2. Experiments on DUC 2004

We also evaluate FSum model on human-generated dataset DUC 2004. The detailed experimental results of six state-of-the-art models and our proposed FSum model are listed in Table 4. Again, the results show that FSum model achieves the best performance on all metrics consistently. Therefore, combined with the results on Gigaword, we can conclude that our FSum method

[4] https://github.com/harvardnlp/sent-summary.

[5] http://duc.nist.gov/data.html.

[6] https://pytorch.org/.

[7] https://github.com/huggingface/pytorch-pretrained-BERT.

Table 3
Recall-measures ROUGE socres on Gigaword and DUC 2004 dataset.

Method	Gigaword			DUC 2004		
	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-1	ROUGE-2	ROUGE-L
BERT (Ours)	35.58	18.78	33.68	25.12	8.09	22.11
$FSum_{att}$	36.28	19.75	34.41	26.78	8.86	23.64
$FSum_{gate}$	39.33	22.45	37.31	29.43	10.29	25.57
FSum ($FSum_{cat}$)	39.53	22.77	37.68	29.86	10.72	25.89

Table 4
Recall-measures ROUGE socres on DUC 2004 dataset.

Method	ROUGE-1	ROUGE-2	ROUGE-L
BERT (Ours)	25.12	8.29	22.11
ABS [2]	26.55	7.06	22.05
ABS+ [51]	28.18	8.49	23.81
Featseq2seq [34]	28.61	9.42	25.24
SEASS [5]	29.21	9.56	25.51
ERAML [52]	29.33	10.24	25.24
GLEAM [53]	29.51	9.78	25.60
FSum ($FSum_{cat}$)	29.86	10.72	25.89

is effective by leveraging the Frame semantic information to identify important semantic information from sentences and guide the summary generation. For other people to do the future work, we also list the F-measures ROUGE socres on DUC 2004 dataset in Table 5, our FSum model also achieves a significantly better results.

4.4.3. Effect of interaction layer

In Section 3.4, we explored three alternative approaches to integrating the source sentence representation with the selected Frame representation using Gigaword dataset (similar conclusions can be drawn from DUC 2004 data). The experimental results are given in Table 5, The results show that our FSum ($FSum_{cat}$) model outperforms the other two interaction mechanisms, in terms of all the three evaluation metrics. This is probably because the $FSum_{cat}$ enables the source sentence and selected Frame representation interact with each other through a single self-attention mechanism, while $FSum_{att}$ and $FSum_{gate}$ encode each input sentences separately. For convenience in writing, we use FSum to represent the best performance variation model ($FSum_{cat}$) in the next sections.

4.4.4. Effect of frame selection

To evaluate the effectiveness of Frame Selection, we verify different Frame selection methods using Gigaword dataset:

(1) **NoSlection** directly uses all the Frames in source sentence without any Frame selection methods.

(2) **FPM** only chooses sentence Frames that have a path length ≤ 2 to candidate summary Frames.

(3) **FSM** only uses the similarity score between candidate summary Frames and source sentence Frames, and do not consider the relationship between Frames.

From Table 6, we observe that both **FPM** and **FSM** contribute to the overall performance of our model. No matter which of the two Frame Selection methods we choose, their performance are all better than **NoSlection**, indicating frame selection is valuable in helping select important and relevant Frames from source sentence to guide the summary generation by leveraging Frame-to-Frame relations.

Table 6
Comparison with different frame selection.

Method	ROUGE-1	ROUGE-2	ROUGE-L
FSum	41.56	23.43	38.71
NoSlection	39.83	21.21	36.85
FPM	40.26	21.74	37.16
FSM	40.58	22.12	37.34

4.4.5. Ablation study

Recall that Frame Encoder and Source Encoder are two key modules/components of FSum. To determine their individual effects, we test FSum without these key components.

(1) ***–Frame Encoder***, which removes the Frame Encoder and only use the Source Encoder to obtain the text representation.

(2) ***–Source Encoder***, which removes the Source Encoder and only computes the hidden states of the selected Frame sequence by Frame Encoder to generate summary.

The ablation results in Fig. 6 show that without Frame Encoder, the performance degrade significantly and even worse than some non-pretrained models in Table 2, indicating it is an effective way to distill semantic information of source sentence with frame semantics. We also observe that model without Source Encoder also performs worse than FSum, verifying these two innovative steps play crucial roles for generating high quality summaries.

In Table 7, we present a real example, where a summary generated by FSum model has better quality than –Frame Encoder and –Source Encoder. For instance, –Frame Encoder has wrongly replaced *engineers* to *scientists* and ignored important word *sustained*, while —Source Encoder has ignored the major purpose is to for *sustained development*. By integrating them together, FSum manages to keep the main content of the source sentence and generate an accurate and informative summary successfully.

4.4.6. Human evaluation

Following the existing work [60–62], we conduct human evaluation on three criteria: (1) Informativeness, which evaluates how much concrete information the summary contains; (2) Consistency, which indicates how consistent is the summary to the source sentence; (3) Readability, which evaluates whether the summary follows the grammar and easy to read. We random select 100 test samples on Gigaword data and invite three graduate students with NLP knowledge to rate the generated summaries

Table 5
F-measures ROUGE socres on Gigaword and DUC 2004 dataset.

Method	Gigaword			DUC 2004		
	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-1	ROUGE-2	ROUGE-L
BERT (Ours)	36.08	19.16	34.45	26.78	9.54	24.27
$FSum_{att}$	39.64	21.13	37.86	29.67	11.91	26.04
$FSum_{gate}$	41.24	23.11	38.28	31.38	13.24	27.32
FSum ($FSum_{cat}$)	41.56	23.43	38.71	31.75	13.57	27.61

Table 7
An example of summaries generated by different models. –Frame Encoder and –Source Encoder represent our proposed FSum model without Frame Encoder and Source Encoder, respectively.

Source sentence	Chinese vice premier Huang Ju said here Wednesday that worldwide engineers should cooperate with each other to contribute more to sustained development of the human society.
Target summary	Engineers urged to contribute more to sustained development Chinese vice premier
Source frames	Statement, Locative_relation, Giving, Increment, Collaboration,Being_obligated, Event, People_by_vocation, Leadership, Origin, People
Candidate frames	Request, Giving, Increment, Leadership, Event, People_by_vocation
Selected frames	Statement, Giving, Increment, Leadership, Event, People_by_vocation
–Frame Encoder	Vice premier urges scientists to contribute more to development
–Source Encoder	Chinese vice premier urges global engineers to cooperate
FSum	Chinese vice premier urges worldwide engineers to contribute more to sustained development

Table 8
The results of human evaluation.

Method	Informativeness	Consistency	Readability
BERT	4.22	4.15	4.37
$FSum_{att}$	4.29	4.22	4.39
$FSum_{gate}$	4.37	4.29	4.47
$FSum$ ($FSum_{cat}$)	4.41	4.34	4.52
Ground truth	4.54	4.49	4.57

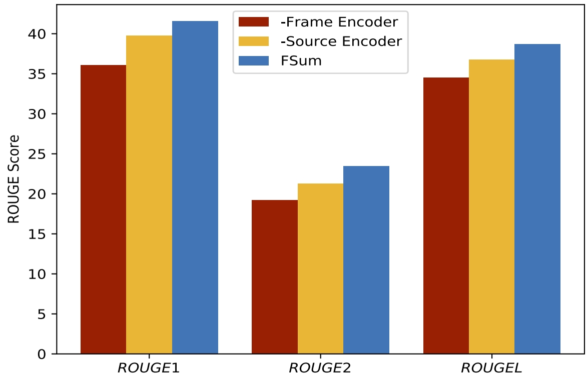

Fig. 6. Ablation study of FSum model.

as well as ground truth summaries on a scale of 1 to 5. They are given both summaries and source sentences, and unaware of the identities of the different models. Then, we average the scores of each summary from human.

The results are shown in Table 8, which shows the superiority of our proposed FSum method. We observe that our FSum model achieves 4.41, 4.34 and 4.52 in terms of three evaluation metrics, which are 0.19, 0.19, and 0.15 better than BERT, indicating our model can generate better quality summaries by integrating Frame semantics information. Note the differences between our proposed method and ground truth are relatively small, again signifying the advantage of the proposed method.

5. Conclusion and future work

In this paper, we propose a novel FSum model to introduce Frame Semantic information to guide sentence summarization. In particular, we design a new method to select important and relevant Frames that are critical for generating summary in given source sentence, through Frame-to-Frame relations and the similarity between source sentence Frames and summary Frames. In addition, an interaction mechanism has been proposed to integrate source sentence representation and Frame representation into a comprehensive semantic representation. The extensive experimental results demonstrate that our model outperforms all the baseline models significantly and consistently on two benchmark datasets.

There are three potential future directions to extend our work. Firstly, our method can be improved by integrating external knowledge, such as knowledge graph, to supplementary Frame semantic information, inspired by related models [12,37]. Secondly, we are interested in the automatic summary evaluation, inspired by the correlation between source text and target summary which provide by FrameNet. Finally, in this paper, we focus on the effectiveness of integrating Frame semantics knowledge to abstractive summarization, so we introduce the widely used BERT and transformer to construct our framework. Nevertheless, our model can extend to different types of neural networks and verify their performance on abstractive summarization, such as capsule networks [8], deep recurrent belief network [25], reinforcement learning and generative adversarial networks [7], Graph Convolutional Networks (GCN) [63].

CRediT authorship contribution statement

Yong Guan: Conceptualization, Methodology, Software, Writing - original draft. **Shaoru Guo:** Investigation, Editing. **Ru Li:** Conceptualization, Supervision. **Xiaoli Li:** Writing - review & editing. **Hu Zhang:** Validation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was sponsored by the National Natural Science Foundation of China (No. 61936012, No. 61772324).

References

[1] I. Mani, Automatic Summarization, Vol. 3, John Benjamins Publishing, 2001.
[2] A.M. Rush, S. Chopra, J. Weston, A neural attention model for abstractive sentence summarization, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Lisbon, Portugal, 2015, pp. 379–389, http://dx.doi.org/10.18653/v1/D15-1044, URL https://www.aclweb.org/anthology/D15-1044.
[3] A. Nenkova, K. McKeown, Automatic summarization, Found. Trends Inform. Retrieval 5 (2011) 103–233, http://dx.doi.org/10.1561/1500000015.
[4] X. Zhang, F. Wei, M. Zhou, HIBERT: Document level pre-training of hierarchical bidirectional transformers for document summarization, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 5059–5069, http://dx.doi.org/10.18653/v1/P19-1499, URL https://www.aclweb.org/anthology/P19-1499.
[5] Q. Zhou, N. Yang, F. Wei, M. Zhou, Selective encoding for abstractive sentence summarization, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 1095–1104, http://dx.doi.org/10.18653/v1/P17-1101, URL https://www.aclweb.org/anthology/P17-1101.
[6] K. Wang, X. Quan, R. Wang, BiSET: Bi-directional selective encoding with template for abstractive summarization, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 2153–2162, http://dx.doi.org/10.18653/v1/P19-1207, URL https://www.aclweb.org/anthology/P19-1207.

[7] W. Zhao, H. Peng, S. Eger, E. Cambria, M. Yang, Towards scalable and reliable capsule networks for challenging NLP applications, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 1549–1559, http://dx.doi.org/10.18653/v1/P19-1150.

[8] Y. Li, Q. Pan, S. Wang, T. Yang, E. Cambria, A generative model for category text generation, Inform. Sci. 450 (2018) 301–315, http://dx.doi.org/10.1016/j.ins.2018.03.050.

[9] W.-T. Hsu, C.-K. Lin, M.-Y. Lee, K. Min, J. Tang, M. Sun, A unified model for extractive and abstractive summarization using inconsistency loss, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 132–141, http://dx.doi.org/10.18653/v1/P18-1013, URL https://www.aclweb.org/anthology/P18-1013.

[10] S. Gehrmann, Y. Deng, A. Rush, Bottom-up abstractive summarization, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 4098–4109, http://dx.doi.org/10.18653/v1/D18-1443, URL https://www.aclweb.org/anthology/D18-1443.

[11] Y. You, W. Jia, T. Liu, W. Yang, Improving abstractive document summarization with salient information modeling, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 2132–2141.

[12] Z. Cao, F. Wei, W. Li, S. Li, Faithful to the original: Fact aware neural abstractive summarization, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 32, (1), 2018.

[13] B. Gunel, C. Zhu, M. Zeng, X. Huang, Mind the facts: Knowledge-boosted coherent abstractive text summarization, in: NeurIPS 2019, 2019, URL https://www.microsoft.com/en-us/research/publication/mind-the-facts-knowledge-boosted-coherent-abstractive-text-summarization/.

[14] C.J. Fillmore, et al., Frame semantics and the nature of language, in: Annals of the New York Academy of Sciences: Conference on the Origin and Development of Language and Speech, Vol. 280, (1) 1976, pp. 20–32.

[15] C.F. Baker, C.J. Fillmore, J.B. Lowe, The berkeley framenet project, in: 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1, Association for Computational Linguistics, Montreal, Quebec, Canada, 1998, pp. 86–90, http://dx.doi.org/10.3115/980845.980860, URL https://www.aclweb.org/anthology/P98-1013.

[16] S. Guo, R. Li, H. Tan, X. Li, Y. Guan, H. Zhao, Y. Zhang, A frame-based sentence representation for machine reading comprehension, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 891–896, http://dx.doi.org/10.18653/v1/2020.acl-main.83, URL https://www.aclweb.org/anthology/2020.acl-main.83.

[17] S. Guo, Y. Guan, R. Li, X. Li, H. Tan, Incorporating Syntax and Frame Semantics in Neural Network for Machine Reading Comprehension, in: Proceedings of the 28th International Conference on Computational Linguistics, 2020, pp. 2635–2641.

[18] D. Das, D. Chen, A.F. Martins, N. Schneider, N.A. Smith, Frame-semantic parsing, Comput. Linguist. 40 (1) (2014) 9–56.

[19] C. Napoles, M. Gormley, B. Van Durme, Annotated gigaword, in: Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-Scale Knowledge Extraction, Association for Computational Linguistics, 2012, pp. 95–100.

[20] P. Over, H. Dang, D. Harman, DUC in context, Inf. Process. Manage. 43 (6) (2007) 1506–1520.

[21] K. Knight, D. Marcu, Summarization beyond sentence extraction: A probabilistic approach to sentence compression, Artificial Intelligence 139 (1) (2002) 91–107.

[22] M. Banko, V.O. Mittal, M.J. Witbrock, Headline generation based on statistical translation, in: Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics, 2000, pp. 318–325.

[23] L. Zhou, E. Hovy, Template-filtered headline summarization, in: Text Summarization Branches Out, 2004, pp. 56–60.

[24] G. Chen, D. Ye, Z. Xing, J. Chen, E. Cambria, Ensemble application of convolutional and recurrent neural networks for multi-label text categorization, in: Proceedings of the International Joint Conference on Neural Networks, 2017, pp. 2377–2383.

[25] I. Chaturvedi, Y.-S. Ong, I.W. Tsang, R.E. Welsch, E. Cambria, Learning word dependencies in text by means of a deep recurrent belief network, Knowl.-Based Syst. 108 (2016) 144–154, http://dx.doi.org/10.1016/j.knosys.2016.07.019.

[26] Y. Ma, H. Peng, T. Khan, E. Cambria, A. Hussain, Sentic LSTM: a hybrid network for targeted aspect-based sentiment analysis, Cogn. Comput. 10 (4) (2018) 639–650, http://dx.doi.org/10.1007/s12559-018-9549-x.

[27] I. Sutskever, O. Vinyals, Q.V. Le, Sequence to sequence learning with neural networks, in: Advances in Neural Information Processing Systems, 2014, pp. 3104–3112.

[28] A. See, P.J. Liu, C.D. Manning, Get To The Point: Summarization with Pointer-Generator Networks, Association for Computational Linguistics, 2017, pp. 1073–1083, URL http://dblp.uni-trier.de/db/conf/acl/acl2017-1.html#SeeLM17,

[29] J. Lin, X. Sun, S. Ma, Q. Su, Global encoding for abstractive summarization, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 163–169, http://dx.doi.org/10.18653/v1/P18-2027, URL https://www.aclweb.org/anthology/P18-2027.

[30] L. Li, W. Liu, M. Litvak, N. Vanetik, Z. Huang, In conclusion not repetition: Comprehensive abstractive summarization with diversified attention based on determinantal point processes, in: Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 822–832, http://dx.doi.org/10.18653/v1/K19-1077, URL https://www.aclweb.org/anthology/K19-1077.

[31] C. Zhu, Z. Yang, R. Gmyr, M. Zeng, X. Huang, Make lead bias in your favor: A simple and effective method for news summarization, 2019, arXiv preprint arXiv:1912.11602.

[32] Z. Cao, W. Li, S. Li, F. Wei, Retrieve, rerank and rewrite: Soft template based neural summarization, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 152–161, http://dx.doi.org/10.18653/v1/P18-1015, URL https://www.aclweb.org/anthology/P18-1015.

[33] L. Dong, N. Yang, W. Wang, F. Wei, X. Liu, Y. Wang, J. Gao, M. Zhou, H.-W. Hon, Unified language model pre-training for natural language understanding and generation, in: Advances in Neural Information Processing Systems, 2019, pp. 13063–13075.

[34] R. Nallapati, B. Zhou, C. Gulcehre, B. Xiang, et al., Abstractive text summarization using sequence-to-sequence RNNs and beyond, in: Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, Association for Computational Linguistics, Berlin, Germany, 2016, pp. 280–290, http://dx.doi.org/10.18653/v1/K16-1028, URL https://www.aclweb.org/anthology/K16-1028.

[35] S. Takase, J. Suzuki, N. Okazaki, T. Hirao, M. Nagata, Neural headline generation on abstract meaning representation, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, 2016, pp. 1054–1059.

[36] S. Wiseman, S. Shieber, A. Rush, Learning neural templates for text generation, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 3174–3187, http://dx.doi.org/10.18653/v1/D18-1356, URL https://www.aclweb.org/anthology/D18-1356.

[37] C. Zhu, W. Hinthorn, R. Xu, Q. Zeng, M. Zeng, X. Huang, M. Jiang, Boosting factual correctness of abstractive summarization with knowledge graph, 2020, arXiv preprint arXiv:2003.08612.

[38] S. Rothe, S. Narayan, A. Severyn, Leveraging pre-trained checkpoints for sequence generation tasks, Trans. Assoc. Comput. Linguist. 8 (2020) 264–280.

[39] K. Song, X. Tan, T. Qin, J. Lu, T.-Y. Liu, MASS: Masked sequence to sequence pre-training for language generation, in: K. Chaudhuri, R. Salakhutdinov (Eds.), Proceedings of the 36th International Conference on Machine Learning, in: Proceedings of Machine Learning Research, vol. 97, PMLR, Long Beach, California, USA, 2019, pp. 5926–5936, URL http://proceedings.mlr.press/v97/song19d.html.

[40] D. Xiao, H. Zhang, Y. Li, Y. Sun, H. Tian, H. Wu, H. Wang, ERNIE-GEN: An enhanced multi-flow pre-training and fine-tuning framework for natural language generation, 2020, arXiv:2001.11314.

[41] J. Zhang, Y. Zhao, M. Saleh, P. Liu, Pegasus: Pre-training with extracted gap-sentences for abstractive summarization, in: International Conference on Machine Learning, PMLR, 2020, pp. 11328–11339.

[42] Y. Yan, W. Qi, Y. Gong, D. Liu, N. Duan, J. Chen, R. Zhang, M. Zhou, Prophetnet: Predicting future n-gram for sequence-to-sequence pre-training, 2020, arXiv preprint arXiv:2001.04063.

[43] S. Liu, Y. Chen, S. He, K. Liu, J. Zhao, Leveraging framenet to improve automatic event detection, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2016, pp. 2134–2143.

[44] X. Zhang, X. Sun, H. Wang, Duplicate question identification by integrating framenet with neural networks, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[45] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2018, arXiv preprint arXiv:1810.04805.

[46] R. Li, J. Wu, Z. Wang, Q. Chai, Implicit role linking on chinese discourse: Exploiting explicit roles and frame-to-frame relations, in: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 2015, pp. 1263–1271.

[47] Z. Zhang, Y. Wu, H. Zhao, Z. Li, S. Zhang, X. Zhou, X. Zhou, Semantics-aware BERT for language understanding, in: The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, the Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, the Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, AAAI Press, New York, NY, USA, 2020, pp. 9628–9635, URL https://aaai.org/ojs/index.php/AAAI/article/view/6510.

[48] M. Feng, B. Xiang, M.R. Glass, L. Wang, B. Zhou, Applying deep learning to answer selection: A study and an open task, in: 2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), IEEE, 2015, pp. 813–820.

[49] L. Perez-Beltrachini, Y. Liu, M. Lapata, Generating summaries with topic templates and structured convolutional decoders, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 5107–5116, http://dx.doi.org/10.18653/v1/P19-1504, URL https://www.aclweb.org/anthology/P19-1504.

[50] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, 2017, pp. 5998–6008.

[51] S. Chopra, M. Auli, A.M. Rush, Abstractive sentence summarization with attentive recurrent neural networks, in: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, San Diego, California, 2016, pp. 93–98, http://dx.doi.org/10.18653/v1/N16-1012, URL https://www.aclweb.org/anthology/N16-1012.

[52] H. Li, J. Zhu, J. Zhang, C. Zong, Ensure the correctness of the summary: Incorporate entailment knowledge into abstractive sentence summarization, in: Proceedings of the 27th International Conference on Computational Linguistics, Association for Computational Linguistics, Santa Fe, New Mexico, USA, 2018, pp. 1430–1441, URL https://www.aclweb.org/anthology/C18-1121.

[53] Y. Gao, Y. Wang, L. Liu, Y. Guo, H. Huang, Neural abstractive summarization fusing by global generative topics, Neural Comput. Appl. (2019) 1–10.

[54] C.-Y. Lin, E. Hovy, Automatic evaluation of summaries using n-gram co-occurrence statistics, in: Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, 2003, pp. 150–157.

[55] S. Chopra, M. Auli, A.M. Rush, Abstractive sentence summarization with attentive recurrent neural networks, in: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2016, pp. 93–98.

[56] G. Klein, Y. Kim, Y. Deng, J. Senellart, A. Rush, OpenNMT: Open-source toolkit for neural machine translation, in: Proceedings of ACL 2017, System Demonstrations, Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 67–72, URL https://www.aclweb.org/anthology/P17-4012.

[57] J. Gehring, M. Auli, D. Grangier, D. Yarats, Y.N. Dauphin, Convolutional sequence to sequence learning, in: Proceedings of the 34th International Conference on Machine Learning - Volume 70, in: ICML'17, JMLR.org, 2017, pp. 1243–1252.

[58] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6980.

[59] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (1) (2014) 1929–1958.

[60] W. Li, J. Xu, Y. He, S. Yan, Y. Wu, X. Sun, Coherent comments generation for chinese articles with a graph-to-sequence model, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 4843–4852, http://dx.doi.org/10.18653/v1/P19-1479, URL https://www.aclweb.org/anthology/P19-1479.

[61] N. Vanetik, M. Litvak, E. Churkin, M. Last, An unsupervised constrained optimization approach to compressive summarization, Inform. Sci. 509 (2020) 22–35, http://dx.doi.org/10.1016/j.ins.2019.08.079, URL http://www.sciencedirect.com/science/article/pii/S0020025516312506.

[62] L. Pan, Y. Xie, Y. Feng, T.-S. Chua, M.-Y. Kan, Semantic graphs for generating deep questions, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 1463–1475, http://dx.doi.org/10.18653/v1/2020.acl-main.135, URL https://www.aclweb.org/anthology/2020.acl-main.135.

[63] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: Proceedings of the 5th International Conference on Learning Representations, in: ICLR '17, 2017, URL https://openreview.net/forum?id=SJU4ayYgl.