Structure-Infused Copy Mechanisms for Abstractive Summarization

Kaiqiang Song
Computer Science Dept.
University of Central Florida
Orlando, FL 32816, USA
kqsong@knights.ucf.edu

Lin Zhao
Research and Tech. Center
Robert Bosch LLC
Sunnyvale, CA 94085, USA
lin.zhao@us.bosch.com

Fei Liu
Computer Science Dept.
University of Central Florida
Orlando, FL 32816, USA
feiliu@cs.ucf.edu

Abstract

Seq2seq learning has produced promising results on summarization. However, in many cases, system summaries still struggle to keep the meaning of the original intact. They may miss out important words or relations that play critical roles in the syntactic structure of source sentences. In this paper, we present structure-infused copy mechanisms to facilitate copying important words and relations from the source sentence to summary sentence. The approach naturally combines source dependency structure with the copy mechanism of an abstractive sentence summarizer. Experimental results demonstrate the effectiveness of incorporating source-side syntactic information in the system, and our proposed approach compares favorably to state-of-the-art methods.

1 Introduction

Recent years have witnessed increasing interest in abstractive summarization. The systems seek to condense source texts to summaries that are concise, grammatical, and preserve the important meaning of the original texts (Nenkova and McKeown, 2011). The task encompasses a number of high-level text operations, e.g., paraphrasing, generalization, text reduction and reordering (Jing and McKeown, 1999), posing a considerable challenge to natural language understanding.

Src	A Mozambican man suspect of murdering Jorge Microsse, director of Maputo central prison, has ***escaped*** from the city's police headquarters, local media reported on Tuesday.
Ref	Mozambican suspected of killing Maputo prison director ***escapes***
Sys	mozambican man ***arrested*** for murder
Src	An Alaska father who was too drunk to drive ***had*** his 11-year-old son take the wheel, authorities said.
Ref	Drunk Alaska dad ***has*** 11 year old drive home
Sys	alaska father who was too drunk to drive

Table 1: Example source sentences, reference and system summaries produced by a neural attentive seq-to-seq model. System summaries fail to preserve summary-worthy content of the source (e.g., main verbs) despite their syntactic importance.

The sequence-to-sequence learning paradigm has achieved remarkable success on abstractive summarization (Rush et al., 2015; Nallapati et al., 2016; See et al., 2017; Paulus et al., 2017). While the results are impressive, individual system summaries can appear unreliable and fail to preserve the meaning of the source texts. Table 1 presents two examples. In these cases, the syntactic structure of source sentences is relatively *rare* but perfectly normal. The first sentence contains two appositional phrases ("suspect of murdering Jorge Microsse," "director of Maputo central prison") and the second sentence has a relative clause ("who was too drunk to drive"), both located between the subject and the main verb. The system, however, fails to identify the main verb in both cases; it instead chooses to focus on the first few words of the source sentences. We observe that *rare syntactic constructions* of the source can pose problems for neural summarization systems, possibly for two reasons. First, similar to *rare words*, certain syntactic constructions do not occur frequently enough in the training data to allow the system to learn the patterns. Second, neural summarization systems are not explicitly informed of the syntactic structure of the source sentences and they tend to bias towards sequential recency.

This work is licensed under a Creative Commons Attribution 4.0 International License. License details: http://creativecommons.org/licenses/by/4.0/.

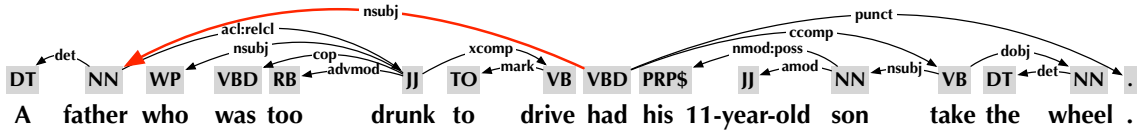


Figure 1: An example dependency parse tree created for the source sentence in Table 1. If important dependency edges such as "father ← had" can be preserved in the summary, the system summary is likely to preserve the meaning of the original.

In this paper we seek to address this problem by incorporating source syntactic structure in neural sentence summarization to help the system identify summary-worthy content and compose summaries that preserve the important meaning of the source texts. We present structure-infused copy mechanisms to facilitate copying source words and relations to the summary based on their semantic and structural importance in the source sentences. For example, if important parts of the source syntactic structure, such as a dependency edge from the main verb to the subject ("father" ← "had," shown in Figure 1), can be preserved in the summary, the "missing verb" issue in Table 1 can be effectively alleviated. Our model therefore learns to recognize important source words and source dependency relations and strives to preserve them in the summaries. Our research contributions include the following:

- we introduce novel neural architectures that encourage salient source words/relations to be preserved in summaries. The framework naturally combines the dependency parse tree structure with the copy mechanism of an abstractive summarization system. To the best of our knowledge, this is the first attempt at comparing various neural architectures for this purpose;
- we study the effectiveness of several important components, including the vocabulary size, a coverage-based regularizer (See et al., 2017), and a beam search with reference mechanism (Tan et al., 2017);
- through extensive experiments we demonstrate that incorporating syntactic information in neural sentence summarization is effective. Our approach surpasses state-of-the-art published systems on the benchmark dataset.[1]

2 Related Work

Prior to the deep learning era, sentence syntactic structure has been utilized to generate summaries with an "*extract-and-compress*" framework. Compressed summaries are generated using a joint model to extract sentences and drop non-important syntactic constituents (Daume III and Marcu, 2002; Berg-Kirkpatrick et al., 2011; Thadani and McKeown, 2013; Durrett et al., 2016), or a pipeline approach that combines generic sentence compression (McDonald, 2006; Clarke and Lapata, 2008; Filippova et al., 2015) with a sentence pre-selection or post-selection process (Zajic et al., 2007; Galanis and Androutsopoulos, 2010; Wang et al., 2013; Li et al., 2013; Li et al., 2014). Although syntactic information is helpful for summarization, there has been little prior work investigating how best to combine sentence syntactic structure with the neural abstractive summarization systems.

Existing neural summarization systems handle syntactic structure only implicitly (Kikuchi et al., 2016; Chen et al., 2016; Zhou et al., 2017; Tan et al., 2017; Paulus et al., 2017). Most systems adopt a "*cut-and-stitch*" scheme that picks words either from the vocabulary or the source text and stitch them together using a recurrent language model. However, there lacks a mechanism to ensure structurally salient words and relations in source sentences are preserved in the summaries. The resulting summary sentences can contain misleading information (e.g., "mozambican man arrested for murder" flips the meaning of the original) or grammatical errors (e.g., verbless, as in "alaska father who was too drunk to drive").

Natural language generation (NLG)-based abstractive summarization (Carenini and Cheung, 2008; Gerani et al., 2014; Fabbrizio et al., 2014; Liu et al., 2015; Takase et al., 2016) also makes extensive use of structural information, including syntactic/semantic parse trees, discourse structures, and domain-specific templates built using a text planner or an OpenIE system (Pighin et al., 2014). In particular, Cao et al. (2018) leverage OpenIE and dependency parsing to extract fact tuples from the source text and use those to improve the faithfulness of summaries.

[1]We made our system publicly available at: `https://github.com/KaiQiangSong/struct_infused_summ`

Different from the above approaches, this paper seeks to directly incorporate source-side syntactic structure in the copy mechanism of an abstractive sentence summarization system. It learns to recognize important source words and relations during training, while striving to preserve them in the summaries at test time to aid reproduction of factual details. Our intent of incorporating source syntax in summarization is different from that of neural machine translation (NMT) (Li et al., 2017a; Chen et al., 2017), in part because NMT does not handle the information loss from source to target. In contrast, a summarization system must selectively preserve source content to render concise and grammatical summaries. We specifically focus on sentence summarization, where the goal is to reduce the first sentence of an article to a title-like summary. We believe even for this reasonably simple task there remains issues unsolved.

3 Our Approach

We seek to transform a source sentence $\mathbf{x}$ to a summary sentence $\mathbf{y}$ that is concise, grammatical, and preserves the meaning of the source sentence. A source word is replaced by its Glove embedding (Pennington et al., 2014) before it is fed to the system; the vector is denoted by $\mathbf{x}_i$ ($i \in [S]$; 'S' for source). Similarly, a summary word is denoted by $\mathbf{y}_t$ ($t \in [T]$; 'T' for target). If a word does not appear in the input vocabulary, it is replaced by a special '$\langle \text{unk} \rangle$' token. We begin this section by describing the basic summarization framework, followed by our new copy mechanisms used to encourage source words and dependency relations to be preserved in the summary.

3.1 The Basic Framework

We build an encoder-decoder architecture for this work. An encoder condenses the entire source text to a continuous vector; it also learns a vector representation for each unit of the source text (e.g., words as units). In this work we use a two-layer stacked bi-directional Long Short-Term Memory (Hochreiter and Schmidhuber, 1997) networks as the encoder, where the input to the second layer is the concatenation of hidden states from the forward and backward passes of the first layer. We obtain the hidden states of the second layer; they are denoted by $\mathbf{h}_i^e$. The source text vector is constructed by averaging over all $\mathbf{h}_i^e$ and passing the vector through a feedforward layer with $\tanh$ activation to convert from the encoder hidden states to an initial decoder hidden state ($\mathbf{h}_0^d$). This process is illustrated in Eq. (2).

$$\mathbf{h}_i^e = f_e(\mathbf{h}_{i-1}^e, \mathbf{x}_i) \quad \mathbf{h}_t^d = f_d(\mathbf{h}_{t-1}^d, \mathbf{y}_{t-1}) \tag{1}$$

$$\mathbf{h}_0^d = \tanh(\mathbf{W}^{h_0} \frac{1}{S} \sum_{i=1}^{S} \mathbf{h}_i^e + \mathbf{b}^{h_0}) \tag{2}$$

A decoder unrolls the summary by predicting one word at a time. During training, the decoder takes as input the embeddings of ground truth summary words, denoted by $\mathbf{y}_t$, while at test time $\mathbf{y}_t$ are embeddings of system predicted summary words (i.e., teacher forcing). We implement an LSTM decoder with the attention mechanism. A context vector $\mathbf{c}_t$ is used to encode the source words that the system attends to for generating the next summary word. It is defined in Eqs (3-5), where $[\cdot||\cdot]$ denotes the concatenation of two vectors. The $\boldsymbol{\alpha}$ matrix measures the strength of interaction between the decoder hidden states $\{\mathbf{h}_t^d\}$ and encoder hidden states $\{\mathbf{h}_i^e\}$. To predict the next word, the context vector $\mathbf{c}_t$ and $\mathbf{h}_t^d$ are concatenated and used as input to build a new vector $\widetilde{\mathbf{h}}_t^d$ (Eq. (6)). $\widetilde{\mathbf{h}}_t^d$ is a surrogate for semantic meanings carried at time step t of the decoder. It is subsequently used to compute a probability distribution over the output vocabulary (Eq. (7)).

$$e_{t,i} = \mathbf{v}^\top \tanh(\mathbf{W}^e[\mathbf{h}_t^d||\mathbf{h}_i^e] + b^e) \tag{3}$$

$$\alpha_{t,i} = \frac{\exp(e_{t,i})}{\sum_{i'=1}^{S} \exp(e_{t,i'})} \tag{4}$$

$$\mathbf{c}_t = \textstyle\sum_{i=1}^{S} \alpha_{t,i} \mathbf{h}_i^e \tag{5}$$

$$\widetilde{\mathbf{h}}_t^d = \tanh(\mathbf{W}^h[\mathbf{h}_t^d||\mathbf{c}_t] + \mathbf{b}^h) \tag{6}$$

$$P_{vocab}(w) = \text{softmax}(\mathbf{W}^y \widetilde{\mathbf{h}}_t^d + \mathbf{b}^y) \tag{7}$$

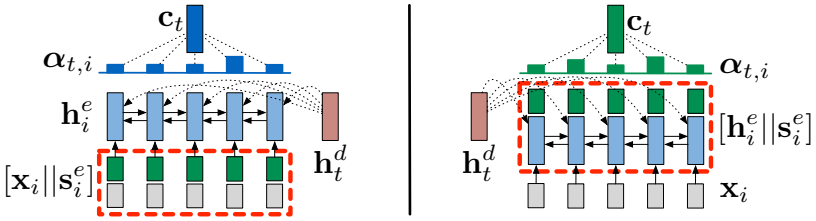


Figure 2: System architectures for 'Struct+Input' (left) and 'Struct+Hidden' (right). A critical question we seek to answer is whether the structural embeddings ($\mathbf{s}_i^e$) should be supplied as input to the encoder (left) or be exempted from encoding and directly concatenated with the encoder hidden states (right).

The copy mechanism (Gulcehre et al., 2016; See et al., 2017) allows words in the source sequence to be selectively copied to the target sequence. It expands the search space for summary words to include both the output vocabulary and the source text. The copy mechanism can effectively reduce out-of-vocabulary tokens in the generated text, potentially aiding a number of applications such as MT (Luong et al., 2015b) and text summarization (Gu et al., 2016; Cheng and Lapata, 2016; Zeng et al., 2017).

Our copy mechanism employs a 'switch' to estimate the likelihood of generating a word from the vocabulary (p_{gen}) vs. copying it from the source text ($1 - p_{gen}$). The basic model is similar to that of the pointer-generator networks (See et al., 2017). The switch is a feedforward layer with sigmoid activation (Eq. (8)). At time step t, its input is a concatenation of the decoder hidden state $\mathbf{h}_t^d$, context vector $\mathbf{c}_t$, and the embedding of the previously generated word $\mathbf{y}_{t-1}$. For predicting the next word, we combine the generation and copy probabilities, shown in Eq. (9). If a word w appears once or more in the input text, its copy probability ($\sum_{i:w_i=w} \alpha_{t,i}$) is the sum of the attention weights over all its occurrences. If w appears in both the vocabulary and source text, $P(w)$ is a weighted sum of the two probabilities.

$$p_{gen} = \sigma(\mathbf{W}^z[\mathbf{h}_t^d||\mathbf{c}_t||\mathbf{y}_{t-1}]) + b^z) \tag{8}$$

$$P(w) = p_{gen}P_{vocab}(w) + (1 - p_{gen}) \sum_{i:w_i=w} \alpha_{t,i} \tag{9}$$

3.2 Structure-Infused Copy Mechanisms

The aforementioned copy mechanism attends to source words based on their "semantic" importance encoded in $\{\alpha_{t,i}\}$, which measures the semantic relatedness of the encoder hidden state $\mathbf{h}_i^e$ and the decoder hidden state $\mathbf{h}_t^d$ (Eq. (4)). However, the source syntactic structure is ignored. This is problematic, because it hurts the system's ability to effectively identify summary-worthy source words that are syntactically important. We next propose three strategies to inject source syntactic structure to the copy mechanism.

3.2.1 Shallow Combination

Inspired by compressive summarization via structured prediction (Berg-Kirkpatrick et al., 2011; Almeida and Martins, 2013), we hypothesize that structural labels, such as the incoming dependency arc and the depth in a dependency parse tree, can be helpful to predict word importance. We consider six categories of structural labels in this work; they are presented in Table 2. Each structural label is mapped to a fixed-length, trainable structural embedding. However, a critical question remains as to where the structural embeddings should be injected in the existing neural architecture. This problem has not yet been systematically investigated. In this work, we compare two settings:

Structural info	Example
(1) depth in the dependency parse tree	0
(2) label of the incoming edge	'root'
(3) number of outgoing edges	3
(4) part-of-speech tag	'VBD'
(5) absolution position in the source text	9
(6) relative position in the source text	(0.5, 0.6]

Table 2: Six categories of structural labels. Example labels are generated for word 'had' in Figure 1. Relative word positions are discretized into ten buckets.

- **Struct+Input** concatenates structural embeddings of position i (flattened into one vector $\mathbf{s}_i^e$) with the source word embedding $\mathbf{x}_i$ and uses them as a new form of input to the encoder: $\mathbf{x}_i \Rightarrow [\mathbf{x}_i||\mathbf{s}_i^e]$;

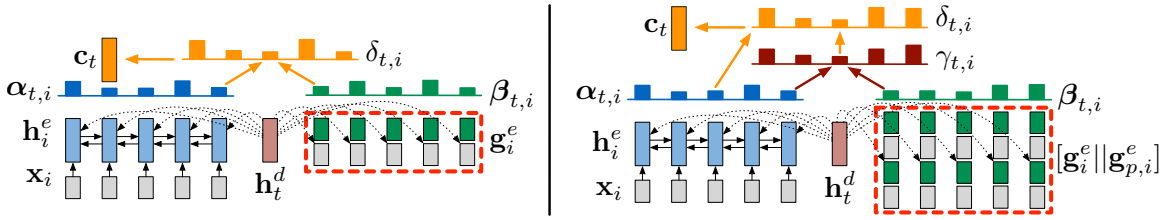


Figure 3: System architectures for 'Struct+2Way+Word' (left) and 'Struct+2Way+Relation' (right). $\beta_{t,i}$ (left) measures the structural importance of the i-th source word; $\beta_{t,i}$ (right) measures the saliency of the dependency edge pointing to the i-th source word. $\mathbf{g}_{p,i}^e$ is the structural embedding of the parent. In both cases $\delta_{t,i}$ replaces $\alpha_{t,i}$ to become the new attention value used to estimate the context vector $\mathbf{c}_t$.

- **Struct+Hidden** concatenates structural embeddings of position i (flattened) with the encoder hidden state $\mathbf{h}_i^e$ and uses them as a new form of hidden states: $\mathbf{h}_i^e \Rightarrow [\mathbf{h}_i^e||\mathbf{s}_i^e]$.

The architectural difference is illustrated in Figure 2. Structural embeddings are important complements to existing neural architectures. However, it is unclear whether they should be supplied as input to the encoder or be left out of the encoding process and directly concatenated with the encoder hidden states. This is a critical question we seek to answer by comparing the two settings. Note that an alternative setting is to separately encode words and structural labels using two RNN encoders, we consider this as a subproblem of the "Struct+Input" case.

The above models complement state-of-the-art by combining semantic and structural signals to determine summary-worthy content. Intuitively, a source word is copied to the summary for two reasons: it contains salient semantic content, or it serves a critical syntactic role in the source sentence. Without explicitly modeling the two factors, 'semantics' can outweigh 'structure,' resulting in summaries that fail to keep the original meaning intact. In the following we propose a two-way mechanism to separately model the "semantic" and "structural" importance of source words.

3.2.2 2-Way Combination (+Word)

Our new architecture involves two attention matrices that are parallel to each other, denoted by $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. $\alpha_{t,i}$ is defined as previously in Eq. (3-4). It represents the "semantic" aspect, calculated as the strength of interaction between the encoder hidden state $\mathbf{h}_i^e$ and the decoder hidden state $\mathbf{h}_t^d$. In contrast, $\beta_{t,i}$ measures the "structural" importance of the i-th input word to generating the t-th output word, calculated by comparing the structure-enhanced embedding $\mathbf{g}_i^e$ with the decoder hidden state $\mathbf{h}_t^d$ (Eq. (10-11)). We use $\mathbf{g}_i^e = [\mathbf{s}_i^e||\mathbf{x}_i]$ as a primitive (unencoded) representation of the i-th source word.

We define $\delta_{t,i} \propto \alpha_{t,i} + \epsilon\beta_{t,i}$ as a weighted sum of $\alpha_{t,i}$ and $\beta_{t,i}$, where a trainable coefficient ϵ is introduced to balance the contribution from both sides (Eq. (12)). Merging semantic and structural salience at this stage allows us to acquire an accurate estimate of how important the i-th source word is to predicting the t-th output word. $\delta_{t,i}$ replaces $\alpha_{t,i}$ to become the new attention value. It is used to calculate the context vector $\mathbf{c}_t$ (Eq. (13)). A reliable estimate of $\mathbf{c}_t$ is crucial as it is used to estimate the generation probability over the vocabulary ($P_{vocab}(w)$, Eq. (6-7)), the switch value (p_{gen}, Eq. (8)), and ultimately used to predict the next word ($P(w)$, Eq. (9)).

$$f_{t,i} = \mathbf{u}^\top \tanh(\mathbf{W}^f[\mathbf{g}_i^e||\mathbf{h}_t^d] + \mathbf{b}^f) \tag{10}$$

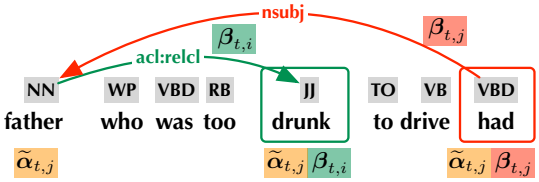
$$\beta_{t,i} = \frac{\exp(f_{t,i})}{\sum_{i'=1}^{S}\exp(f_{t,i'})} \tag{11}$$

$$\delta_{t,i} = \frac{\alpha_{t,i} + \epsilon\beta_{t,i}}{\sum_{i'=1}^{S}(\alpha_{t,i'} + \epsilon\beta_{t,i'})} \tag{12}$$

$$\mathbf{c}_t = \sum_{i=1}^{S}\delta_{t,i}\mathbf{h}_i^e \tag{13}$$

3.2.3 2-Way Combination (+Relation)

We observe that salient source relations also play a critical role in predicting the next word. For example, if a dependency edge ("father" $\xleftarrow{\text{nsubj}}$ "had") is salient and "father" is selected to be included in the summary, it is likely that "had" will be selected next such that a salient source relation ("nsubj") is preserved in the summary. Because summary words tend to follow the word order of the original, we assume selecting a source word and including it in the summary has an impact on its subsequent source words, but not the reverse.



In this formulation we use $\beta_{t,i}$ to capture the saliency of the dependency edge pointing to the i-th source word. Thus, an edge $w_j \leftarrow w_i$ has its salience score saved in $\beta_{t,j}$; and conversely, an edge $w_j \rightarrow w_i$ has its salience score in $\beta_{t,i}$. $\boldsymbol{\beta}$ is calculated in the same way as described in Eq. (10-11). However, we replace $\mathbf{g}_i^e$ with $[\mathbf{g}_i^e||\mathbf{g}_{p,i}^e]$ so that a dependency edge is characterized by the embeddings of its two endpoints ($\mathbf{g}_{p,i}^e$ is the parent embedding). The architectural difference between "Struct+2Way+Word" and "Struct+2Way+Relation" is illustrated in Figure 3.

To obtain the likelihood of w_j being selected to the summary prior to time step t, we define $\widetilde{\alpha}_{t,j} = \sum_{t'=0}^{t-1} \alpha_{t',j}$ that sums up the individual probabilities up to time step t-1. Assume there is a dependency edge $w_j \rightarrow w_i$ (j<i) whose salience score is denoted by $\beta_{t,i}$. At time step t, we calculate $\widetilde{\alpha}_{t,j}\beta_{t,i}$ (or $\widetilde{\alpha}_{t,j}\beta_{t,j}$ for edge $w_j \leftarrow w_i$) as the probability of w_i being selected to the summary, given that *one* of its prior words w_j (j<i) is included in the summary and there is a dependency edge connecting the two. By summing the impact over *all* its previous words, we obtain the likelihood of the i-th source word being included to the summary at time step t in order to preserve salient source relations; this is denoted by $\gamma_{t,i}$ (Eq. (15)). Next, we define $\delta_{t,i} \propto \alpha_{t,i} + \epsilon\gamma_{t,i}$ as a weighted combination of semantic and structural salience (Eq. (16)). $\delta_{t,i}$ replace $\alpha_{t,i}$ to become the new attention values used to estimate the context vector $\mathbf{c}_t$ (Eq. (13)). Finally, the calculation of generation probabilities $P_{vocab}(w)$, switch value p_{gen}, and probabilities for predicting the next word $P(w)$ remains the same as previously (Eq. (6-9)).

$$\widetilde{\alpha}_{t,j} = \sum_{t'=0}^{t-1} \alpha_{t',j} \tag{14}$$

$$\gamma_{t,i} = \sum_{j:j<i} \begin{cases} \widetilde{\alpha}_{t,j}\beta_{t,i} & \text{if } w_j \rightarrow w_i \\ \widetilde{\alpha}_{t,j}\beta_{t,j} & \text{if } w_j \leftarrow w_i \end{cases} \tag{15}$$

$$\delta_{t,i} = \frac{\alpha_{t,i} + \epsilon\gamma_{t,i}}{\sum_{i'=1}^{S}(\alpha_{t,i'} + \epsilon\gamma_{t,i'})} \tag{16}$$

3.3 Learning Objective and Beam Search

We next describe our learning objective, including a coverage-based regularizer (See et al., 2017), and a beam search with reference mechanism (Tan et al., 2017). We want to investigate the effectiveness of these techniques on sentence summarization, which has not been explored in previous work.

Learning objective. Our training proceeds by minimizing a per-target-word cross-entropy loss function. A regularization term is applied to the $\boldsymbol{\alpha}$ matrix. Recall that $\alpha_{t,i} \in [0, 1]$ measures the interaction strength between the t-th output word and the i-th input word. Naturally, we expect a 1-to-1 mapping between the two words. The coverage-based regularizer, proposed by See et al., (2017), encourages this behavior by tracking the historical attention values attributed to the i-th input word (up to time step t-1), denoted by $\widetilde{\alpha}_{t,i} = \sum_{t'=0}^{t-1} \alpha_{t',i}$. The approach then takes the minimum between $\widetilde{\alpha}_{t,i}$ and $\alpha_{t,i}$, which has the practical effect of forcing $\alpha_{t,i}$ ($\forall t$) to be close to either 0 or 1, otherwise a penalty will be applied. The regularizer Ω is defined in Eq. (17), where M is the size of the mini-batch, S and T are the lengths of the source

and target sequences. For two-way copy mechanisms, $\boldsymbol{\delta}$ replaces $\boldsymbol{\alpha}$ to become the new attention values, we therefore apply regularization to $\boldsymbol{\delta}$ instead of $\boldsymbol{\alpha}$. When the regularizer applies, the objective becomes minimizing $(\mathcal{L} + \Omega)$.

$$\Omega = \lambda \sum_{m=1}^{M} \frac{1}{T^{(m)} S^{(m)}} \sum_{t=1}^{T^{(m)}} \sum_{i=1}^{S^{(m)}} \Big(\min(\widetilde{\alpha}_{t,i}, \alpha_{t,i}) \Big) \tag{17}$$

Beam search with reference. During testing, we employ greedy search to generate system summary sequences. For the task of summarization, the ground truth summary sequences are usually close to the source texts. This property can be leveraged in beam search. Tan et al., (2017) describe a beam search with reference mechanism that rewards system summaries that have a high degree of bigram overlap with the source texts. We describe it in Eq. (18), where where $\mathcal{S}(w)$ denotes the score of word w. $\mathcal{B}(\mathbf{y}_{<t}, \mathbf{x})$ measures the number of bigrams shared by the system summary (up to time step t-1) and the source text; $\{\mathbf{y}_{<t}, w\}$ adds a word w to the end of the system summary. The shorter the source text (measured by length S), the more weight a shared bigram will add to the score of the current word w. A hyperparameter η controls the degree of closeness between the system summary and the source text.

$$\mathcal{S}(w) = \log P(w) + \eta \frac{\mathcal{B}(\{\mathbf{y}_{<t}, w\}, \mathbf{x}) - \mathcal{B}(\mathbf{y}_{<t}, \mathbf{x})}{S} \tag{18}$$

4 Experiments

We evaluate the proposed structure-infused copy mechanisms for summarization in this section. We describe the dataset, experimental settings, baselines, and finally, evaluation results and analysis.

4.1 Data Sets

We evaluate our proposed models on the Gigaword summarization dataset (Parker, 2011; Rush et al., 2015). The task is to reduce the first sentence of an article to a title-like summary. We obtain dependency parse trees for source sentences using the Stanford neural network parser (Chen and Manning, 2014). We also use the standard train/valid/test data splits. Following (Rush et al., 2015), the train and valid splits are pruned[2] to improve the data quality. Spurious pairs that are repetitive, overly long/short, and pairs whose source and summary sequences have little word overlap are removed. No pruning is performed for instances in the test set. The processed corpus contains 4,018K training instances. We construct two (non-overlapped) validation sets: "valid-4096" contains 4,096 randomly sampled instances from the valid split; it is used for hyperparameter tuning and early stopping. "valid-2000" is used for evaluation; it allows the models to be trained and evaluated on pruned instances. Finally, we report results on the standard Gigaword test set (Rush et al., 2015) containing 1,951 instances ("test-1951").

4.2 Experimental Setup

We use the Xavier scheme (Glorot and Bengio, 2010) for parameter initialization, where weights are initialized using a Gaussian distribution $\mathbf{W}_{i,j} \sim \mathcal{N}(0,\, \sigma)$, $\sigma = \sqrt{\frac{2}{n_{in}+n_{out}}}$; n_{in} and n_{out} are numbers of the input and output units of the network; biases are set to be 0. We further implement two techniques to accelerate mini-batch training. First, all training instances are sorted by the source sequence length and partitioned into mini-batches. The shorter sequences are padded to have the same length as the longest sequence in the batch. All batches are shuffled at the beginning of each epoch. Second, we introduce a variable-length batch vocabulary containing only source words of the current mini-batch and words of the output vocabulary. $P(w)$ in Eq. (9) only needs to be calculated for words in the batch vocabulary. It is magnitudes smaller than a direct combination of the input and output vocabularies. Finally, our input vocabulary contains the most frequent 70K words in the source texts and summaries. The output vocabulary contains 5K words by default. More network parameters are presented in Table 3.

[2] https://github.com/facebookarchive/NAMAS/blob/master/dataset/filter.py

Input vocabulary size	70K
Output vocabulary size	5K (default)
Dim. of word embeddings	100
Dim. of structural embeddings	16
Num. of encoder/decoder hidden units	256
Adam optimizer (Kingma and Ba, 2015)	lr = 1e-4
Coeff. for coverage-based regularizer	$\lambda = 1$
Coeff. for beam search with reference	$\eta \approx 13.5$
Beam size	$K = 5$
Minibatch size	$M = 64$
Early stopping criterion (max 20 epochs)	valid. loss
Gradient clipping (Pascanu et al., 2013)	g $\in$ [-5, 5]

Table 3: Parameter settings of our summarization system.

System	Gigaword Valid-2000 R-1	R-2	R-L
Baseline	42.48	21.34	40.18
Struct+Input	42.44	21.75	40.46
Struct+Hidden	42.88	21.81	40.63
Struct+2Way+Word	**43.21**	21.84	**40.86**
Struct+2Way+Relation	42.83	**21.85**	40.60

Table 4: Results on the Gigaword valid-2000 set (full-length F1). Models implementing the structure-infused copy mechanisms ("Struct+*") outperform the baseline.

S:	the government filed another round of criminal charges in a widening stock options scandal
T:	options scandal widens
B:	government files more charges in stock options scandal
I:	another round of criminal charges in stock options scandal
H:	charges filed in stock options scandal
W:	another round of criminal charges in stock options scandal
R:	government files another round of criminal charges in options scandal

Table 5: Example system summaries. 'S:' source; 'T:' target; 'B:' baseline; 'I:' Struct+Input; 'H:' Struct+Hidden; 'W:' 2Way+Word; "R:" 2Way+Relation. "2Way+Relation" is able to preserve important source relations in the summary, e.g., "government ~~‹nsubj~~ files," "files ~~dobj›~~ round," and "round ~~nmod›~~ charges."

S:	red cross negotiators from rivals north korea and south korea held talks wednesday on emergency food shipments to starving north koreans and agreed to meet again thursday
T:	koreas meet in beijing to discuss food aid from south eds
B:	north korea , south korea agree to meet again
I:	north korea , south korea meet again
H:	north korea , south korea meet on emergency food shipments
W:	north korea , south korea hold talks on food shipments
R:	north korea , south korea hold talks on emergency food shipments

Table 6: Example system summaries. "Struct+Hidden" and "2Way+Relation" successfully preserve salient source words ("emergency food shipments"), which are missed out by other systems. We observe that copying "hold talks" from the source also makes the resulting summaries more informative than using the word "meet."

4.3 Results

ROUGE results on valid set. We first report results on the Gigaword valid-2000 dataset in Table 4. We present R-1, R-2, and R-L scores (Lin, 2004) that respectively measures the overlapped unigrams, bigrams, and longest common subsequences between the system and reference summaries[3]. Our baseline system ("Baseline") implements the seq2seq architecture with the basic copy mechanism (Eq. (1-9)). It is a strong baseline that resembles the pointer-generator networks described in (See et al., 2017). The structural models ("Struct+*") differ from the baseline only on the structure-infused copy mechanisms. All models are evaluated without the coverage regularizer or beam search (§3.3) to ensure fair comparison. Overall, we observe that models equipped with the structure-infused copy mechanisms are superior to the baseline, suggesting that combining source syntactic structure with the copy mechanism is effective. We found that the "Struct+Hidden" architecture, which directly concatenates structural embeddings with the encoder hidden states, outperforms "Struct+Input" despite that the latter requires more parameters. "Struct+2Way+Word" also demonstrates strong performance, achieving 43.21%, 21.84%, and 40.86% F_1 scores, for R-1, R-2, and R-L respectively.

ROUGE results on test set. We compare our proposed approach with a range of state-of-the-art neural summarization systems. Results on the standard Gigaword test set ("test-1951") are presented in Table 7. Details about these systems are provided in Table 8. Overall, our proposed approach with structure-infused pointer networks perform strongly, yielding ROUGE scores that are on-par with or surpassing state-of-the-art published systems. Notice that the scores on the valid-2000 dataset are generally higher than those of test-1951. This is because the (source, summary) pairs in the Gigaword test set are not pruned (see §4.1). In some cases, none (or very few) of the summary words appear in the source. This may cause difficulties to the systems equipped with the copy mechanism. The "Struct+2Way+Word" architecture that respectively models the semantic and syntactic importance of source words achieves the highest scores. It outperforms its counterpart of "Struct+2Way+Relation," which seeks to preserve source dependency relations in summaries. We conjecture that the imperfect dependency parse trees generated

[3] w/ ROUGE options: -n 2 -m -w 1.2 -c 95 -r 1000

System	Gigaword Test-1951 R-1	R-2	R-L
ABS (Rush et al., 2015)	29.55	11.32	26.42
ABS+ (Rush et al., 2015)	29.76	11.88	26.96
Luong-NMT (Chopra et al., 2016)	33.10	14.45	30.71
RAS-LSTM (Chopra et al., 2016)	32.55	14.70	30.03
RAS-Elman (Chopra et al., 2016)	33.78	15.97	31.15
ASC+FSC1 (Miao and Blunsom, 2016)	34.17	15.94	31.92
lvt2k-1sent (Nallapati et al., 2016)	32.67	15.59	30.64
lvt5k-1sent (Nallapati et al., 2016)	35.30	16.64	32.62
Multi-Task (Pasunuru et al., 2017)	32.75	15.35	30.82
DRGD (Li et al., 2017b)	36.27	17.57	33.62
Baseline (this paper)	35.43	17.49	33.39
Struct+Input (this paper)	35.32	17.50	33.25
Struct+2Way+Relation (this paper)	35.46	17.51	33.28
Struct+Hidden (this paper)	**35.49**	17.61	33.33
Struct+2Way+Word (this paper)	35.47	**17.66**	**33.52**

Table 7: Results on the Gigaword test-1951 set (full-length F1). Models with structure-infused copy mechanisms ("Struct+*") perform well. Their R-2 F-scores are on-par with or outperform state-of-the-art published systems.

ABS and **ABS+** (Rush et al., 2015) are the first work introducing an encoder-decoder architecture for summarization.

Luong-NMT (Chopra et al., 2016) is a re-implementation of the attentive stacked LSTM encoder-decoder of Luong et al. (2015a).

RAS-LSTM and **RAS-Elman** (Chopra et al., 2016) describe a convolutional attentive encoder that ensures the decoder focuses on appropriate words at each step of generation.

ASC+FSC1 (Miao and Blunsom, 2016) presents a generative auto-encoding sentence compression model jointly trained on labelled/unlabelled data.

lvt2k-1sent and **lvt5k-1sent** (Nallapati et al., 2016) address issues in the attentive encoder-decoder framework, including modeling keywords, capturing sentence-to-word structure, and handling rare words.

Multi-Task w/ Entailment (Pasunuru et al., 2017) combines entailment with summarization in a multi-task setting.

DRGD (Li et al., 2017b) describes a deep recurrent generative decoder learning latent structure of summary sequences via variational inference.

Table 8: Existing summarization methods.

by the parser may affect the "Struct+2Way+Relation" results. However, because the Gigaword dataset does not provide gold-standard annotations for parse trees, we could not easily verify this and will leave it for future work. In Table 5 and 6, we present system summaries produced by various models.

Linguistic quality. To further gauge the summary quality, we hire human workers from the Amazon Mechanical Turk platform to rate summaries on a Likert scale of 1 to 5 according to three criteria (Zhang and Lapata, 2017): *fluency* (is the summary grammatical and well-formed?), *informativeness* (to what extent is the meaning of the original sentence preserved in the summary?), and *faithfulness* (is the summary accurate and faithful to the original?). We sample 100 instances from the test set and employ 5 turkers to rate each summary; their averaged scores are presented in Table 9. We found that "Struct+2Way+Relation" outperforms "Struct+Input" on all three criteria. It also compares favorably to ground-truth summaries on "fluency" and "faithfulness." On the other hand, the ground-truth summaries, corresponding to article titles, are judged as less satisfying according to human raters.

System	Info.	Fluency	Faithful.
Struct+Input	2.9	3.3	3.0
Struct+2Way+Relation	3.0	3.4	3.1
Ground-truth Summ.	3.2	3.5	3.1

Table 9: Informativeness, fluency, and faithfulness scores of summaries. They are rated by Amazon turkers on a Likert scale of 1 (worst) to 5 (best). We choose to evaluate Struct+2Way+Relation (as oppose to 2Way+Word) because it focuses on preserving source relations in the summaries.

Dependency relations. We investigate the source dependency relations preserved in the summaries in Table 10. A source relation is considered preserved if both its words appear in the summary. We observe that the models implementing structure-infused copy mechanisms (e.g., "Struct+2Way+Word") are more likely to preserve important dependency relations in the summaries, including nsubj, dobj, amod, nmod, and nmod:poss. Dependency relations that are less important (mark, case, conj, cc, det) are less likely to be preserved. These results show that our structure-infused copy mechanisms can learn to recognize the importance of dependency relations and selectively preserve them in the summaries.

Coverage and reference beam. In Figure 11, we investigate the effect of applying the coverage regularizer ("coverage") and reference-based beam search ("ref_beam") (§3.3) to our models. The coverage regularizer is applied in a second training stage, where the system is trained for an extra 5 epochs with coverage and the model yielding the lowest validation loss is selected. Both coverage and ref_beam can improve the system performance. Our observation suggests that ref_beam is an effective addition to shorten the gap between different systems.

System	nsubj	dobj	amod	nmod	nmod:poss	mark	case	conj	cc	det
Baseline	7.23	12.07	20.45	8.73	12.46	15.83	14.84	9.72	5.03	2.22
Struct+Input	7.03	11.72	19.72	**9.17**↑	12.46	15.35	14.69	9.55	4.67	1.97
Struct+Hidden	**7.78**↑	**12.34**↑	**21.11**↑	**9.18**↑	**14.86**↑	14.93	**15.84**↑	9.47	3.93	**2.65**↑
Struct+2Way+Word	**7.46**↑	**12.69**↑	**20.59**↑	**9.03**↑	**13.00**↑	15.83	14.43	8.86	3.48	1.91
Struct+2Way+Relation	**7.35**↑	**12.07**↑	**20.59**↑	8.68	**13.47**↑	15.41	14.39	9.12	4.30	1.89

Table 10: Percentages of source dependency relations (of various types) preserved in the system summaries.

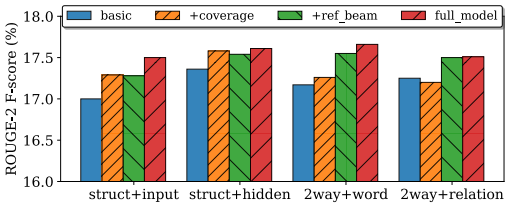


Table 11: Effects of applying the coverage regularizer and the reference beam search to structural models, evaluated on test-1951. Combining both yields the highest scores.

$\|V\|$	R-2	Train Speed	InVcb	InVcb+Src
1K	13.99	2.5h/epoch	60.57	76.04
2K	15.35	2.7h/epoch	69.71	80.72
5K	17.25	3.2h/epoch	79.98	86.51
10K	17.62	3.8h/epoch	88.26	92.18

Table 12: Results of the "Struct+2Way+Relation" system trained using output vocabularies of various sizes ($|V|$), evaluated on test-1951 w/o coverage or ref_beam. The training speed is calculated as the elapsed time (hours) per epoch, tested on a GTX 1080Ti GPU card.

Output vocabulary size. Finally, we investigate the impact of the output vocabulary size on the summarization performance in Table 12. All our models by default use an output vocabulary of 5K words in order to make the results comparable to state-of-the-art-systems. However, we observe that there is a potential to further boost the system performance (17.25→17.62 R-2 F1-score, w/o coverage or ref_beam) if we had chosen to use a larger vocabulary (10K) and can endure a slightly longer training time (1.2x). In Table 12, we further report the percentages of reference summary words covered by the output vocabulary ("InVcb") and covered by either the output vocabulary or the source text ("InVcb+Src"). The gap between the two conditions shortens as the size of the output vocabulary is increased.

5 Conclusion

In this paper, we investigated structure-infused copy mechanisms that combine source syntactic structure with the copy mechanism of an abstractive summarization system. We compared various system architectures and showed that our models can effectively preserve salient source relations in summaries. Results on benchmark datasets showed that the structural models are on-par with or surpass state-of-the-art published systems.

References

Miguel B. Almeida and Andre F. T. Martins. 2013. Fast and robust compressive summarization with dual decomposition and multi-task learning. In *Proceedings of ACL*.

Taylor Berg-Kirkpatrick, Dan Gillick, and Dan Klein. 2011. Jointly learning to extract and compress. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

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

Giuseppe Carenini and Jackie Chi Kit Cheung. 2008. Extractive vs. NLG-based abstractive summarization of evaluative text: The effect of corpus controversiality. In *Proceedings of the Fifth International Natural Language Generation Conference (INLG)*.

Danqi Chen and Christopher D. Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, and Hui Jiang. 2016. Distraction-based neural networks for document summarization. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI)*.

Huadong Chen, Shujian Huang, David Chiang, and Jiajun Chen. 2017. Improved neural machine translation with a syntax-aware encoder and decoder. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. In *Proceedings of ACL*.

Sumit Chopra, Michael Auli, and Alexander M. Rush. 2016. Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of NAACL*.

James Clarke and Mirella Lapata. 2008. Global inference for sentence compression: An integer linear programming approach. *Journal of Artificial Intelligence Research*.

Hal Daume III and Daniel Marcu. 2002. A noisy-channel model for document compression. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Greg Durrett, Taylor Berg-Kirkpatrick, and Dan Klein. 2016. Learning-based single-document summarization with compression and anaphoricity constraints. In *Proceedings of the Association for Computational Linguistics (ACL)*.

Giuseppe Di Fabbrizio, Amanda J. Stent, and Robert Gaizauskas. 2014. A hybrid approach to multi-document summarization of opinions in reviews. *Proceedings of the 8th International Natural Language Generation Conference (INLG)*.

Katja Filippova, Enrique Alfonseca, Carlos Colmenares, Lukasz Kaiser, and Oriol Vinyals. 2015. Sentence compression by deletion with lstms. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Dimitrios Galanis and Ion Androutsopoulos. 2010. An extractive supervised two-stage method for sentence compression. In *Proceedings of NAACL-HLT*.

Shima Gerani, Yashar Mehdad, Giuseppe Carenini, Raymond T. Ng, and Bita Nejat. 2014. Abstractive summarization of product reviews using discourse structure. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of ACL*.

Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. 2016. Pointing the unknown words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Sepp Hochreiter and Jurgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Hongyan Jing and Kathleen McKeown. 1999. The decomposition of human-written summary sentences. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*.

Yuta Kikuchi, Graham Neubig, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. 2016. Controlling output length in neural encoder-decoders. In *Proceedings of EMNLP*.

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

Chen Li, Fei Liu, Fuliang Weng, and Yang Liu. 2013. Document summarization via guided sentence compression. In *Proceedings of EMNLP*.

Chen Li, Yang Liu, Fei Liu, Lin Zhao, and Fuliang Weng. 2014. Improving multi-documents summarization by sentence compression based on expanded constituent parse tree. In *Proceedings of EMNLP*.

Junhui Li, Deyi Xiong, Zhaopeng Tu, Muhua Zhu, Min Zhang, and Guodong Zhou. 2017a. Modeling source syntax for neural machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Piji Li, Wai Lam, Lidong Bing, and Zihao Wang. 2017b. Deep recurrent generative decoder for abstractive text summarization. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Chin-Yew Lin. 2004. ROUGE: a package for automatic evaluation of summaries. In *Proceedings of ACL Workshop on Text Summarization Branches Out*.

Fei Liu, Jeffrey Flanigan, Sam Thomson, Norman Sadeh, and Noah A. Smith. 2015. Toward abstractive summarization using semantic representations. In *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015a. Effective approaches to attention-based neural machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Minh-Thang Luong, Ilya Sutskever, Quoc V. Le, Oriol Vinyals, and Wojciech Zaremba. 2015b. Addressing the rare word problem in neural machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Ryan McDonald. 2006. Discriminative sentence compression with soft syntactic evidence. In *Proceedings of EACL*.

Yishu Miao and Phil Blunsom. 2016. Language as a latent variable: Discrete generative models for sentence compression. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Caglar Gulcehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning (CoNLL)*.

Ani Nenkova and Kathleen McKeown. 2011. Automatic summarization. *Foundations and Trends in Information Retrieval*.

Robert Parker. 2011. English Gigaword fifth edition LDC2011T07. *Philadelphia: Linguistic Data Consortium*.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Ramakanth Pasunuru, Han Guo, and Mohit Bansal. 2017. Towards improving abstractive summarization via entailment generation. In *Proceedings of the Workshop on New Frontiers in Summarization*.

Romain Paulus, Caiming Xiong, and Richard Socher. 2017. A deep reinforced model for abstractive summarization. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the Conference Empirical Methods in Natural Language Processing (EMNLP)*.

Daniele Pighin, Marco Cornolti, Enrique Alfonseca, and Katja Filippova. 2014. Modelling events through memory-based, open-ie patterns for abstractive summarization. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for sentence summarization. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Sho Takase, Jun Suzuki, Naoaki Okazaki, Tsutomu Hirao, and Masaaki Nagata. 2016. Neural headline generation on abstract meaning representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. 2017. Abstractive document summarization with a graph-based attentional neural model. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Kapil Thadani and Kathleen McKeown. 2013. Sentence compression with joint structural inference. In *Proceedings of CoNLL*.

Lu Wang, Hema Raghavan, Vittorio Castelli, Radu Florian, and Claire Cardie. 2013. A sentence compression based framework to query-focused multi-document summarization. In *Proceedings of ACL*.

David Zajic, Bonnie J. Dorr, Jimmy Lin, and Richard Schwartz. 2007. Multi-candidate reduction: Sentence compression as a tool for document summarization tasks. *Information Processing and Management*.

Wenyuan Zeng, Wenjie Luo, Sanja Fidler, and Raquel Urtasun. 2017. Efficient summarization with read-again and copy mechanism. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Xingxing Zhang and Mirella Lapata. 2017. Sentence simplification with deep reinforcement learning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Qingyu Zhou, Nan Yang, Furu Wei, and Ming Zhou. 2017. Selective encoding for abstractive sentence summarization. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.