applied sciences

MDPI

Article

Abstractive Sentence Compression with Event Attention

Su Jeong Choi [1], Ian Jung [1], Seyoung Park [1] and Seong-Bae Park [2,*]

[1] School of Computer Science and Engineering, Kyungpook National University, 80 Daehak-ro, Daegu 41566, Korea; sjchoi@sejong.knu.ac.kr (S.J.C.); ianjung@sejong.knu.ac.kr (I.J.); seyoung@knu.ac.kr (S.P.)

[2] Department of Computer Science and Engineering, Kyung Hee University, 1732 Deogyeong-daero, Gyeonggi-do 17104, Korea

* Correspondence: sbpark71@khu.ac.kr

Received: 8 August 2019; Accepted: 17 September 2019; Published: 20 September 2019

Abstract: Sentence compression aims at generating a shorter sentence from a long and complex source sentence while preserving the important content of the source sentence. Since it provides enhanced comprehensibility and readability to readers, sentence compression is required for summarizing news articles in which event words play a key role in delivering the meaning of the source sentence. Therefore, this paper proposes an abstractive sentence compression with event attention. In compressing a sentence of news articles, event words should be preserved as important information for sentence compression. For this, event attention is proposed which focuses on the event words of the source sentence in generating a compressed sentence. The global information in the source sentence is as significant as event words, since it captures the information of a whole source sentence. As a result, the proposed model generates a compressed sentence by combining both attentions. According to experimental results, the proposed model outperforms both the normal sequence-to-sequence model and the pointer generator on three datasets, namely the MSR dataset, Filippova dataset, and Korean sentence compression dataset. In particular, it shows 122% higher BLEU score than the sequence-to-sequence model. Therefore, the proposed model is effective in sentence compression.

Keywords: sentence compression; event attention; global attention; sequence-to-sequence; pointer generator

1. Introduction

Sentence compression is an NLP task which aims at generating a compact sentence from a long and complex source sentence while preserving the important content of the source sentence. Since it generates a shorter and more condensed sentence than its source sentence, it is also known as sentence-level summarization. The need to sentence compression is increasing with the rapid growth of web and mobile contents, since sentence compression allows web and mobile users to efficiently catch and understand the contents. That is, sentence compression does not only enhance the comprehensibility and readability of the contents for the users, but also saves time and cost [1,2]. Thus, sentence compression is regarded as a significant and useful technique.

There have been many studies about sentence compression and the studies are separated into two approaches: a deletion-based approach and an abstractive approach. The deletion-based approach generates a target sentence by removing unimportant and unnecessary words from the source sentence [3–8]. That is, a target compressed sentence is a subsequence of a source sentence. Filippova et al. solved sentence compression as a binary classification task with a LSTM-based model [9]. In this model, LSTM(Long Short-term Memory) determines whether each word in a source sentence

should be removed or not. This approach is straightforward, but dissimilar to sentence compression of human beings in that the target sentence by human beings often contains some new words that do not appear in the source sentence. In addition, the approach often produces a broken sentence by a few grammatical errors. This phenomenon gets severer in morphologically complex languages such as Korean and Japanese. Therefore, the approach is not appropriate for expressive sentence compression.

Some abstractive compression methods have been proposed to solve the problem [10–14]. With the growth of neural networks in recent years, Yu et al. proposed an abstractive compression model with sequence-to-sequence learning and an attention mechanism [15]. They used two kinds of decoders. One is a deletion decoder and the other is a copy-generation decoder. When the encoded vector of a source sentence is decoded, it is first passed to the deletion decoder. After that, the vector is incorporated with the result of deletion decoder, and the incorporated vector is fed to the copy-generation decoder. Although this is somewhat similar to human compression, it has some problems. First, the errors become cumulative, since the copy-generation decoder generates a target sentence based on the result of the deletion decoder. Thus, if the deletion decoder produces some errors, it is difficult for the copy-generation decoder to generate a correct target sentence. In addition, its attention mechanism often fails to focus on important content of a source sentence. In particular, when a source sentence is long, the attention failure gets severer [8,16,17].

This paper proposes an abstractive sentence compression model with an event attention. This model is based on the sequence-to-sequence model [16,18] and the pointer generator [19] to generate an expressive target sentence, and adopts an event attention to focus on important eventual content in the source sentence. One of the major applications of sentence compression is to summarize news articles in which event words play a key role. Therefore, the event attention regards event words as influential words for sentence compression and focuses on them in decoding the vector of a source sentence. Even though event words provide key information for knowing what is going on in a source sentence, it is also important to understand a source sentence globally. Thus, the global attention [20] is also used to capture the global information of a source sentence. Then, the final attention is represented by combining the event attention and the global attention. Three datasets are used to evaluate the proposed model which are MSR dataset [21], Filippova dataset [9], and Korean sentence compression dataset. As evaluation metrics, ROUGE [22], BLEU [23] and compression ratio [24] are adopted. According to the experimental results, the ROUGE-L of the proposed model is 122.1% higher than a standard sequence-to-sequence model on MSR dataset. Even in compression ratio, the proposed model outperforms both the sequence-to-sequence and point generator. These results prove that event words are important information for sentence compression and the proposed model compresses a source sentence effectively by focusing on event words.

The rest of this paper is organized as follows. Section 2 describes related work on sentence compression and summarization, and Section 3 explains the sequence-to-sequence model for sentence compression. Section 4 proposes an abstractive sentence compression model with event attention. Section 5 shows the experimental results and results analysis. Finally, Section 6 draws our conclusions.

2. Related Work

There have been several studies on sentence compression and the studies are separated into two kinds of approaches: the deletion-based approach and the abstractive approach. Sentence compression by deletion-based approach is to decide whether each word in a source sentence remains in a target sentence. That is, it regards sentence compression as a binary sequence labeling problem. Linguistic information is usually used to determine the sequence labels [25–27], but some previous studies compressed a sentence by pruning the dependency parse tree of a source sentence [7,10]. Filippova et al. proposed an unsupervised compression method to prune the dependency parse tree [28]. In their method, the weights of the edges in a dependency parse tree are computed using syntactic and length constraints, and then they are used to prune the dependency tree. After that, they proposed a supervised version which extends the unsupervised method [6]. For supervised learning of sentence

compression, they also released a dataset for sentence compression. In this version, the weight of every edge in a parse tree is computed by a linear function of lexical, syntactic, and semantic features, and the relevance weights of the features are determined using the released data. All these models depend greatly on the accuracy of parsing a source sentence, but current techniques for parsing natural language sentences are not fully trustworthy yet.

With the growth of neural networks, some studies have adopted a neural network to determine word drops. Filippova et al. claimed that syntactic information could be unnecessary for sentence compression [9]. Thus, they adopted a LSTM model to determine if a word should be deleted in the compressed sentence based on the deletion/retention of its previous word in a source sentence. In addition, Hasegawa et al. used a LSTM constrained with a target length [29]. These methods do not use any syntactic information, but some experiments showed that the methods could be improved if syntactic information is incorporated into the methods [7].

One of main problems using neural networks for sentence compression is that named entities in a source sentence are usually considered to be unknown words since the neural models use a pretrained word embedding, where high frequency of unknown words often leads to performance decrease of the models. Wang et al. hypothesized that the incorporation of syntactic information into a compression model would be helpful in sentence compression and domain adaptability [7]. Thus, they proposed a LSTM-based model which uses syntactic information such as POS tags and parsing information. They showed empirically that syntactic information is influential and leverages the robustness of a model in cross-domain applications. However, the deletion-based approach has the chronic limitation of being unable to generate expressive sentences.

The abstractive approach to sentence compression can be regarded as sentence paraphrasing. Thus, it can generate some unseen words and deliver more expressiveness in the target compressed sentence. That is, it produces more expressive compressed sentences than the deletion-based approach. Inspired by neural machine translation, Rush et al. applied an attention-based sequence-to-sequence model to sentence compression [11]. Since a sequence-to-sequence model is structurally simple, their model has the merit that it does not require any pre- and post-processing for sentence compression. On the other hand, Vu et al. adopted a memory-augmented recurrent neural network [30]. The memory-augmented model uses a memory matrix to get a better understanding of a source sentence. However, these studies are vulnerable to named entities since they generate a sentence with a fixed vocabulary set. To overcome this problem, Yu et al. proposed an operation network that mimics human sentence compression [15]. Since this model is capable of copying, editing, and generating words, it can imitate human compression as well as overcome limits of the deletion-based approach. However, due to the nature of sequence-to-sequence models, these studies have difficulty compressing long and complex sentences.

Some novel attention mechanisms have been proposed to attack the difficulty. Chopra et al. proposed a convolutional attention-based model [31] in which the attention helps to understand a source sentence by capturing its context. On the other hand, Kamigatio et al. incorporated a higher-order syntactic attention into a sequence-to-sequence model [8]. The attention is computed with a chain of dependency relations in the dependency graph of a source sentence. The problem of the studies is that the attention is somewhat helpful in understanding a sentence, but fails to focus on the words that deliver important information such as an event in a long and complex sentence. Therefore, this paper proposes an event attention which enables the network to regard event words as salient words.

3. Sentence Compression by a Sequence-to-Sequence Model

Given a source sentence $X = (x_1, x_2, x_3, ..., x_n)$, sentence compression aims to generate a target sentence $Y = (y_1, y_2, y_3, ..., y_m)$, where n and m are sentence lengths and $n > m$. Thus, a sentence compression model can be formulated by a conditional probability

$$P(Y|X) = \prod_{t=1}^{m} P(y_t|y_{<t}, X). \quad (1)$$

Since optimizing Equation (1) is an objective of standard sequence-to-sequence models [16,18], a sequence-to-sequence model can be used as a sentence compression model. A sequence-to-sequence model has an encoder and a decoder. The encoder which is usually implemented as a bi-directional RNN [32] takes the source sentence $\mathbf{X} = (\mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3}, ..., \mathbf{x_n})$ where $\mathbf{x_i}$ is an embedding vector of the word x_i. At each time step t, the encoder outputs a forward hidden state $\overrightarrow{h_t} = f(\mathbf{x_t}, \overrightarrow{h_{t-1}})$ and a backward hidden state $\overleftarrow{h_t} = f(\mathbf{x_t}, \overleftarrow{h_{t-1}})$, where the function f is a recurrent unit such as GRU [16] or LSTM [33]. The hidden state h_t at time t is expressed as a concatenation of $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$. That is,

$$h_t = [\overrightarrow{h_t}; \overleftarrow{h_t}],$$

where $[a; b]$ denotes the concatenation of a vector a and a vector b. The concatenation of the final forward and backward states of the encoder

$$Q = [\overrightarrow{h_{final}}; \overleftarrow{h_{final}}] \quad (2)$$

is used as a representation of the source sentence $\mathbf{X}$.

The decoder which is also implemented as an RNN generates a target compressed sentence Y with vectors computed by the encoder. It computes a decoder state s_t at each time t as

$$s_t = f(y_{t-1}, s_{t-1}, Q). \quad (3)$$

Then, P_{vocab}, the probabilistic distribution over all words, is obtained by

$$P_{vocab} = softmax(W_v s_t + b_v), \quad (4)$$

where W_v and b_v are the parameters to be tuned. That is, the decoder state s_t is fed to the softmax function to produce P_{vocab}. Finally, the word with the highest probability in P_{vocab} is chosen as the output word at time t. Since the sequence-to-sequence model is trained with a set of pairs of a long source sentence and its short compressed sentence, it eventually learns how to transform a long sentence to a short sentence, which is sentence compression.

4. Abstractive Sentence Compression with Event Attention

Event words in news articles play a key role in delivering the meaning of a source sentence, since an event means something that happens or occurs [34]. For instance, Table 1 shows compressed sentences of a source sentence "*If IBM has* ***miscalculated*** *the demand, it will* ***suffer*** *badly as both the high operating costs and depreciation on the huge capital investment for the East Fishkill factory* ***drag*** *down earnings.*" in which the event words are expressed as bold. Since the event words contain main actions of the source sentence, they are usually preserved in the compressed sentence. As a result, the compressed sentences in Table 1 contain at least one of the words '*miscalculated*', '*suffer*', and '*drag*'. Therefore, the event words must be regarded as salient words in generating a compressed sentence.

To focus on the event words of a source sentence in generating a compressed sentence, the event words should be first extracted. In this paper, the event words are extracted using the event extraction system proposed by Chambers et al. [35]. When a source sentence $X = (x_1, x_2, x_3, ..., x_n)$ is given, the system returns an event mask vector $E_X = (e_1, e_2, e_3, ..., e_n)$ where e_i is 1 if x_i is an event word and 0 otherwise. After that, a compressed sentence of X is generated by a sequence-to-sequence model. Figure 1 depicts the proposed sequence-to-sequence model to generate a compressed sentence from a vectorized source sentence $\mathbf{X}$ and the event mask vector E_X. For a given source sentence $\mathbf{X}$, the encoder outputs the hidden state h_i's and the source sentence representation Q by Equation (2).

Table 1. An example of a source sentence and its compressed sentences in MSR dataset.

Source sentence	If IBM has **miscalculated** the demand, it will **suffer** badly as both the high operating costs and depreciation on the huge capital investment for the East Fishkill factory **drag** down earnings.
Compressed sentences	· If IBM has miscalculated the demand, high operating costs and depreciation will drag down earnings. · Miscalculation will lead IBM to have both high operating costs and capital investment depreciation for the East Fishkill earnings. · If IBM miscalculated demand, it will suffer as high operating costs and depreciation on capital investment for the factory lower earnings. · IBM will suffer if it miscalculates the demand as the operating costs and depreciation on investment for East Fishkill drag down earnings. · If IBM has miscalculated the demand, it will suffer badly as both the high operating costs and depreciation on the huge capital.

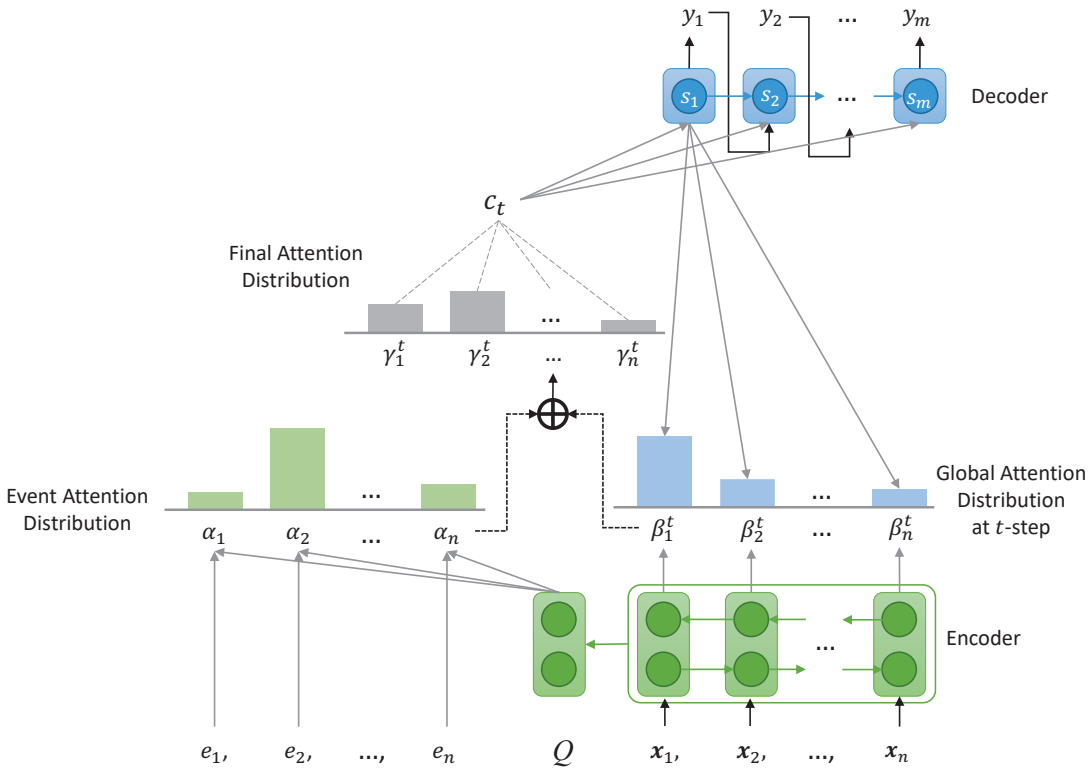

Figure 1. The overall process of abstractive sentence compression with event attention.

To compress the source sentence elaborately, two types of attention, which are *event attention* and *global attention*, are adopted in the proposed model. The event attention allows the model to focus on the event words in the source sentence, while the global attention provides the overall understanding of the source sentence. The event attention weight α_i of an encoder hidden state h_i is computed using h_i, the source sentence representation Q, and the event mask e_i. That is,

$$\begin{aligned} \alpha_i &= \frac{exp(u_i)}{\sum_{j=1}^{n} exp(u_j)}, \\ u_i &= V_e^{\top} g(W_h h_i + W_q Q + W_e e_i + b_{eattn}), \end{aligned} \tag{5}$$

where V_e, W_h, W_q and W_e are weight parameters and learned from training data. b_{eattn} is a bias vector, and g is a non-linear function.

Since the event attention helps the decoder focus on event words during decoding, it pays less attention to the salient words which deliver the overall meaning of the source sentence. Thus, an attention proposed by Bahdanau et al. [20] is adopted as a global attention of the proposed model. The reason the attention is adopted is that it is one of the widely used and standard attention [36]. When the decoder generates the t-th word, the global attention weight β_i^t of h_i is computed by

$$\begin{aligned} \beta_i^t &= \frac{exp(d_i^t)}{\sum_{j=1}^{n} exp(d_i^j)}, \\ d_i^t &= V_g^\top g(W_{h'} h_i + W_s s_{t-1} + b_{gattn}), \end{aligned} \tag{6}$$

where V_g, $W_{h'}$ and W_s are weight parameters and learned from training data, b_{gattn} is a bias vector for the global attention, and g is a non-linear function. The two attention weights are then combined by a weighted sum. That is, the final attention weight γ_i^t of a hidden state h_i at the t-th word generation is

$$\gamma_i^t = (1 - \lambda)\alpha_i + \lambda\beta_i^t, \tag{7}$$

where λ ($0 \leq \lambda \leq 1$) is a hyper-parameter to control the ratio of the event attention and the global attention.

The final attention weights, γ_i^t's, are used to compute the context vector c_t as

$$c_t = \sum_{i=1}^{n} \gamma_i^t h_i,$$

and the context vector is used in generating y_t, the t-th word by the decoder. Since the decoder depends on c_t as well as s_t in generating y_t, Equation (4) is rewritten as

$$P_{vocab} = softmax(W_v[s_t; c_t] + b_v).$$

Then, the probability of choosing a word w from a vocabulary set V is $P(w) = P_{vocab}(w)$. The proposed model is trained to minimize the risk $\mathcal{R}$ over training set T defined as

$$\mathcal{R}(\theta) = \sum_{(X,Y)\in T} -\log P(Y|X;\theta),$$

where θ is the set of all parameters of the proposed model.

Since the model above is based on a sequence-to-sequence model, it suffers from the chronic out-of-vocabulary problem [37]. To solve the problem, the model is extended to host the copy technique of the pointer generator [19]. The copy technique deals with the out-of-vocabulary problem by allowing the model to copy unknown words of a source sentence into the target sentence. That is, after expanding the vocabulary set $V' = V \cup U$ where U is a set of unknown words in the source sentence, $P(w \in V')$ is updated as

$$P(w) = P_{gen} \cdot P_{vocab}(w) + (1 - P_{gen}) \sum_{i:w_i=w} \gamma_i^t.$$

by introducing a soft switch P_{gen} which either generates a word from P_{vocab} or copies a word from the source sentence. If P_{gen} is higher than $1 - p_{gen}$, the word w is generated from P_{vocab}. Otherwise, it is copied from the source sentence where the word to be copied is determined by the final attention γ_i^t. The switch P_{gen} can be expressed as a generation probability computed by a network with two linear layers. Thus, P_{gen} is computed using the context vector c_t, the decoder state s_t, and the previous decoder output y_{t-1} as

$$P_{gen} = \sigma(W_{c'}(W_c[s_t; c_t] + b_c) + b_{c'}),$$

where W_c, $W_{c'}$, b_c and $b_{c'}$ are the learnable weight parameters of the network and σ is the sigmoid function.

5. Experiments

5.1. Experimental Setting

The proposed model is evaluated with three kinds of datasets: MSR dataset [21] (see Supplementary Materials), Filippova dataset [9] (see Supplementary Materials), and Korean sentence compression dataset. The simple statistics on each dataset is given in Table 2. The MSR dataset proposed by Toutanova et al. is used for abstractive sentence compression. It consists of a newswires, business letters, journals, and technical documents from the Open American National Corpus, and the compressed sentences are created manually. In this dataset, the numbers of training, validation, and test examples are 21,145, 1908, and 3370 pairs, respectively. The average length of source sentences is 193.2, while that of target sentences is 133.6. Filippova dataset [9] is used for deletion-based sentence compression. This dataset is automatically produced from Google News using the method proposed by Filippova et al [6]. The total number of sentence pairs is 10,000, and target sentences have 13.7 fewer words than source sentences on average. The last Korean sentence compression dataset is designed for testing morphologically complex languages. This dataset contains 3117 source sentences from Korean news articles, and the sentences are split to 8:1:1 for training, validation, and test sets, respectively. Its target sentences are created by a native speaker and their length is short on average by 6.01 words when compared to the source sentences.

Table 2. Simple statistics on the datasets used.

Dataset	MSR	Filippova	Korean
No. of training examples	21,145	8000	2493
No. of validation examples	1908	1000	312
No. of test examples	3370	1000	312
Average length of source sentences	31.85	23.04	14.12
Average length of target sentences	21.97	9.34	8.11

For the experiments below, the dimension of word embeddings is set as 256, and that of all hidden layers is set as 512. LSTM [33] is used for the function f in Equation (3) and the hyperbolic tangent is used for the function g in Equation (5) and (6). The proposed network is trained with batch size 64 and learning rate 0.001, and is optimized with an Adam optimizer [38]. The value of λ in Equation (7) is 0.6 which is estimated using MSR validation set. BLEU [23], ROUGE [22], and compression ratio (CR) [24] are used as evaluation metrics, where **CR** is computed as

$$\mathbf{CR} = \frac{1}{u}\sum_{i=1}^{u}\frac{\text{No. of tokens in the } i\text{-th compressed sentence}}{\text{No. of tokens in the } i\text{-th source sentence}},$$

and u is the number of sentences in a test set. Since the test set consists of pairs of a source sentence and its target sentence, the golden compression ratio can be computed from the test set. The golden compression ratio on MSR dataset is 68.60%, while those on Filippova dataset and Korean dataset are 42.62% and 47.52%, respectively. Two baselines are adopted for comparing the proposed model with existing models. One baseline is the standard sequence-to-sequence model proposed by Cho et al. [16], and the other is the pointer generator [19] in which the copy technique is applied to the sequence-to-sequence model.

5.2. Experimental Results

Table 3 shows the sentence compression performance on MSR dataset. In this table, 'seq-to-seq' and 'PG' denote the sequence-to-sequence model and the pointer generator, respectively. The model of Yu et al. is based on the sequence-to-sequence with a deletion decoder and a copy-generator

decoder [15]. Unlike the proposed model, it first conducts the deletion of a source sentence using the deletion decoder and then either generates words or copies the source sentence through the copy-generator decoder. According to the table, the use of event attention improves its base model. That is, 'seq-to-seq + Event' shows higher performance than 'seq-to-seq' in ROUGE metrics, and 'PG + Event' outperforms 'PG' in all metrics. In particular, 'PG + Event' achieves the best performance. On the other hand, the performance of Yu's model is lower than 'PG + Event', even if it is also based on the copy mechanism. This is because the errors by the deletion decoder are easy to propagate to the copy-generation decoder in Yu's model.

Table 3. Experimental results on MSR dataset.

	BLEU	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**	**CR**(%)
seq-to-seq	18.32	30.05	10.42	26.87	69.39 (+0.79)
seq-to-seq + Event	18.04	45.70	25.94	41.63	59.03 (−9.57)
Yu et al.	26.30	36.21	17.43	33.72	65.53 (−3.07)
PG	31.70	61.35	41.91	56.66	66.46 (−2.14)
PG + Event	**34.41**	**63.25**	**43.58**	**59.69**	**67.97** (−0.63)

The compression ratios denoted as **CR** are also given in this table to see how much a method compresses source sentences. The value within parentheses denotes the difference between the compression ratio of a method and the golden ratio. As a result, the smaller the absolute value of the difference is, the better a method compresses source sentences. The golden compression ratio on MSR dataset is 68.60%, and the difference between it and the compression of ratio of 'PG + Event' is smallest as −0.63. This result implies that the proposed model is an effective compressor as well as a good writer.

Table 4 shows the results of sentence compression on Filippova dataset. When the event attention is applied to a baseline model, the performance of the baseline model improves. That is, 'seq-to-seq + Event' is better than 'seq-to-seq' and 'PG + Event' outperforms 'PG' for all evaluation metrics. The proposed model works effectively even with the dataset for deletion-based compression. However, in compression ratio, the proposed model is only slightly worse than 'PG'. The golden compression ratio of Filippova dataset is 42.62%, but the compression ratio of 'PG' is 42.45% while that of 'PG + Event' is 41.70%. The main reason 'PG' is better than 'PG + Event' in compression ratio is that this dataset is designed for deletion-based compression.

Table 4. Experimental results on Filippova dataset.

	BLEU	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**	**CR** (%)
seq-to-seq	25.62	53.14	36.12	50.59	38.61 (−4.01)
seq-to-seq + Event	28.70	56.03	39.77	53.51	38.72 (−3.90)
PG	42.50	68.60	57.06	65.83	**42.45** (−0.17)
PG + Event	**46.28**	**72.54**	**58.97**	**68.94**	41.70 (−0.92)

To verify that the proposed model works for morphologically complex languages, we conducted sentence compression on Korean dataset. Table 5 shows the result on Korean sentence compression. Overall, the proposed model outperforms all baselines. That is, 'PG + Event' shows the best performance for all metrics. One thing to note is that the performance difference between 'PG + Event' and 'seq-to-seq + Event' is large. This is because Korean dataset is relatively small, but its vocabulary size is large. Under this circumstance, the copy mechanism is much helpful in solving the out-of-vocabulary problem. In compression ratio, both 'PG' and 'PG + Event' are relatively closer to the golden compression ratio of Korean dataset. Please note that the overall performance of the proposed model for this dataset is lower than those for other datasets. This is because it is difficult

to generate good Korean sentences with a small dataset since Korean is a morphologically complex language. However, even for this dataset, the proposed model outperforms its competitors.

Table 5. Experimental results on Korean dataset.

	BLEU	ROUGE-1	ROUGE-2	ROUGE-L	CR(%)
seq-to-seq	6.28	26.03	9.44	23.06	38.06 (−9.46)
seq-to-seq + Event	12.38	36.16	18.32	33.08	56.09 (+8.57)
PG	29.30	64.99	42.38	61.31	**45.26** (−2.26)
PG + Event	**31.52**	**66.09**	**44.62**	**62.72**	44.95 (−2.57)

Table 6 presents some examples of compressed sentences by 'PG' and 'PG + Event'. Sentence 1 and 2 in this table are from MSR dataset, while Sentence 3 comes from Filippova dataset. Bold words indicate event words in all source sentences. The important phrases of Sentence 1 are '*80% of youth will report increased supervised time in safe environments*' and '*80% of participants will report increased conflict resolution skills*', and the phrases contain the event word '*report increased*'. The compressed sentence by 'PG' is "*Anticipated outcomes from the spring survey include supervised increased conflict % of participants will report increased conflict resolution.*" which misses the important information such as '*supervised time in safe environments*' and thus delivers distorted meaning from the source sentence. On the other hand, the compressed sentence by the proposed 'PG + Event' delivers more precise meaning than that by 'PG', since it contains '*80% of youth will report in safe environments*' and '*they will report increased conflict resolution skills*.'

Table 6. Some examples of compressed sentences by compression models.

	Sentence 1	Sentence 2	Sentence 3
Source	Anticipated outcomes from the spring survey include: 80% of youth will **report increased** supervised time in safe environments. 80% of participants will **report increased** conflict resolution skills.	Support is **needed** both to **maintain** and **expand** these comprehensive programs. Please **help** the American Cancer Society **continue** its vital work.	why are homeowners **reporting** that their glass door suddenly **shattered**?
Target	Expected outcomes from survey: 80% of youth will report more time in safe places. 80% of people will report greater conflict resolution skills.	Maintenance and expansion of our programs needs support. Help the American Cancer Society continue.	reporting their glass door suddenly shattered
PG	Anticipated outcomes from the spring survey include supervised increased conflict % of participants will report increased conflict resolution.	nature much to maintain these comprehensive programs. The American Cancer Society's vital work.	why are homeowners glass door strong
PG + Event	Anticipated outcomes from the spring survey include: 80% of youth will report in safe environments. They will report increased conflict resolution skills.	Support the American Cancer Society continue its vital work and help to maintain.	reporting their glass door shattered

To verify the attention weights when decoding, the weights for Sentence 1 are shown in Figure 2. The figure shows the source sentence in the horizontal axis and the generated sentence in the vertical axis. The darker the color of each word cell is, the heavier the attention weight of the word is. The upper figure shows the compressed sentence by 'PG' and the bottom one is by 'PG + Event'. The bold tokens are those recognized as event words. In the attention by 'PG', after the word '*include*', attention should have given to '*youth*' and '*increased supervised time*' or '*in the safe environments*', but is given to a wrong word '*conflict*'. As a result, 'PG' generates '*participants will report increased conflict resolution*'. In the bottom figure, the attention weights by'PG + Event' are relatively ideal. The model is attentive to all event words, and the words are generated in the compressed sentence. In addition, globally important phrases such as '*youth*', '*safe environments*', and '*conflict resolution skills*' get attentive, and then the

compressed sentence is generated semantically correctly. Since the proposed model is designed to consider both event words and global information, it can generate an effective compressed sentence.

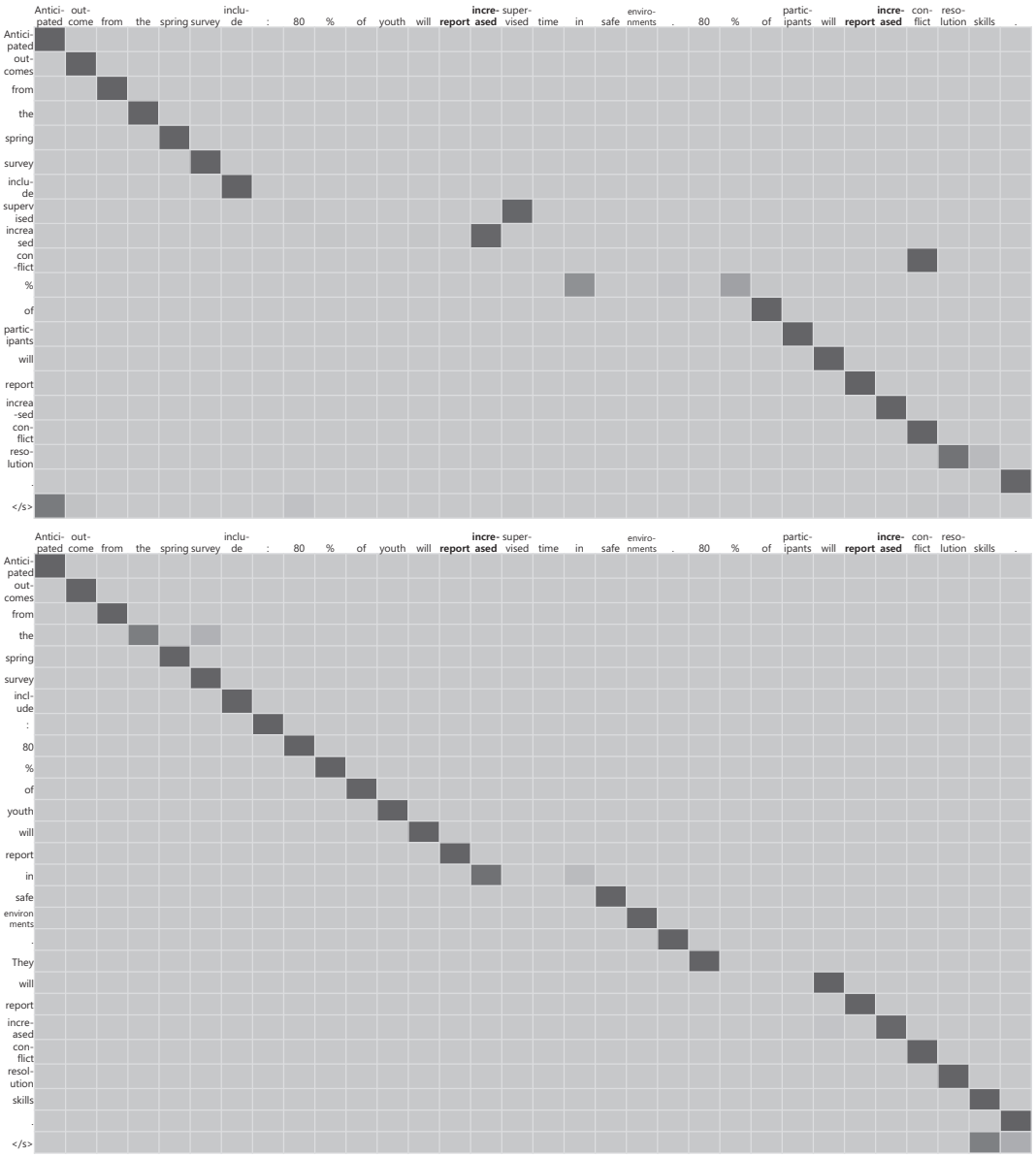

Figure 2. The attention weights on Sentence 1 in Table 6.

Sentence 2 also shows the superiority of the proposed model. The important phrases of the source sentence is '*support is needed to maintain and expand*' and '*help the American Cancer Society continue*'. As a compressed sentence, 'PG' generates '*nature much to maintain these comprehensive programs. The American Cancer Society's vital work*' which is wrong semantically and grammatically. On the other hand, the compressed sentence by'PG + Event' is '*Support the American Cancer Society continue its vital work and help to maintain*.' This sentence contains the event word '*support*' and delivers the meaning of source sentence correctly. Lastly, Sentence 3 is sampled from Filippova dataset and is shorter than other examples. The key information of the source sentence is '*report that their glass door shattered.*' 'PG + Event' generates '*reporting their glass door shattered*' in which salient words such as '*report*' and '*shatter*' are involved. As a result, one can find out '*glass door shatters*' from the compressed sentence, but not from the compressed sentence by 'PG'. The sentence by 'PG' is grammatically incomplete and the fact related to '*homeowners reporting*' is not found. Figure 3 shows attention weights for the source sentence

of Sentence 3. The top figure is the weights by 'PG' and the bottom is those by 'PG + Event'. 'PG' does not regard '*reporting*' as a salient word, and thus it does not generate the word in the compressed sentence. In addition, even if '*shattered*' is attentive, it generates '*strong*' instead of it. On the other hand, 'PG + Event' pays attention to '*reporting*' and '*shatter*' as important information in terms of event. In addition, the globally salient phrase '*their glass door*' is also focused by 'PG + Event'. As a result, 'PG + Event' is able to generate a compressed sentence which is semantically and grammatically correct. These examples show that it is effective for sentence compression to use event attention as well as global attention.

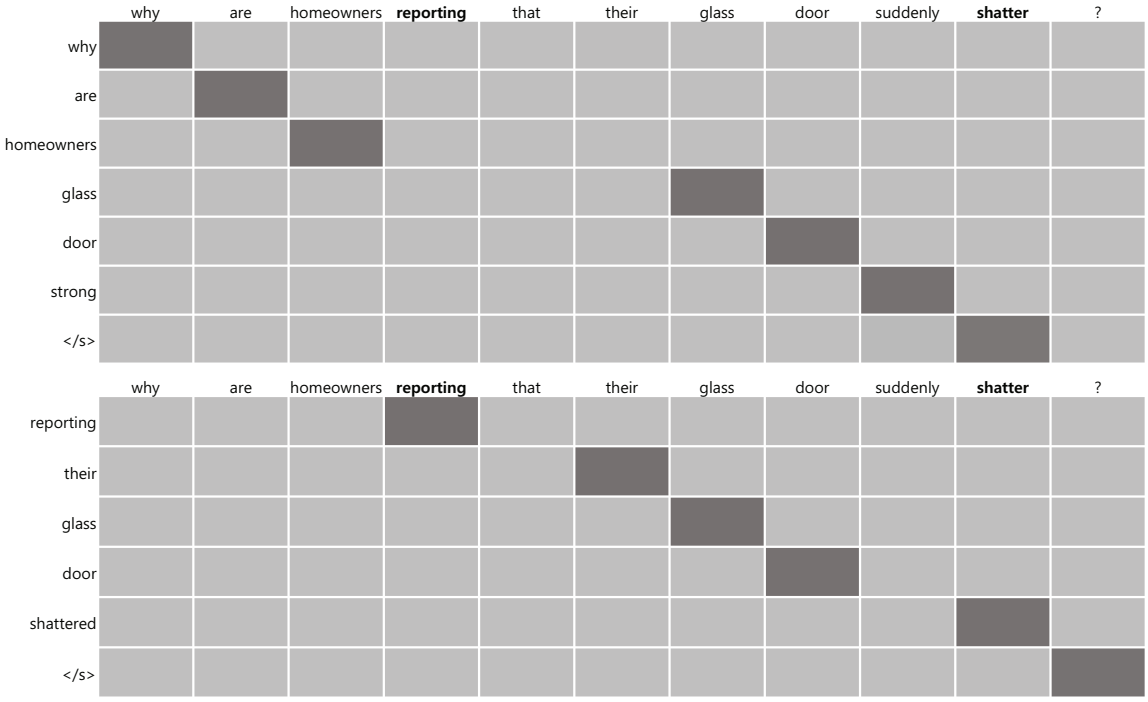

Figure 3. The attention weights on Sentence 3 in Table 6.

6. Conclusions

We have proposed an abstractive sentence compression model with event attention. Sentence compression is the task of generating a compact sentence from a source sentence while preserving the important content of the source sentence. However, existing models for sentence compression have a limitation that their attention often fails to focus on important context of a source sentence especially when the source sentence is long and complex. In this paper, we handled the problem with event attention. The proposed event attention focuses on event words since event words are important information for sentence compression and deliver the meaning of source sentences. In addition to event attention, the global attention was also used which helps to understand source sentences because it captures global information of the source sentence. Therefore, the proposed model compresses source sentences by combining event and global attention. For the evaluation of the proposed model, the proposed model has been compared with two baselines on three standard datasets. According to the experimental results, it outperforms all baselines for all datasets of MSR dataset, Filippova dataset, and Korean sentence compression dataset. In particular, it shows 122% higher BLEU score than the sequence-to-sequence model on MSR dataset. This result shows that event words are valuable information and the proposed model is effective for sentence compression. For future work, we will extend the proposed model for multi-sentence compression. The current model compresses just one or two sentences, but real news articles consist of multi-sentences. Thus, multi-sentence compression is required for real applications.

Supplementary Materials: The MSR dataset [21] and Filippova dataset [9] are used to support this study and are available at https://www.microsoft.com/en-us/research/project/intelligent-editing and https://github.com/google-research-datasets/sentence-compression. The Korean sentence compression dataset generated during the current study are available in [KoreanSentenceCompression] repository (https://github.com/daisy-choi/KoreanSentenceCompression).

Author Contributions: Conceptualization, S.J.C. and I.J.; Funding acquisition, S.P.; Investigation, S.J.C. and I.J.; Methodology, S.J.C. and I.J.; Project administration, S.P.; Supervision, S.P. and S.-B.P.; Validation, S.-B.P.; Visualization, S.J.C.; Writing—original draft, S.J.C.; Writing—review & editing, S.-B.P.

Acknowledgments: This study was supported by the BK21 Plus project (SW Human Resource Development Program for Supporting Smart Life) funded by the Ministry of Education, School of Computer Science and Engineering, Kyungpook National University, Korea (21A20131600005).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Clarke, C.; Agichtein, E.; Dumais, S.; White, R. The Influence of Caption Features on Clickthrough Patterns in Web Search. In Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Amsterdam, The Netherlands, 23–27 July 2007; pp. 135–142.
2. Kanungo, T.; Orr, D. Predicting the Readability of Short Web Summaries. In Proceedings of the 2th ACM International Conference on Web Search and Data Mining, Barcelona, Spain, 9–12 February 2009; pp. 202–211.
3. Knight, K.; Marcu, D. Summarization beyond sentence extraction: A probabilistic approach to sentence compression. *Artif. Intell.* **2002**, *139*, 91–107. [CrossRef]
4. McDonald, R. Discriminative Sentence Compression with Soft Syntactic Evidence. In Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics, Trento, Italy, 3–7 April 2006; pp. 297–304.
5. Filippova, K. Multi-Sentence Compression: Finding Shortest Paths in Word Graphs. In Proceedings of the 23th International Conference on Computational Linugustics, Beijing, China, 23–27 August 2010; pp. 322–330.
6. Filippova, K.; Altun, Y. Overcoming the Lack of Parallel Data in Sentence Compression. In Proceedings of the 2013 Conference on Empirical Method in Natural Language Processing, Seattle, WA, USA, 18–21 October 2013; pp. 1481–1491.
7. Wang, L.; Jiang, J.; Chieu, H.; Ong, C.; Song, D.; Liao, L. Can Syntax Help? Improving an LSTM-based Sentence Compression Model for New Domains. In Proceedings of the 55th Annual Meetings of the Association for Computational Linguistics, Vancouver, BC, Canada, 30–4 July 2017; pp. 1385–1393.
8. Kamigaito, H.; Hayashi, K.; Hirao, T.; Nagata, M. Higher-order Syntactic Attention Network for Long Sentence Compression. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, New Orleans, LA, USA, 1–6 June 2018; pp. 1716–1726.
9. Filippova, K.; Alfonseca, E.; Colmenares, C.; Kaiser, L.; Vinyals, O. Sentence Compression by Deletion with LSTMs. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; pp. 360–368.
10. Galanis, D.; Androutsopoulos, I. A New Sentence Compression Dataset and Its Use in an Abstractive Generate-and-Rand Sentence Compressor. In Proceedings of the UCNLG+Eval: Language Generation and Evaluation Workshop, Edinburgh, Scotland, UK, 31 July 2011; pp. 1–11.
11. Rush, A.; Chopra, S.; Weston, J. A Neural Attention Model for Sentence Summarization. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; pp. 379–389.
12. Shafieibavani, E.; Ebrahimi, M.; Wong, R.; Chen, F. An Efficient Approach for Multi-Sentence Compression. In Proceedings of the 8th Asian Conference on Machine Learning, Hamilton, New Zealand, 16–18 November 2016; pp. 414–429.
13. Zhang, X.; Lapata, M. Sentence Simplication with Deep Reinforcement Learning. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Procedding, Copenhagen, Denmark, 7–11 September 2017; pp. 584–594.

14. Mallinson, J.; Sennrich, R.; Lapata, M. Sentence Compression for Arbitrary Language via Multilingual Pivoting. In Proceedings of the 2018 Conference on Empirical Methods in Natural Lanague Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 2453–2464.
15. Yu, N.; Zhang, J.; Huang, M.; Zhu, Z. An Operation Network for Abstractive Sentence Compression. In Proceedings of the 27th International Conference on Computational Linguistics, Santa Fe, NM, USA, 20–26 August 2018; pp. 1065–1076.
16. Cho, K.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In Proceedings of the 2014 Conference on Empirical Method in Natural Language Processing, Doha, Qatar, 25–29 October 2014; pp. 1724–1734.
17. Pouget-Abadie, J.; Bahdanau, D.; van Merriënboer, B.; Bengio, Y. Overcoming the Curse of Sentence Length for Neural Machine Translation using Automatic Segmentation. In Proceedings of the SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, Doha, Qatar, 25 October 2014; pp. 78–85.
18. Sutskever, I.; Vinyals, O.; Le, G. Sequence to Sequence Learning with Neural Networks. In In Proceedings of the Annual Conference on Neural Information Processing Systems 2014, Montreal, QC, Canada, 8–13 December 2014; Advances in Neural Information Processing Systems 27; pp. 3104–3112.
19. See, A.; Liu, P.; Manning, C. Get To The Point: Summarization with Pointer-Generator Networks. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, BC, Canada, 30 July–4 August 2017; pp. 1073–1083.
20. Bahdanau, D.; Cho, K.; Bengio, Y. Neural Machine Tranlation by Jointly Learning to Align and Translate. In Proceedings of the International Conference on Learning Representations, Shanghai, China, 23–26 June 2015.
21. Toutanova, K.; Brockett, C.; Tran, K.; Amershi, S. A Dataset and Evaluation metrics for Abstractive Compression of Sentences and Short Paragraphs. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–5 November 2016; pp. 340–350.
22. Lin, C. ROUGE: A Package for Automatic Evaluation of Summaries. In *Proceedings of Text Summarization Branches Out*; Association for Computational Linguistics: Barcelona, Spain, 2004; pp. 74–81.
23. Papineni, K.; Roukos, S.; Ward, T.; Zhu, W.J. BLEU: A Method for Automatic Evaluation of Machine Translation. In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, Philadelphia, PA, USA, 6–12 July 2002; pp. 311–318.
24. Napoles, C.; Durme, B.; Callison-Burch, C. Evaluating sentence compression: Pitfalls and suggested remedies. In Proceedings of the Workshop on Monolingual Text-To-Text, Portland, OR, USA, 24 June 2011; pp. 91–97.
25. Jing, H. Sentence Reduction for Automatic Text Summarization. In Proceedings of the 6th Conference on Applied Natural Language Processing, Seattle, WA, USA, 29 April–4 May 2000; pp. 310–315.
26. Clarke, J.; Lapata, M. Global Inference for Sentence Compression: An Integer Linear Programming Approach. *J. Artif. Intell. Res.* **2008**, *31*, 399–429. [CrossRef]
27. Fevry, T.; Phang, J. Unsupervised Sentence Compression using Denoising Auto-Encoders. In Proceedings of the 22th Conference on Computational Natural Language Learning, Hong Kong, China, 3–4 November 2018; pp. 413–422.
28. Filippova, K.; Strube, M. Dependency Tree Based Sentence Compression. In Proceedings of the 5th International Natural Language Generation Conference, Tilburg, The Netherlands, 5–8 November 2008; pp. 25–32.
29. Hasegawa, S.; Kikuchi, Y.; Takamura, H.; Okumura, M. Japanese Sentence Compression with a Large Training Dataset. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, BC, Canada, 30–4 July 2017; pp. 281–286.
30. Vu, T.; Hu, B.; Munkhdalai, T.; Yu, H. Sentence Simplification with Memory-Augmented Neural Networks. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, New Orleans, LA, USA, 1–6 June 2018; pp. 79–85.
31. Chopra, S.; Auli, M.; Rush, A. 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, San Diego, CA, USA, 12–17 June 2016; pp. 93–98.
32. Schuster, M.; Paliwal, K. Bidirectional Recurrent Neural Networks. *IEEE Trans. Signal Process.* **1997**, *45*, 2673–2681. [CrossRef]
33. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]

34. Pustejovsky, J.; Castano, J.; Ingria, R.; Sauri, R.; Gaizauskas, R.; Setzer, A.; Katz, G.; Radev, D. TimeML: Robust Specification of Event and Temporal Expression in Text. In Proceedings of the 5th International Workshop on Computational Semantics, Tilburg, The Netherlands, 15–17 January 2003; pp. 28–34.
35. Chambers, N.; Cassidy, T.; McDowell, B.; Bethard, S. Dense Event Ordering with a Multi-Pass Architecture. *Trans. Assoc. Comput. Linguistics* **2014**, *2*, 273–284. [CrossRef]
36. Chaudhari, S.; Polatkan, G.; Ramanath, R.; Mithal, V. An Attentive Survey of Attention Models. *arXiv* **2019**, arXiv:1904.02874.
37. Goldberg, Y. *Neural Network Methods in Natural Language Processing*; Synthesis Lectures on Human Language Technologies; Morgan & Claypool Publishers: San Rafael, CA, USA, 2017; Volume 10, pp. 1–309.
38. Kingma, D.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015.