Is *Killed* More Significant than *Fled*? A Contextual Model for Salient Event Detection

Disha Jindal[†], **Daniel Deutsch, Dan Roth**
Department of Computer and Information Science, University of Pennsylvania
{djjindal, ddeutsch, danroth}@seas.upenn.edu

Abstract

Identifying the key events in a document is critical to holistically understanding its important information. Although measuring the salience of events is highly contextual, most previous work has used a limited representation of events that omits essential information. In this work, we propose a highly contextual model of event salience that uses a rich representation of events, incorporates document-level information and allows for interactions between latent event encodings. Our experimental results on an event salience dataset (Liu et al., 2018) demonstrate that our model improves over previous work by an absolute 2-4% on standard metrics, establishing a new state-of-the-art performance for the task. We also propose a new evaluation metric which addresses flaws in previous evaluation methodologies. Finally, we discuss the importance of salient event detection for the downstream task of summarization.[1]

1 Introduction

Identifying the salient information in a given piece of text is a ubiquitous and important problem in natural language understanding. While important parts of the text have been identified by attending to entities (Dunietz and Gillick, 2014), elementary discourse units (Xu et al., 2020), or whole sentences (Zhou et al., 2018; Liu and Lapata, 2019), in this work, we choose to model extracting important events (Liu et al., 2018; Choubey et al., 2018). Events are the core parts of most sentences – they center around a predicate and include its key arguments – yet they are compact semantic units, and a salient event in a sentence could carry the sentence's meaning efficiently. Extracting important events has been shown to be central to many downstream tasks, such as summarization (Marujo, 2015), storyline creation (Martin et al., 2018) and question answering (Kociský et al., 2018).

To model the importance of an event, it is critical to understand its context: who is involved, where did it happen, what other events is it related to, and more. For instance, in Figure 1, the difference in salience of the "fled" events in each document is significantly influenced by their arguments ("2,100 Colombians" versus "shopkeepers"). Previous work that identifies salient events has a limited event representation that is unable to capture these important contextual signals (Liu et al., 2018; Choubey et al., 2018). In contrast, the model which we propose in this work (§3) directly models the context of an event in three different ways as follows.

First, instead of representing an event by only its *predicate mention*, our representation includes the subject, object, time, and location of the event (§3.1). Second, we directly incorporate global features into the model that capture hierarchical relations between events, abstract event frames, position information, and more (§3.2). Third, we use a neural network architecture that includes an inter-event interaction layer, which allows information to be passed between latent event encodings so that other events may increase or decrease another's importance (§3.3). Our experimental results on a standard event salience dataset (Liu et al., 2018) demonstrate that these contextual signals significantly increase performance (§4.4). We find that our model performs 2-4% better than previous work, setting a new state-of-the-art performance for the task.

[†] Currently at Google Research (djjindal@google.com).

[1] Our code is available at http://cogcomp.org/page/publication_view/918.

Proceedings of the 28th International Conference on Computational Linguistics, pages 114–124
Barcelona, Spain (Online), December 8-13, 2020

If Colombia is going to be another Vietnam, as everyone keeps saying, then Ecuador is going to become the Cambodia of this war, Maximo Abad Jaramillo, the mayor here, warned. We are not ready for this war, we don't want to be a part of it, but we are being dragged into the conflict against our will. In December alone, the local police say, {20 people}$_{\textbf{subject}}$ were **killed** {here}$_{\textbf{object}}$, 15 of them in clashes among Colombians. As of Dec. 31, nearly {2,100 Colombians}$_{\textbf{subject}}$ had **fled** {the fighting just across the border}$_{\textbf{object}}$ and registered with the Roman Catholic Church in Lago Agrio.

Israeli troops and tanks occupied positions in Jenin on Oct. 18, the day after {Palestinian radicals}$_{\textbf{subject}}$ **killed** the {Israeli minister of tourism}$_{\textbf{object}}$ {in a Jerusalem hotel}$_{\textbf{location}}$. Yael Shaluka, who works at a bakery in the market, said she heard one of the gunmen screaming, Kill them! As terrified {shopkeepers}$_{\textbf{subject}}$ **fled** {their stalls}$_{\textbf{object}}$, the men cut through an aisle of the market, now pursued by policemen and an army reservist.

Figure 1: Two sample documents from the New York Times Annotated Corpus with "killed" and "fled" events. Event mentions in bold, arguments in braces. The colors red and blue indicate whether that event is annotated as salient or not. Some of the events' arguments (e.g., "2,100 Colombians" and "Israeli minister of tourism") elevate the importance of their respective events.

In addition to proposing a model for salient event detection, we also provide a new evaluation metric for the task (§4.2). Previous work has evaluated the top-k salient events a model outputs using precision@k and recall@k, where the recall term is normalized by the total number of salient events in the document. However, this metric has some undesirable properties; for example, a perfect model could have, for different documents, variable recall@k scores that depend on the number of events in the document. We propose a more interpretable metric, normalized recall@k, that addresses these issues. In addition, our new metric avoids duplication counting due to co-referenced events.

Finally, we discuss the potential impact that modeling salient events could have on the downstream task of extractive summarization (§5). We find that the sentence-level extractive oracle that is frequently used to train summarization systems misses a significantly large portion of sentences with important events, a result which suggests that an event-based oracle could provide a better supervision signal. Further, because events are more fine-grained than full sentences, an event-focused model can discard unimportant information in a sentence to generate a more-concise summary.

The contributions of this work are three-fold: (1) We propose a contextual model for salient event detection that achieves state-of-the-art results; (2) We provide a sensible and more interpretable metric for evaluating extracted salient events; (3) We demonstrate that extractive summarization systems could potentially gain from modeling important events.

2 Related Work

Important Information Identification has been a topic of interest in NLP community since the 1980s and through these years, researchers have defined salience in multiple ways. Mann and Thompson (1988) divides the text into nuclei and satellite with an idea that a satellite is incomprehensible without the nucleus but, after removing some satellites, the text can still be understood. Upadhyay et al. (2016) discusses identifying events that would have triggered the author to write that article. Choubey et al. (2018) proposed the idea that the central events have a large number of coreferential event mentions and those mentions are spread throughout the document. However, we believe that in realistic documents, redundancy is commonly used for various rhetorical or other reasons, it is not necessarily the case that frequently mentioned events convey the main point of the article. We follow proposals from entity salience (Dunietz and Gillick, 2014) and event salience work (Liu et al., 2018) that suggest that these are difficult to explicitly define, but can be learned from observing human summaries: events that appear in the summary are salient.

Event representation has also evolved over time. Chambers and Jurafsky (2008) represented narrative events as pairs of verb and the grammatical dependency relation between the verb and the entity. Do et al. (2011) included nominal predicates, using the nominal form of verbs and lexical items under the Event frame in FrameNet (Baker et al., 1998). Further work by Balasubramanian et al. (2013), Pichotta and Mooney (2014), and others incorporated arguments such as propositional objects. However, most of the

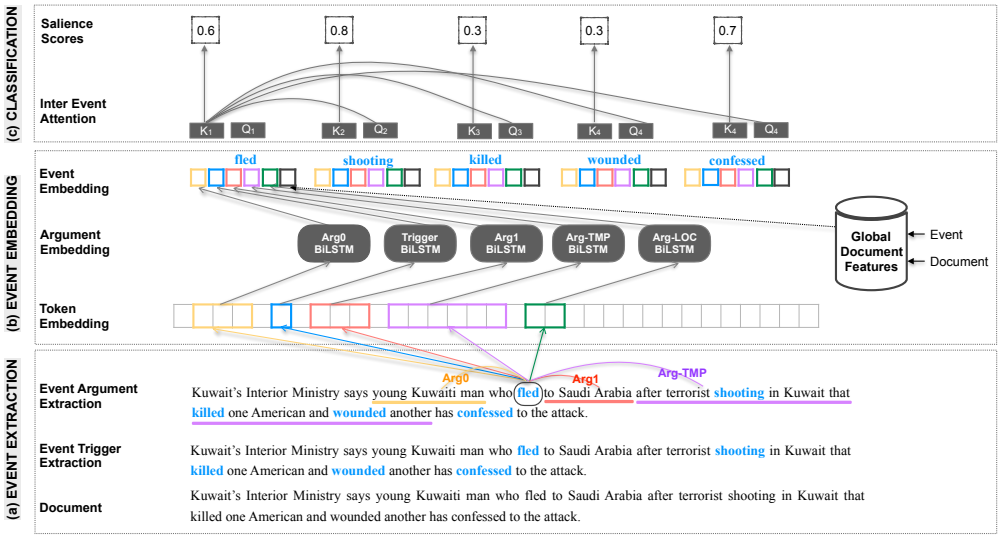

Figure 2: Illustration of our proposed framework. Given a document, we first extract all events from it. Then, we grab the token level embeddings of all constituents from the document encoded using BERT, and compose an event level embedding. Finally, in the classification module, all events attend to/vote each other and the salience score of an event is calculated by accumulating the votes from all events.

work on event salience identification has represented events as verbs/nominal event mentions. To address this, we use a more holistic event representation which includes nominal/verbal event mentions, entities in the subject and object of the predicate as well as the time and location. A similar representation was used by Peng et al. (2016) to build an event detection and co-reference system.

From modeling perspective, most earlier work (Decker, 1985; Kay and Aylett, 1996) on event salience has built rule based systems (e.g. presence of the event in the main clause, its voice, etc.). More recent work has focused on capturing coreference relation between events (Choubey et al., 2018) and on automatically capturing salient specific interactions between the discourse units (Liu et al., 2018). However, as mentioned earlier, we believe that events are highly contextual, so we use a more expressive model to obtain contextualized embeddings for events. It additionally helps us in capturing local inter-event interactions and, to capture the document level interactions, we design a number of global document level features.

3 Salient Event Extraction

We choose to model salient event extraction as a binary classification task. For every verbal and nominal event mention $e_1, \ldots, e_n$ in a document, the goal is to predict whether or not each event is salient. In our experimental study we assume that the event mentions are provided.

In the following sections, we describe our model's event representation (§3.1), the global features which are incorporated into the representation (§3.2), and the network architecture and inter-event interaction mechanisms (§3.3).

3.1 Event Representation

The standard method for representing an event is to define it based on the span of text that represents its predicate mention in a document. However, this representation is clearly suboptimal because many other important signals in the text are missing which could add critical information to determine the importance of an event. For instance, it is difficult to determine whether a "snow" event is important by itself, but knowing that the event took place in during the summer increases the rarity of the event, potentially elevating its importance in the document.

Subsequently, we define an event to be a 5-tuple $e_i = (m_i, s_i, o_i, t_i, \ell_i)$ where each item in the tuple

corresponds to the contiguous span of text for the event mention, subject, object, time, and location, respectively. In practice, the arguments for each event correspond to the ARG0, ARG1, ARGM-LOC, and ARGM-TMP arguments output from verbal and nominal semantic role labeling systems (He et al., 2017; Khashabi et al., 2018).

After the arguments for each event have been extracted, they are combined together to form a vector representation for the event, denoted $\mathbf{e}_i$, as follows. We first obtain the BERT embedding (Devlin et al., 2019) for every token in the input document. Then, since each of the items of the event tuple is a contiguous span of text, we create a fixed-size representation for each argument by encoding the corresponding tokens using a bidirectional LSTM. There is a separate LSTM encoder for each argument type. Finally, the vectors for all of the arguments are concatenated together to form the event encoding, $\mathbf{e}_i = [\mathbf{m}_i; \mathbf{s}_i; \mathbf{o}_i; \mathbf{t}_i; \boldsymbol{\ell}_i]$.

3.2 Event Augmentation with Global Features

Although BERT embeddings have proven to be highly beneficial for a large number of tasks, they may not encode all of the information that is useful for the task. This is especially true for high-level document features with long-range dependencies. Therefore, we augment the event representation from the previous section with features that leverage the event structure, document-level statistics, event-event relations, and event abstractions. Refer to Table 1(b) for a summary of the statistics of these features on the New York Times (NYT) Annotated Corpus (Sandhaus, 2008). We extract the following features:

1. **Parent Score:** This feature leverages the hierarchical relations between events in a document. The intuition is that the higher level and more abstract events are relatively more salient. We use the model from Wang et al. (2020) to identify the parent-child relationship between every event pair. An event is called a child of another event if it is a subevent of the parent (e.g., "shooting" may be a subevent/child of an "attack" event). The Parent Score of an event is defined as the number of child events it has.

2. **Frame Name:** Framename provides a more abstract understanding of the event compared to the event trigger. All event triggers (9716 in total) in the Event Salience corpus (Liu et al., 2018) belong to a total of 569 frames (annotated using Semafor (Das and Smith, 2011)). For instance, Figure 3 (right) shows all event triggers under the frame "*Killing*." As can be seen from the figure, the frequency of all events within a frame is very different, whereas their salience in the text is usually the same. Therefore, this feature enables events with low frequency to leverage the understanding from more frequent events under the same frame.

3. **Sentence Location:** One of the most commonly used (Dunietz and Gillick, 2014; Liu et al., 2018) features for salience related tasks. It represents the first location of sentence containing this event.

4. **Event Trigger Frequency:** The number of times the event trigger appears in the document. Table 1 shows that salient events are, on average, significantly more frequent than non salient events.

5. **Argument Frequency:** We leverage the event structure to design this feature. Argument Frequency of an event is the maximum number of times any of its arguments (ARG0/1) appear in the document.

6. **Named Argument:** Since Named Entities signify their relative importance compared to other entities, we add a binary feature representing whether any of the event arguments is a Named Entity.

After these features have been computed for each event, they are concatenated to the event encoding $\mathbf{e}_i$ to get the final representation.

3.3 Inter-Event Interactions

After each event has been encoded into a vector representation $\mathbf{e}_i$, the model needs to make a binary decision about whether or not an event is salient. Our baseline model assigns a label $\hat{y}_i \in \{0, 1\}$ to event e_i using a sigmoid classification layer:

$$\hat{y}_i = \sigma(\mathbf{W}\mathbf{e}_i + b) \tag{1}$$

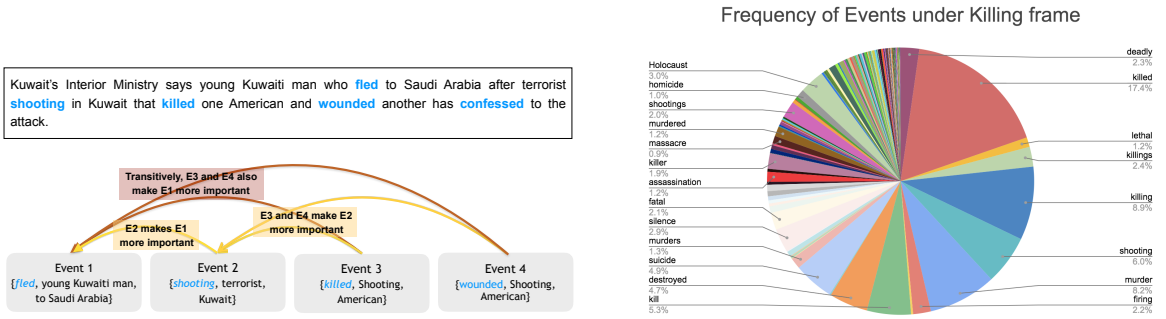

Figure 3: **(Left)** Example inter-event relations. Yellow lines represent lexical matches between event representations, and brown lines imply a transitive relationship between events. Although Events 1 and 3 have no lexical matches, we are able to infer a relationship between them due to both having lexical matches with Event 2. **(Right)** Event trigger frequencies for the *Killing* frame on the New York Times Annotated Corpus.

where $\mathbf{W}$ and b are learned parameters. The model is trained using a binary cross-entropy loss and denoted CEE-BASE (Contextual Event Extractor).

Intuitively, it may be beneficial to allow for the event representation vectors to interact with each other, thereby allowing one event to increase or decrease the salience of another event. In our model, this is done by adding additional modules in between the event encoding and classification layer. We experiment with two different methods of inter-event interactions as follows.

Inter-Event Attention Module The idea behind this classification module is to capture inter-event votes (Gu et al., 2020). The events which get higher votes from others will have a larger attention score, with the intuition that the supporting events will increase the salience of their corresponding main events and that the irrelevant and noisy events will be ignored. Specifically, given the representation of an event $\mathbf{e}_i$, a key and query vector are calculated as

$$\mathbf{k}_i = \mathbf{W}_k \mathbf{e}_i \tag{2}$$

$$\mathbf{q}_i = \mathbf{W}_q \mathbf{e}_i \tag{3}$$

where $\mathbf{W}_k$ and $\mathbf{W}_q$ are learned parameters. Then, the attention score a_i of an event is calculated as follows:

$$\hat{y}_i = \sigma\left(\sum_{j=1, j\neq i}^{n} \mathbf{q}_j^{\intercal} \cdot \mathbf{k}_i\right) \tag{4}$$

We refer to models that use this attention mechanism as CEE-IEA.

Dynamic Memory Module Since events are highly contextual, a given event forms discourse relations with other events in the document. For instance, in Figure 3 (left), we assume that events 1 and 2 are related to each other because their arguments have some lexical overlap. Since the same is true for events 2 and 3, we can infer that events 1 and 3 might be related even though they share no lexical overlap themselves. To capture such transitive inter-event relations, we repurpose Dynamic Memory Networks (Xiong et al., 2016, DMNs) for our task. DMNs make T passes over the input event vectors, each time refining a episodic memory vector $\mathbf{m}_i^t$ for event e_i based on the previous iteration and the other events. The multiple iterations allow information to flow transitively across events. Finally, the output score is calculated based on $\mathbf{m}_i^T$:

$$\mathbf{m}_i^t = \text{ReLU}(\mathbf{W}_1[\mathbf{m}_i^{t-1}; \mathbf{c}_i^t; \mathbf{e}_i] + b_1) \tag{5}$$

$$\hat{y}_i = \sigma(\mathbf{W}_2 \mathbf{m}_i^T + b_2) \tag{6}$$

where $\mathbf{c}_i^t = \text{AttnGRU}([\mathbf{e}_j]); j = [1, n], j \neq i$ (see Xiong et al. (2016) for details about the AttnGRU function).

	Train	Val	Test
# Documents originally	598738	64000	64598
# With no salient events	98036	11733	11981
# With missing data	8103	974	1010
# After filtering	492599	51293	51607

	Non Salient	Salient
Avg. Sentence Location	8.57	6.62
Avg. Trigger Frequency	2.18	6.76
Avg. Parent Score	0.045	0.097
Avg. Arg Frequency	3.34	3.47
Avg. Named Arg	0.13	0.15

Table 1: Dataset Statistics. **(Left)** Preprocessing: $\sim$18.3% documents do not have any salient events and $\sim$1.5% are missing body/abstract. Results in Tables 2, 3 and 4 are computed on the clean filtered set. **(Right)** Feature statistics of Salient vs Non Salient Events.

4 Experiments

4.1 Dataset

For the experimental evaluation of our contextual event salience module (and subsequent discussion of extractive summarization), we use the New York Times dataset (Sandhaus, 2008), which is a large corpus of articles that were published between 1996 and 2007 and their corresponding summaries. We use the salient event annotations for this dataset provided by Liu et al. (2018). In their work, events in the document are marked as salient based on whether the event mention's lemma is present in the abstractive summary. Due to the annotation procedure, around 18.3% of the instances had no salient events and were subsequently omitted from the evaluation. Refer to Table 1 (left) for the details on dataset preparation.

4.2 Evaluation Metric

We evaluate our event salience model on three metrics: Precision@k (P@k), Recall@k (R@k) and Normalized Recall@k (NR@k) for k = 1, 5, 10. Previous work has reported results on P@k and R@k where:

$$\text{P@k} = \frac{\textit{\# of salient events in top k predictions}}{k}$$

$$\text{R@k} = \frac{\textit{\# of salient events in top k predictions}}{\textit{\# of salient events in the doc}}$$

However, R@k is not a very comprehensible metric because: i) The maximum value of recall is variable and ii) averaging R@k across documents with different number of salient events is biased towards documents with fewer events. Consequently, we propose a new metric, NR@k, which gives equal importance to all documents and has a maximum value is 1 for all k, making the metric easier to understand.

$$\text{NR@k} = \frac{\textit{\# of salient events in top k predictions}}{\textit{min(k, \# of salient events in the doc)}}$$

Additionally, we can see from the Table 1 (right) that the average trigger frequency of non salient events is 2.18 whereas those of salient events is 6.76. So, depending upon the correference ability of a model, for each true positive, on an average the model can be given a reward of 6.76 whereas for each false positive, the model can be penalized by a score of 2.18. For a fair understanding of a model's ability to identify top k events, a reward/penalty of one should be given to each unique event. Therefore, we also calculate P@k and NR@k using only the top-k *unique* model predictions. For a direct comparison of previously reported results, we also include results with the original metrics.

4.3 Implementation Details

We use sub word tokenizer from BERT to tokenize the documents and fine-tune *bert-base-uncased*[2] version of BERT in all of our settings. Our models are implemented in PyTorch (Paszke et al., 2017). Token level BERT embeddings of size 768 are passed through different BiLSTM modules to get embeddings of the event mention (of size 512) and all other constituents (of size 64 each) to get the final event embedding of size 768. To add the Frame Name feature from § 3.2, we first get the event embedding of size 768 as

[2]https://git.io/fhbJQ

Method	P@1	P@5	P@10	NR@1	NR@5	NR@10
LOCATION	44.65	36.49	28.07	44.65	52.81	62.27
FREQUENCY	51.27	37.20	27.85	51.27	51.68	60.59
KCE	61.55	43.94	32.73	61.55	62.81	71.85
CEE-BASE(-GF)	58.17	38.90	28.15	58.17	58.89	66.45
CEE-DMN(-GF)	58.58	40.10	29.12	58.58	59.12	67.52
CEE-IEA(-GF)	59.44	41.47	30.59	59.44	59.98	68.03
CEE-IEA	**65.44**(+3.89%)	**46.85**(+2.91%)	**34.11**(+1.38%)	**65.44**(+3.89%)	**66.90**(+4.09%)	**74.33**(+2.48%)

Table 2: Event Salience Identification results on the New York Times Annotated Corpus using the normalized and unique version of the metrics. CEE-BASE represents the base model from §3.3. CEE-IEA and CEE-DMN are the models with inter-event interaction layers. -GF signifies the corresponding models without augmenting with global document features from §3.2. Improvements (in parentheses) and statistical significance (underlined) are with respect to the KCE baseline.

Method	P@1	P@5	P@10	R@1	R@5	R@10
LOCATION	43.50	37.63	30.62	9.88	32.71	46.43
FREQUENCY	55.62	49.27	42.18	9.69	34.86	52.30
KCE	61.81	52.36	44.54	11.58	39.36	57.79
CEE-IEA	**65.47**(+3.66%)	**54.26**(+1.90%)	**44.93**(+0.39%)	**13.12**(+1.54%)	**42.03**(+2.67%)	**59.65**(+1.86%)

Table 3: Event Salience Identification results on the New York Times Annotated Corpus using the metrics (non-unique or normalized) used in the previous work.

mentioned above. After that, we get a BERT embedding of size 768 for the corresponding frame name. Both of these are concatenated to get the final event embedding of size 1536.

Model parameters are optimized by Adam (Kingma and Ba, 2015). Our models are trained for 30k steps on 4 GPUs (TITAN RTX). We evaluated the model after every 750 steps and saved the best checkpoint based on the validation loss. The test results are reported by evaluating our test set's performance on this checkpoint.

4.4 Results

We compare our model to three baseline models: LOCATION (which selects the first k events), FREQUENCY (which selects the k events which appear most frequently in the document), and the Kernel Centrality Estimation (KCE) model from Liu et al. (2018). KCE models relationships between events using K Gaussian kernels and adds *Sentence Location* and *Frequency* features along with three others which capture similarity with entities and other events in the document. The main results on the event salience task are summarized in Table 2.

First, among the baseline models, KCE performs consistently the best across all evaluation metrics with an absolute 10 point improvement over LOCATION and FREQUENCY on both P@1 and NR@1.

Then, we compare the results of our base model with the two attention-based variants without including any global features from the document. The dynamic memory network CEE-DMN provides an 0.41 point improvement in P@1 and NR@1. The inter-event attention module CEE-IEA provides an even larger improvement, with 1.27. The stronger performance of the CEE-IEA model shows that the voting attention module helps to promote the salient events and suppress the noisy and background events.

The addition of the global features in Table 2 on top of the better performing inter-event attention model provides yet the largest improvement over the baseline models. Specifically, it improves over KCE by 3.89% P@1. The improvement is consistent even with higher values of k, with a 1.38% improvement in P@10 and 2.48% NR@10. This result suggests that the global features are indeed critical for identifying salient events and that neither the event representation nor inter-event interactions capture the same information that the features do.

Then, in Table 3, we present the results of our best performing model and the baseline models using the

Method	P@1	P@5	P@10	NR@1	NR@5	NR@10
CEE-IEA(-GF)	59.44	41.47	30.59	59.44	59.98	68.03
+(FN)	60.88(+1.44%)	42.27(+0.8%)	30.75(+0.16%)	60.88(+1.44%)	61.21(+1.23%)	68.68(+0.65%)
+(SL,TF)	61.77(+0.89%)	44.80(+2.53%)	33.27(+2.52%)	61.77(+0.89%)	63.49(+2.28%)	71.98(+3.30%)
+(PS)	65.07(+3.30%)	46.48(+1.68%)	33.99(+0.72%)	65.07(+3.30%)	65.78(+2.29%)	73.42(+1.44%)
+(AF,NA)	65.44(+0.37%)	46.85(+0.37%)	34.11(+0.12%)	65.44(+0.37%)	66.90(+1.12%)	74.33(+0.91%)

Table 4: Ablation results demonstrating relative importance of each feature. Abbreviations: FN = Frame Name, SL = Sentence Location, TF = Trigger Frequency, AF = Argument Frequency, NA = Named Argument and PS = Parent Score. Improvements are shown w.r.t the previous row and statistically significant improvements using a permutation test with $p < 0.0001$ are underlined.

original metrics formulation (without the normalization or uniqueness). The improvement of the model in this work over the baseline models is similarly consistent at all values of k.

All together, the results from our experiments demonstrate that the CEE-IEA model with global features performs better at extracting salient events and sets a new state-of-the-art performance for this task.

4.5 Ablation: Feature Contribution

Due to the significant improvement in the model's performance when the global features were added, we conduct an ablation study on the features to understand what contributes most to the gains. Table 4 shows the performance of the CEE-IEA model as different features are successively added to the model. We observe that all of the features contribute positively to both the precision and recall of the models at all values of k. Among the features, the two which provided the largest P@1 improvements are those which have not been used by previous work: Parent Score and Frame Name. This result suggests that providing the model with information that distinguishes high- and low-level events is beneficial for detecting salient events.

As the value of k increases, the benefit of Sentence Location and Trigger Frequency features increase. This shows that these features help identifying salient events from the rest. Whereas, Parent Score and Frame Name features further help recognize the most salient event among these.

5 Case Study: Extractive Summarization

In this section, we discuss the potential benefits of modeling event salience for the downstream task of *extractive summarization*. Ideally, a good summary captures information about the key *events* in the original document. However, most common models for extractive summarization operate at the more coarse-grained sentence level. We discuss below why we believe that modeling *event* importance rather than sentence importance has the potential to benefit summarization.

An Improved Supervision Signal First, the extractive summarization training oracles often miss important signals in the data. Extractive summarization systems are trained on 0/1 labels assigned to each sentence in the document to indicate whether or not that sentence should be selected to the summary. The most common procedure for obtaining these labels is to greedily select document sentences while the ROUGE (Lin, 2004) score between the selected sentences and a reference summary still increases (Nallapati et al., 2017).

We compared this labeling procedure to an alternative method based on events. The event-based oracle selects the sentence which contains the first occurrence of an event in the document for each event in the summary. For our analysis, the sentences selected by the ROUGE-based and event-based methods are divided into three sets: B, sentences chosen by both; R, sentences chosen by ROUGE only; and E, sentences chosen by the event method only. The relative sentence coverage for both sets of selected sentences is calculated as follows.

$$E_{Cov} = \frac{1}{N}\sum_{i=1}^{N}\frac{|B_i|}{|B_i| + |R_i|} \qquad R_{Cov} = \frac{1}{N}\sum_{i=1}^{N}\frac{|B_i|}{|B_i| + |E_i|}$$

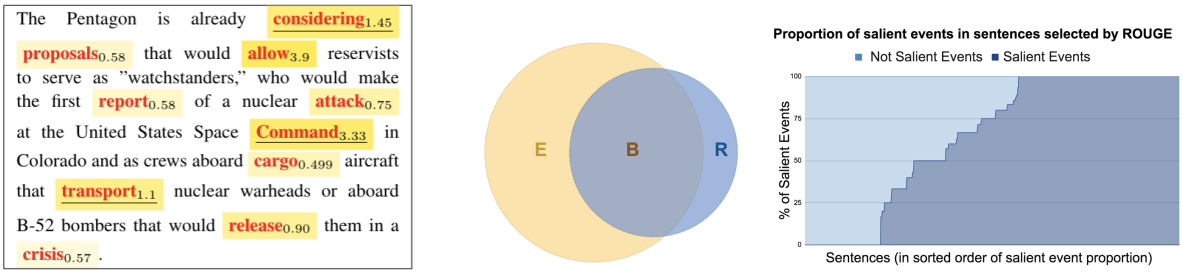

Figure 4: **(Left) Predicted salience scores** of all events from our event based model are shown in the subscripts. In contrast to a sentence based summarization system, event based system provides a much more fine grained output by predicting the relative importance of each *event*. **(Middle)** Disjoint sentence sets *E*, *B* and *R*. It shows relative coverage of 79.8% by salient event signal and 48.4% by ROUGE based signal. **(Right)** Proportion of salient events in $\{B \cup R\}$, the light blue area shows that **27.6%** of the events from sentences selected by ROUGE are not salient.

We observe that, E_{Cov} = **79.8**% which means that most of the sentences in the ROUGE-based oracle are covered by event based summaries. However, R_{Cov} = **48.4**%, which means that the standard supervision signal misses a significant number of events present in the reference summary (all salient events in the remaining 51.6% sentences; see Figure 4 (middle)). The lower value of R_{Cov} is evidence that ROUGE-based oracles are missing a valuable supervision signal.

Finer-Grained Selection Since the most popular extractive summarization models are forced to select full sentences to be in the summary, it likely that the model selects a lot of unimportant information. For instance, if a document sentence is quite long and covers a lot of information, it is likely that some of the predicate mentions in it are not salient and may not need to be in the summary. Consequently, including only semantic unit that consist of important events and their arguments may be a better method of identifying text that should belong in the summary. To quantify this, we calculate the number of salient events in the sentences selected by ROUGE (set $\{B \cup R\}$) and observed that **27.6%** of the events in these selected sentences are not salient (see Figure 4 (right)).

Together, these two points demonstrate that the current method of creating oracle for training summarization systems misses important events which are present in the gold summaries while including a significant number of non salient events. This underscores that a good summarization system can benefit from our event salience model to select better events for generating summaries.

6 Conclusion

In this work, we proposed a contextual model for salient event identification. We demonstrated that the three different components of our model (the event representation, global features, and inter-event interaction) combine to produce state-of-the-art results on the New York Times Annotated Corpus. Further, we identified issues with previous evaluation metrics and proposed new intuitive evaluation methods. Finally, we discussed how event salience can be helpful for other downstream applications through a case study of extraction summarization.

Acknowledgements

We would like to thank the anonymous reviewers for their helpful feedback and suggestions. This research is supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via IARPA Contract No. 2019-19051600006 under the BETTER Program, and by contract FA8750-19-2-1004 with the US Defense Advanced Research Projects Agency (DARPA). The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

References

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet Project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, COLING-ACL 1998*, pages 86–90.

Niranjan Balasubramanian, Stephen Soderland, Mausam, and Oren Etzioni. 2013. Generating coherent event schemas at scale. In *EMNLP*.

Nathanael Chambers and Daniel Jurafsky. 2008. Unsupervised learning of narrative event chains. In *ACL*.

Prafulla Kumar Choubey, Kaushik Raju, and Ruihong Huang. 2018. Identifying the most dominant event in a news article by mining event coreference relations. In *NAACL-HLT*.

Dipanjan Das and Noah A. Smith. 2011. Semi-supervised frame-semantic parsing for unknown predicates. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1435–1444, Portland, Oregon, USA, June. Association for Computational Linguistics.

Nan Decker. 1985. The use of syntactic clues in discourse processing. In *ACL*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Quang Do, Yee Seng Chan, and Dan Roth. 2011. Minimally Supervised Event Causality Identification. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Edinburgh, Scotland, 7.

Jesse Dunietz and Daniel Gillick. 2014. A new entity salience task with millions of training examples. In *EACL*.

Xiaotao Gu, Yuning Mao, Jiawei Han, Jialu Liu, Hongkun Yu, Yingfang Wu, Cong Yu, Daniel Finnie, Jiaqi Zhai, and Nicholas Zukoski. 2020. Generating representative headlines for news stories. *ArXiv*, abs/2001.09386.

Luheng He, Kenton Lee, Mike Lewis, and Luke Zettlemoyer. 2017. Deep semantic role labeling: What works and what's next. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 473–483, Vancouver, Canada, July. Association for Computational Linguistics.

Roderick Kay and Ruth Aylett. 1996. Transitivity and foregrounding in news articles: Experiments in information retrieval and automatic summarising. In *ACL*.

Daniel Khashabi, Mark Sammons, Ben Zhou, Tom Redman, Christos Christodoulopoulos, Vivek Srikumar, Nicholas Rizzolo, Lev Ratinov, Guanheng Luo, Quang Do, Chen-Tse Tsai, Subhro Roy, Stephen Mayhew, Zhili Feng, John Wieting, Xiaodong Yu, Yangqiu Song, Shashank Gupta, Shyam Upadhyay, Naveen Arivazhagan, Qiang Ning, Shaoshi Ling, and Dan Roth. 2018. CogCompNLP: Your Swiss Army Knife for NLP. In *Proc. of the International Conference on Language Resources and Evaluation (LREC)*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Tomás Kociský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The narrativeqa reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.

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

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

Zhengzhong Liu, Chenyan Xiong, Teruko Mitamura, and Eduard Hovy. 2018. Automatic event salience identification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1226–1236, Brussels, Belgium, October-November. Association for Computational Linguistics.

William C Mann and Sandra A Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8(3):243–281.

Lara J. Martin, Prithviraj Ammanabrolu, William Hancock, Shruti Singh, Brent Harrison, and Mark O. Riedl. 2018. Event representations for automated story generation with deep neural nets. *ArXiv*, abs/1706.01331.

Luís Marujo. 2015. Event-based multi-document summarization. Ph.D. thesis. Carnegie Mellon University.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *AAAI*.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch.

Haoruo Peng, Yangqiu Song, and Dan Roth. 2016. Event detection and co-reference with minimal supervision. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 392–402, Austin, Texas, November. Association for Computational Linguistics.

Karl Pichotta and Raymond J. Mooney. 2014. Statistical script learning with multi-argument events. In *EACL*.

E. Sandhaus. 2008. The New York Times Annotated Corpus. *Linguistic Data Consortium, Philadelphia*.

Shyam Upadhyay, Christos Christodoulopoulos, and Dan Roth. 2016. "making the news": Identifying noteworthy events in news articles. In *Proceedings of the Fourth Workshop on Events*, pages 1–7, San Diego, California, June. Association for Computational Linguistics.

Haoyu Wang, Muhao Chen, Hongming Zhang, and Dan Roth. 2020. Joint constrained learning for event-event relation extraction.

Caiming Xiong, Stephen Merity, and Richard Socher. 2016. Dynamic memory networks for visual and textual question answering. In *ICML*.

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

Qingyu Zhou, Nan Yang, Furu Wei, Shaohan Huang, Ming Zhou, and Tiejun Zhao. 2018. Neural document summarization by jointly learning to score and select sentences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 654–663, Melbourne, Australia, July. Association for Computational Linguistics.