Automatic Event Salience Identification

Zhengzhong Liu **Chenyan Xiong** **Teruko Mitamura** **Eduard Hovy**
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213, USA
{liu, cx, teruko, hovy}@cs.cmu.edu

Abstract

Identifying the salience (i.e. importance) of discourse units is an important task in language understanding. While events play important roles in text documents, little research exists on analyzing their saliency status. This paper empirically studies the *Event Salience* task and proposes two salience detection models based on content similarities and discourse relations. The first is a feature based salience model that incorporates similarities among discourse units. The second is a neural model that captures more complex relations between discourse units. Tested on our new large-scale event salience corpus, both methods significantly outperform the strong frequency baseline, while our neural model further improves the feature based one by a large margin. Our analyses demonstrate that our neural model captures interesting connections between salience and discourse unit relations (e.g., scripts and frame structures).

1 Introduction

Automatic extraction of prominent information from text has always been a core problem in language research. While traditional methods mostly concentrate on the word level, researchers start to analyze higher-level discourse units in text, such as entities (Dunietz and Gillick, 2014) and events (Choubey et al., 2018).

Events are important discourse units that form the backbone of our communication. They play various roles in documents. Some are more central in discourse: connecting other entities and events, or providing key information of a story. Others are less relevant, but not easily identifiable by NLP systems. Hence it is important to be able to quantify the "importance" of events. For example, Figure 1 is a news excerpt describing a debate around a jurisdiction process: "*trial*" is central as the main discussing topic, while "*war*" is not.

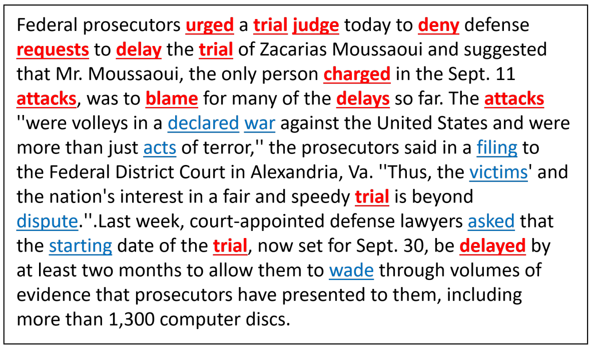
Federal prosecutors **urged** a **trial** **judge** today to **deny** defense **requests** to **delay** the **trial** of Zacarias Moussaoui and suggested that Mr. Moussaoui, the only person **charged** in the Sept. 11 **attacks**, was to **blame** for many of the **delays** so far. The **attacks** ''were volleys in a declared war against the United States and were more than just acts of terror,'' the prosecutors said in a filing to the Federal District Court in Alexandria, Va. ''Thus, the victims' and the nation's interest in a fair and speedy **trial** is beyond dispute.''.Last week, court-appointed defense lawyers asked that the starting date of the **trial**, now set for Sept. 30, be **delayed** by at least two months to allow them to wade through volumes of evidence that prosecutors have presented to them, including more than 1,300 computer discs.

Figure 1: Examples annotations. Underlying words are annotated event triggers; the red bold ones are annotated as salient.

Researchers are aware of the need to identify central events in applications like detecting salient relations (Zhang et al., 2015), and identifying climax in storyline (Vossen and Caselli, 2015). Generally, the salience of discourse units is important for language understanding tasks, such as document analysis (Barzilay and Lapata, 2008), information retrieval (Xiong et al., 2018), and semantic role labeling (Cheng and Erk, 2018). Thus, proper models for finding important events are desired.

In this work, we study the task of **event salience detection**, to find events that are most relevant to the main content of documents. To build a salience detection model, one core observation is that salient discourse units are forming discourse relations. In Figure 1, the "*trial*" event is connected to many other events: "*charge*" is pressed before "*trial*"; "*trial*" is being "*delayed*".

We present two salience detection systems based on the observations. First is a feature based learning to rank model. Beyond basic features like frequency and discourse location, we design features using cosine similarities among events and entities, to estimate the *content organization* (Grimes, 1975): how lexical meaning of elements relates to each other. Similarities from within-sentence or across the whole document are used to capture

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 1226–1236
Brussels, Belgium, October 31 - November 4, 2018. ©2018 Association for Computational Linguistics

interactions on both local and global aspects (§4). The model significantly outperforms a strong "Frequency" baseline in our experiments.

However, there are other discourse relations beyond lexical similarity. Figure 1 showcases some: the script relation (Schank and Abelson, 1977)[1] between "*charge*" and "*trial*", and the frame relation (Baker et al., 1998) between "*attacks*" and "*trial*" ("*attacks*" fills the "charges" role of "*trial*"). Since it is unclear which ones contribute more to salience, we design a Kernel based Centrality Estimation (KCE) model (§5) to capture salient specific interactions between discourse units automatically.

In KCE, discourse units are projected to embeddings, which are trained end-to-end towards the salience task to capture rich semantic information. A set of soft-count kernels are trained to weigh salient specific latent relations between discourse units. With the capacity to model richer relations, KCE outperforms the feature-based model by a large margin (§7.1). Our analysis shows that KCE is exploiting several relations between discourse units: including script and frames (Table 5). To further understand the nature of KCE, we conduct an *intrusion test* (§6.2), which requires a model to identify events from another document. The test shows salient events form tightly related groups with relations captured by KCE.

The notion of salience is subjective and may vary from person to person. We follow the empirical approaches used in entity salience research (Dunietz and Gillick, 2014). We consider the *summarization test*: an event is considered salient if a summary written by a human is likely to include it, since events about the main content are more likely to appear in a summary. This approach allows us to create a large-scale corpus (§3).

In this paper, we make three main contributions. First, we present two event salience detection systems, which capture rich relations among discourse units. Second, we observe interesting connections between salience and various discourse relations (§7.1 and Table 5), implying potential research on these areas. Finally, we construct a large scale event salience corpus, providing a testbed for future research. Our code, dataset and models are publicly available[2].

2 Related Work

Events have been studied on many aspects due to their importance in language. To name a few: event detection (Li et al., 2013; Nguyen and Grishman, 2015; Peng et al., 2016), coreference (Liu et al., 2014; Lu and Ng, 2017), temporal analysis (Do et al., 2012; Chambers et al., 2014), sequencing (Araki et al., 2014), script induction (Chambers and Jurafsky, 2008; Balasubramanian et al., 2013; Rudinger et al., 2015; Pichotta and Mooney, 2016).

However, studies on event salience are premature. Some previous work attempts to approximate event salience with word frequency or discourse position (Vossen and Caselli, 2015; Zhang et al., 2015). Parallel to ours, Choubey et al. (2018) propose a task to find the most dominant event in news articles. They draw connections between event coreference and importance, on hundreds of closed-domain documents, using several oracle event attributes. In contrast, our proposed models are fully learned and applied on more general domains and at a larger scale. We also do not restrict to a single most important event per document.

There is a small but growing line of work on entity salience (Dunietz and Gillick, 2014; Dojchinovski et al., 2016; Xiong et al., 2018; Ponza et al., 2018). In this work, we study the case for events.

Text relations have been studied in tasks like text summarization, which mainly focused on cohesion (Halliday and Hasan, 1976). Grammatical cohesion methods make use of document level structures such as anaphora relations (Baldwin and Morton, 1998) and discourse parse trees (Marcu, 1999). Lexical cohesion based methods focus on repetitions and synonyms on the lexical level (Skorochod'ko, 1971; Morris and Hirst, 1991; Erkan and Radev, 2004). Though sharing similar intuitions, our proposed models are designed to learn richer semantic relations in the embedding space.

Comparing to the traditional summarization task, we focus on events, which are at a different granularity. Our experiments also unveil interesting phenomena among events and other discourse units.

3 The Event Salience Corpus

This section introduces our approach to construct a large-scale event salience corpus, including methods for finding event mentions and obtaining saliency labels. The studies are based on the Annotated New York Times corpus (Sandhaus, 2008), a newswire corpus with expert-written abstracts.

[1] Scripts are prototypical sequences of events: a *restaurant* script normally contains events like "order", "eat" and "pay".

[2] https://github.com/hunterhector/EventSalience

3.1 Automatic Corpus Creation

Event Mention Annotation: Despite many annotation attempts on events (Pustejovsky et al., 2002; Brown et al., 2017), automatic labeling of them in general domain remains an open problem. Most of the previous work follows empirical approaches. For example, Chambers and Jurafsky (2008) consider all verbs together with their subject and object as events. Do et al. (2011) additionally include nominal predicates, using the nominal form of verbs and lexical items under the *Event* frame in FrameNet (Baker et al., 1998).

There are two main challenges in labeling event mentions. First, we need to decide which lexical items are event triggers. Second, we have to disambiguate the word sense to correctly identify events. For example, the word "phone" can refer to an entity (a physical phone) or an event (a phone call event). We use FrameNet to solve these problems. We first use a FrameNet based parser: Semafor (Das and Smith, 2011), to find and disambiguate triggers into frame classes. We then use the FrameNet ontology to select event mentions.

Our frame based selection method follows the Vendler classes (Vendler, 1957), a four way classification of eventuality: *states*, *activities*, *accomplishments* and *achievements*. The last three classes involve state change, and are normally considered as events. Following this, we create an "event-evoking frame" list using the following procedure:

1. We keep frames that are subframes of *Event* and *Process* in the FrameNet ontology.
2. We discard frames that are subframes of state, entity and attribute frames, such as *Entity*, *Attributes*, *Locale*, etc.
3. We manually inspect frames that are not subframes of the above-mentioned ones (around 200) to keep event related ones (including subframes), such as *Arson*, *Delivery*, etc.

This gives us a total of 569 frames. We parse the documents with Semafor and consider predicates that trigger a frame in the list as candidates. We finish the process by removing the light verbs[3] and reporting events[4] from the candidates, similar to previous research (Recasens et al., 2013).

Salience Labeling: For all articles with a human written abstract (around 664,911) in the New York Times Annotated Corpus, we extract event mentions. We then label an event mention as salient if we can find its lemma in the corresponding abstract (Mitamura et al. (2015) showed that lemma matching is a strong baseline for event coreference.). For example, in Figure 1, event mentions in bold and red are found in the abstract, thus labeled as salient. Data split is detailed in Table 1 and §6.

	Train	Dev	Test
# Documents	526126	64000	63589
Avg. # Word	794.12	790.27	798.68
Avg. # Events	61.96	60.65	61.34
Avg. # Entities	197.63	196.95	198.40
Avg. # Salience	8.77	8.79	8.90

Table 1: Dataset Statistics.

3.2 Annotation Quality

While the automatic method enables us to create a dataset at scale, it is important to understand the quality of the dataset. For this purpose, we have conducted two small manual evaluation study.

Our lemma-based salience annotation method is based on the assumption that lemma matching being a strong detector for event coreference. In order to validate this assumption, one of the authors manually examined 10 documents and identified 82 coreferential event mentions pairs between the text body and the abstract. The automatic lemma rule identifies 72 such pairs: 64 of these matches human decision, producing a precision of 88.9% (64/72) and a recall of 78% (64/82). There are 18 coreferential pairs missed by the rule.

The next question is: *is an event really important if it is mentioned in the abstract?* Although prior work (Dunietz and Gillick, 2014) shows that the assumption to be valid for entities, we study the case for events. We asked two annotators to manually annotate 10 documents (around 300 events) using a 5-point Likert scale for salience. We compute the agreement score using Cohen's Kappa (Cohen, 1960). We find the task to be challenging for human: annotators don't agree well on the 5-point scale (Cohens Kappa = 0.29). However, if we collapse the scale to binary decisions, the Kappa between the annotators raises to 0.67. Further, the Kappa between each annotator and automatic labels are 0.49 and 0.42 respectively. These agreement scores are also close to those reported in the entity salience tasks (Dunietz and Gillick, 2014).

[3] Light verbs carry little semantic information: "appear", "be", "become", "do", "have", "seem", "do", "get", "give", "go", "have", "keep", "make", "put", "set", "take".

[4] Reporting verbs are normally associated with the narrator: "argue", "claim", "say", "suggest", "tell".

While errors exist in the automatic annotation process inevitably, we find the error rate to be reasonable for a large-scale dataset. Further, our study indicates the difficulties for human to rate on a finer scale of salience. We leave the investigation of continuous salience scores to future work.

4 Feature-Based Event Salience Model

This section presents the feature-based model, including the features and the learning process.

4.1 Features

Our features are summarized in Table 2.

Basic Discourse Features: We first use two basic features similar to Dunietz and Gillick (2014): *Frequency* and *Sentence Location*. *Frequency* is the lemma count of the mention's syntactic head word (Manning et al., 2014). *Sentence Location* is the sentence index of the mention, since the first few sentences are normally more important. These two features are often used to estimate salience (Barzilay and Lapata, 2008; Vossen and Caselli, 2015).

Content Features: We then design several lexical similarity features, to reflect Grimes' content relatedness (Grimes, 1975). In addition to events, the relations between events and entities are also important. For example, Figure 1 shows some related entities in the legal domain, such as "*prosecutors*" and "*court*". Ideally, they should help promote the salience status for event "*trial*".

Lexical relations can be found both within-sentence (local) or across sentence (global) (Halliday and Hasan, 1976). We compute the local part by averaging similarity scores from other units in the same sentence. The global part is computed by averaging similarity scores from other units in the document. All similarity scores are computed using cosine similarities on pre-trained embeddings (Mikolov et al., 2013).

These lead to 3 content features: *Event Voting*, the average similarity to other events in the document; *Entity Voting*, the average similarity to entities in the document; *Local Entity Voting*, the average similarity to entities in the same sentence. Local event voting is not used since a sentence often contains only 1 event.

4.2 Model

A Learning to Rank (LeToR) model (Liu, 2009) is used to combine the features. Let ev_i denote the ith event in a document d. Its salience score is computed as:

$$f(ev_i, d) = W_f \cdot F(ev_i, d) + b \qquad (1)$$

where $F(ev_i, d)$ is the features for ev_i in d (Table 2); W_f and b are the parameters to learn.

The model is trained with pairwise loss:

$$\sum_{ev^+, ev^- \in d} \max(0, 1 - f(ev^+, d) + f(ev^-, d)), \qquad (2)$$
$$\text{w.r.t. } y(ev^+, d) = +1 \;\&\; y(ev^-, d) = -1.$$

$$y(e_i, d) = \begin{cases} +1, & \text{if } e_i \text{ is a salient entity in } d, \\ -1, & \text{otherwise.} \end{cases}$$

where ev^+ and ev^- represent the salient and non-salient events; y is the gold standard function. Learning can be done by standard gradient methods.

5 Neural Event Salience Model

As discussed in §1, the salience of discourse units is reflected by rich relations beyond lexical similarities, for example, script ("*charge*" and "*trial*") and frame (a "*trial*" of "*attacks*"). The relations between these words are specific to the salience task, thus difficult to be captured by raw cosine scores that are optimized for word similarities. In this section, we present a neural model to exploit the embedding space more effectively, in order to capture relations for event salience estimation.

5.1 Kernel-based Centrality Estimation

Inspired by the kernel ranking model (Xiong et al., 2017), we propose Kernel-based Centrality Estimation (KCE), to find and weight semantic relations of interests, in order to better estimate salience.

Formally, given a document d, the set of annotated events $\mathbb{V} = \{ev_1, \dots ev_i \dots, ev_n\}$, KCE first embed an event into vector space: $ev_i \xrightarrow{Emb} \overrightarrow{ev_i}$. The embedding function is initialized with pre-trained embeddings. It then extract K features for each ev_i:

$$\Phi_K(ev_i, \mathbb{V}) = \{\phi_1(\overrightarrow{ev_i}, \mathbb{V}), \dots, \phi_k(\overrightarrow{ev_i}, \mathbb{V}), \dots, \phi_K(\overrightarrow{ev_i}, \mathbb{V})\}, \qquad (3)$$

$$\phi_k(\overrightarrow{ev_i}, \mathbb{V}) = \sum_{ev_j \in \mathbb{V}} \exp\left(-\frac{(\cos(\overrightarrow{ev_i}, \overrightarrow{ev_j}) - \mu_k)^2}{2\sigma_k^2}\right). \qquad (4)$$

Name	Description
`Frequency`	The frequency of the event lemma in document.
`Sentence Location`	The location of the first sentence that contains the event.
`Event Voting`	Average cosine similarity with other events in document.
`Entity Voting`	Average cosine similarity with other entities in document.
`Local Entity Voting`	Average cosine similarity with entities in the sentence.

Table 2: Event Salience Features.

$\phi_k(\overrightarrow{ev_i}, \mathbb{V})$ is the k-th Gaussian kernel with mean μ_k and variance σ_k^2. It models the interactions between events in its kernel range defined by μ_k and σ_k. $\Phi_K(ev_i, \mathbb{V})$ enforces multi-level interactions among events — relations that contribute similarly to salience are expected to be grouped into the same kernels. Such interactions greatly improve the capacity of the model with negligible increase in the number of parameters. Empirical evidences (Xiong et al., 2017) have shown that kernels in this form are effective to learn weights for task-specific term pairs.

The final salience score is computed as:

$$f(ev_i, d) = W_v \cdot \Phi_K(ev_i, \mathbb{V}) + b, \tag{5}$$

where W_v is learned to weight the contribution of the certain relations captured by each kernel.

We then use the exact same learning objective as in equation (2). The pairwise loss is first backpropagated through the network to update the kernel weights W_v, assigning higher weights to relevant regions. Then the kernels use the gradients to update the embeddings, in order to capture the meaningful discourse relations for salience.

Since the features and KCE capture different aspects, combining them may give superior performance. This can be done by combining the two vectors in the final linear layer:

$$f(ev_i, d) = W_v \cdot \Phi_K(ev_i, \mathbb{V}) + W_f \cdot F(ev_i, d) + b \tag{6}$$

5.2 Integrating Entities into KCE

KCE is also used to model the relations between events and entities. For example, in Figure 1, the entity "*court*" is a frame element of the event "*trial*"; "*United States*" is a frame element of the event "*war*". It is not clear which pair contributes more to salience. We again let KCE to learn it.

Formally, let $\mathbb{E}$ be the list of entities in the document, i.e. $\mathbb{E} = \{en_1, \dots, en_i, \dots, en_n\}$, where en_i is the ith entity in document d. KCE extracts the kernel features about entity-event relations as follows:

$$\Phi_K(ev_i, \mathbb{E}) = \{\phi_1(\overrightarrow{ev_i}, \mathbb{E}), \dots, \phi_k(\overrightarrow{ev_i}, \mathbb{E}), \dots, \phi_K(\overrightarrow{ev_i}, \mathbb{E})\}, \tag{7}$$

$$\phi_k(\overrightarrow{ev_i}, \mathbb{E}) = \sum_{en_j \in \mathbb{E}} \exp\left(-\frac{(\cos(\overrightarrow{ev_i}, \overrightarrow{en_j}) - \mu_k)^2}{2\sigma_k^2}\right) \tag{8}$$

similarly, en_i is embedded by: $en_i \xrightarrow{Emb} \overrightarrow{en_i}$, which is initialized by pre-trained entity embeddings.

We reach the full KCE model by combining all the vectors using a linear layer:

$$f(ev_i, d) = W_e \cdot \Phi_K(ev_i, \mathbb{E}) + W_v \cdot \Phi_K(ev_i, \mathbb{V}) + W_f \cdot F(ev_i, d) + b \tag{9}$$

The model is again trained by equation (2).

6 Experimental Methodology

This section describes our experiment settings.

6.1 Event Salience Detection

Dataset: We conduct our experiments on the salience corpus described in §3. Among the 664,911 articles with abstracts, we sample 10% of the data as the test set and then randomly leave out another 10% documents for development. Overall, there are 4359 distinct event lexical items, at a similar scale with previous work (Chambers and Jurafsky, 2008; Do et al., 2011). The corpus statistics are summarized in Table 1.

Input: The inputs to models are the documents and the extracted events. The models are required to rank the events from the most to least salience.

Baselines: Three methods from previous researches are used as baselines: *Frequency*, *Location* and *PageRank*. The first two are often used to simulate saliency (Barzilay and Lapata, 2008; Vossen and Caselli, 2015). The *Frequency* baseline ranks events based on the count of the headword lemma; the *Location* baseline ranks events using the order of their appearances in discourse. Ties are broken randomly.

Similar to entity salience ranking with PageRank scores (Xiong et al., 2018), our *PageRank* baseline runs PageRank on a fully connected graph whose nodes are the events in documents. The edges are weighted by the embedding similarities between event pairs. We conduct supervised PageRank on this graph, using the same pairwise loss setup as in KCE. We report the best performance obtained by linearly combining *Frequency* with the scores obtained after a one-step random walk.

Evaluation Metric: Since the importance of events is on a continuous scale, the boundary between "important" and "not important" is vague. Hence we evaluate it as a ranking problem. The metrics are the precision and recall value at 1, 5 and 10 respectively. It is adequate to stop at 10 since there are less than 9 salient events per document on average (Table 1). We also report Area Under Curve (AUC). Statistical significance values are tested by permutation (randomization) test with $p < 0.05$.

Implementation Details: We pre-trained word embeddings with 128 dimensions on the whole Annotated New York Times corpus using Word2Vec (Mikolov et al., 2013). Entities are extracted using the TagMe entity linking toolkit (Ferragina and Scaiella, 2010). Words or entities that appear only once in training are replaced with special "unknown" tokens.

The hyper-parameters of the KCE kernels follow previous literature (Xiong et al., 2017). There is one exact match kernel ($\mu = 1, \sigma = 1e^{-3}$) and ten soft-match kernels evenly distributed between $(-1, 1)$, i.e. $\mu \in \{-0.9, -0.7, \ldots, 0.9\}$, with the same $\sigma = 0.1$.

The parameters of the models are optimized by Adam (Kingma and Ba, 2015), with batch size 128. The vectors of entities are initialized by the pre-trained embeddings. Event embeddings are initialized by their headword embedding.

6.2 The Event Intrusion Test: A Study

KCE is designed to estimate salience by modeling relations between discourse units. To better understand its behavior, we design the following **event intrusion test**, following the word intrusion test used to assess topic model quality (Chang et al., 2009).

Event Intrusion Test: The test will present to a model a set of events, including: the **origins**, all events from one document; the **intruders**, some events from another document. Intuitively, if events inside a document are organized around the core content, a model capturing their relations well should easily identify the intruder(s).

Specifically, we take a bag of unordered events $\{O_1, O_2, \ldots, O_p\}$, from a document O, as the origins. We insert into it intruders, events drawn from another document, I: $\{I_1, I_2, \ldots, I_q\}$. We ask a model to rank the mixed event set $M = \{O_1, I_1, O_2, I_2, \ldots\}$. We expect a model to rank the intruders I_i below the origins O_i.

Intrusion Instances: From the development set, we randomly sample 15,000 origin and intruding document pairs. To simplify the analysis, we only take documents with at least 5 salient events. The intruder events, together with the entities in the same sentences, are added to the origin document.

Metrics: AUC is used to quantify ranking quality, where events in O are positive and events in I are negative. To observe the ranking among the salient origins, we compute a separate AUC score between the intruders and the salient origins, denoted as SA-AUC. In other words, SA-AUC is the AUC score on the list with non-salient origins removed.

Experiments Details: We take the full KCE model to compute salient scores for events in the mixed event set M, which are directly used for ranking. Frequency is recounted. All other features (Table 2) are set to 0 to emphasize the relational aspects,

We experiment with two settings: 1. adding only the salient intruders. 2. adding only the non-salient intruders. Under both settings, the intruders are added one by one, allowing us to observe the score change regarding the number of intruders added. For comparison, we add a *Frequency* baseline, that directly ranks events by the Frequency feature.

7 Evaluation Results

This section presents the evaluations and analyses.

7.1 Event Salience Performance

We summarize the main results in Table 3.

Baselines: *Frequency* is the best performing baseline. Its precision at 1 and 5 are higher than 40%. *PageRank* performs worse than *Frequency* on all

Method	**P@01**		**P@05**		**P@10**		**AUC**	
`Location`	0.3555	–	0.3077	–	0.2505	–	0.5226	–
`PageRank`	0.3628	–	0.3438	–	0.3007	–	0.5866	–
`Frequency`	0.4542	–	0.4024	–	0.3445	–	0.5732	–
`LeToR`	0.4753†	+4.64%	0.4099†	+1.87%	0.3517†	+2.10%	0.6373†	+11.19%
`KCE (-EF)`	0.4420	−2.69%	0.4038	+0.34%	0.3464†	+0.54%	0.6089†	+6.23%
`KCE (-E)`	0.4861†‡	+7.01%	0.4227†‡	+5.04%	0.3603†‡	+4.58%	0.6541†‡	+14.12%
`KCE`	0.5049†‡	+11.14%	0.4277†‡	+6.29%	0.3638†‡	+5.61%	0.6557†‡	+14.41%
Method	**R@01**		**R@05**		**R@10**		**W/T/L**	
`Location`	0.0807	–	0.2671	–	0.3792	–	–/–/–	
`PageRank`	0.0758	–	0.2760	–	0.4163	–	–/–/–	
`Frequency`	0.0792	–	0.2846	–	0.4270	–	–/–/–	
`LeToR`	0.0836†	+5.61%	0.2980†	+4.70%	0.4454†	+4.31%	8037 / 48493 / 6770	
`KCE (-EF)`	0.0714	−9.77%	0.2812	−1.18%	0.4321†	+1.20%	6936 / 48811 / 7553	
`KCE (-E)`	0.0925†‡	+16.78%	0.3172†‡	+11.46%	0.4672†‡	+9.41%	11676 / 43294 / 8330	
`KCE`	0.0946†‡	+19.44%	0.3215†‡	+12.96%	0.4719†‡	+10.51%	12554 / 41461 / 9285	

Table 3: Event salience performance. (-E) and (-F) marks removing Entity information and Features from the full KCM model. The relative performance differences are computed against `Frequency`. W/T/L are the number of documents a method wins, ties, and loses compared to `Frequency`. † and ‡ mark the statistically significant improvements over `Frequency`†, `LeToR`‡ respectively.

Feature Groups	P@1	P@5	P@10	R@1	R@5	R@10	AUC
`Loc`	0.3548	0.3069	0.2497	0.0807	0.2671	0.3792	0.5226
`Frequency`	0.4536	0.4018	0.3440	0.0792	0.2846	0.4270	0.5732
+ `Loc`	0.4734	0.4097	0.3513	0.0835	0.2976	0.4436	0.6354
+ `Loc` + `Event`	0.4726	0.4101†	0.3516	0.0831	0.2969	0.4431	0.6365†
+ `Loc` + `Entity`	0.4739	0.4100	0.3518	0.0812	0.2955	0.4418	0.6374
+ `Loc` + `Entity` + `Event`	0.4739	0.4100	0.3518†	0.0832	0.2974	0.4452†	0.6374†
+ `Loc` + `Entity` + `Event` + `Local`	0.4754†	0.4100	0.3517†	0.0837	0.2981	0.4454†	0.6373†

Table 4: Feature Ablation Results. + sign indicates the additional features to `Frequency`. `Loc` is the sentence location feature. `Event` is the event voting feature. `Entity` is the entity voting feature. `Local` is the local entity voting feature. † marks the statistically significant improvements over + `Loc`.

the precision and recall metrics. *Location* performs the worst.

Feature Based: `LeToR` outperforms the baselines significantly on all metrics. Particularly, its P@1 value outperforms the *Frequency* baseline the most (4.64%), indicating a much better estimation on the most salient event. In terms of AUC, `LeToR` outperforms *Frequency* by a large margin (11.19% relative gain).

Feature Ablation: To understand the contribution of individual features, we conduct an ablation study of various feature settings in Table 4. We gradually add feature groups to the *Frequency* baseline. The combination of *Location* (sentence location) and *Frequency* almost sets the performance for the whole model. Adding each voting feature individually produces mixed results. However, adding all voting features improves all metrics. Though the margin is small, 4 of them are statistically significant over *Frequency*+*Location*.

Kernel Centrality Estimation: The `KCE` model further beats `LeToR` significantly on all metrics, by around 5% on AUC and precision values, and by around 10% on the recall values. Notably, the P@1 score is much higher, reaching 50%. The large relative gain on all the recall metrics and the high performance on precision show that `KCE` works really well on the top of the rank list.

Kernel Ablation: To understand the source of performance gain of `KCE`, we conduct an ablation study by removing its components: `-E` removes of entity kernels; `-EF` removes the entity kernels and the features. We observe a performance drop in both cases. Without entities and features, the model only using event information still performs similarly to *Frequency*. The drops are also a reflection of the small number of events (≈ 60 per document) comparing to entities (≈ 200 per document).

		Word2Vec	**KCE**
attack	kill	0.69	0.3
arrest	charge	0.53	0.3
USA (E)	war	0.46	0.3
911 attack (E)	attack	0.72	0.3
attack	trade	0.42	0.9
hotel (E)	travel	0.49	0.9
charge	murder	0.49	0.7
business (E)	increase	0.43	0.7
attack	walk	0.44	-0.3
people (E)	work	0.40	-0.3

Table 5: Similarities between event entity pairs. **Word2vec** shows the cosine similarity in pre-trained embeddings. **KCE** lists their closest kernel mean after training. (E) marks entities.

The study indicates that the relational signals and features contain different but both important information.

Discussion: The superior results of KCE demonstrate its effectiveness in predicting salience. So what additional information does it capture? We revisit the changes made by KCE: 1. it adjusts the embeddings during training. 2. it introduces weighted soft count kernels. However, the *PageRank* baseline also does embedding tuning but produces poor results, thus the second change should be crucial. We plot the learned kernel weights of KCE in Figure 2. Surprisingly, the salient decisions are not linearly related, nor even positively correlated to the weights. In fact, besides the "Exact Match" bin, the highest absolute weights actually appear at 0.3 and -0.3. This implies that embedding similarities do not directly imply salience, breaking some assumptions of the feature based model and *PageRank*.

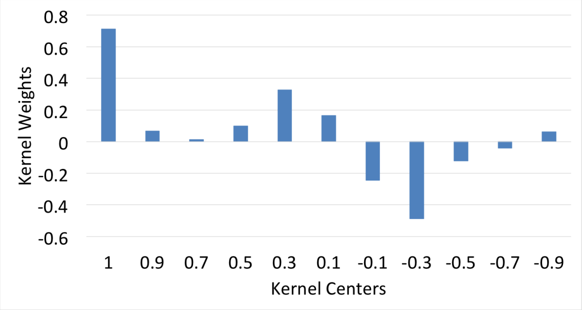

Figure 2: Learned Kernel Weights of KCE

Case Study: We inspect some pairs of events and entities in different kernels and list some examples in Table 5. The pre-trained embeddings are changed a lot. Pairs of units with different raw similarity values are now placed in the same bin. The pairs in Table 3 exhibit interesting types of relations: e.g.,"*arrest-charge*" and "*attack-kill*" form script-like chains; "911 attack" forms a quasi-identity relation (Recasens et al., 2010) with "attack"; "business" and "increase" are candidates as frame-argument structure. While these pairs have different raw cosine similarities, they are all useful in predicting salience. KCE learns to gather these relations into bins assigned with higher weights, which is not achieved by pure embedding based methods. The KCE has changed the embedding space and the scoring functions significantly from the original space after training. This partially explains why the raw voting features and PageRank are not as effective.

7.2 Intrusion Test Results

Figure 3 plots results of the intrusion test . The left figure shows the results of setting 1: adding non-salient intruders. The right one shows the results of setting 2: adding salient intruders. The AUC is 0.493 and the SA-AUC is 0.753 if all intruders are added.

The left figure shows that KCE successfully finds the non-salient intruders. The SA-AUC is higher than 0.8. Yet the AUC scores, which include the rankings of non-salience events, are rather close to random. This shows that the salient events in the origin documents form a more cohesive group, making them more robust against the intruders; the non-salient ones are not as cohesive.

In both settings, KCE produces higher SA-AUC than *Frequency* at the first 30%. However, in setting 2, KCE starts to produce lower SA-AUC than *Frequency* after 30%, then gradually drops to 0.5 (random). This phenomenon is expected since the asymmetry between origins and intruders allow KCE to distinguish them at the beginning. When all intruders are added, KCE performs worse because it relies heavily on the relations, which can be also formed by the salient intruders. This phenomenon is observed only on the salient intruders, which again confirms the cohesive relations are found among salient events.

In conclusion, we observe that the salient events form tight groups connected by discourse relations while the non-salient events are not as related. The observations imply that the main scripts in documents are mostly anchored by small groups of salient events (such as the "Trial" script in

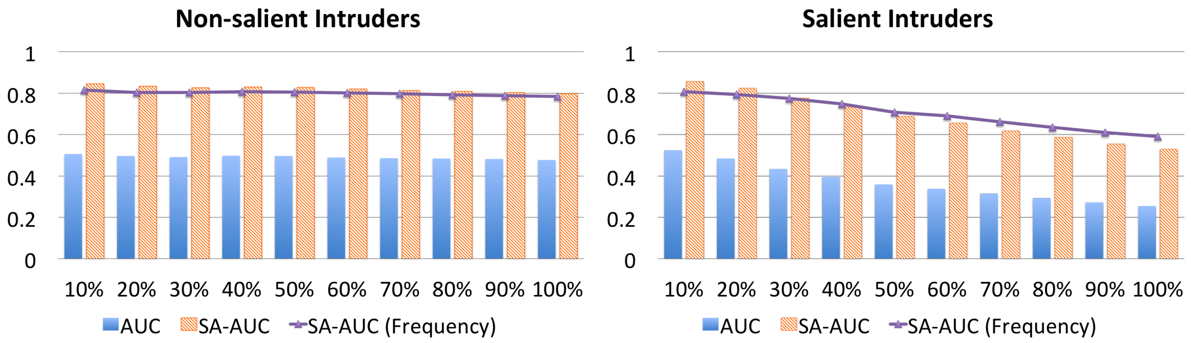

Figure 3: Intruder study results. X-axis shows the percentage of intruders inserted. Y-axis is the AUC score scale. The left and right figures are results from salient and non-salient intruders respectively. The blue bar is AUC. The orange shaded bar is SA-AUC. The line shows the SA-AUC of the frequency baseline.

Example 1). Other events may serve as "backgrounds" (Cheung et al., 2013). Similarly, Choubey et al. (2018) find that relations like event coreference and sequence are important for saliency.

8 Conclusion

We propose two salient detection models, based on lexical relatedness and semantic relations. The feature-based model with lexical similarities is effective, but cannot capture semantic relations like scripts and frames. The KCE model uses kernels and embeddings to capture these relations, thus outperforms the baselines and feature-based models significantly. All the results are tested on our newly created large-scale event salience dataset. While the automatic method inevitably introduces noises to the dataset, the scale enables us to study complex event interactions, which is infeasible via costly expert labeling.

Our case study shows that the salience model finds and utilize a variety of discourse relations: script chain (*attack* and *kill*), frame argument relation (*business* and *increase*), quasi-identity (*911 attack* and *attack*). Such complex relations are not as prominent in the raw word embedding space. The core message is that a salience detection module automatically discovers connections between salience and relations. This goes beyond prior centering analysis work that focuses on lexical and syntax and provide a new semantic view from the script and frame perspective.

In the intrusion test, we observe that the small number of salient events are forming tight connected groups. While KCE captures these relations quite effectively, it can be confused by salient intrusion events. The phenomenon indicates that the salient events are tightly connected, which form the main scripts of documents.

This paper empirically reveals many interesting connections between discourse phenomena and salience. The results also suggest that core script information may reside mostly in the salient events. Limited by the data acquisition method, this paper only models discourse salience as binary decisions. However, salience value may be continuous and may even have more than one aspects. In the future, we plan to investigate these complex settings. Another direction of study is large-scale semantic relation discovery, for example, frames and scripts, with a focus on salient discourse units.

Acknowledgement

This research was supported by DARPA grant FA8750-18-2-0018 funded under the AIDA program and National Science Foundation (NSF) grant IIS-1422676. Any opinions, findings, and conclusions in this paper are the authors and do not necessarily reflect the sponsors. We thank the anonymous reviewers whose suggestions helped clarify this paper.

References

Jun Araki, Zhengzhong Liu, Eduard Hovy, and Teruko Mitamura. 2014. Detecting Subevent Structure for Event Coreference Resolution. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 4553–4558, Reykjavik, Iceland.

CF Baker, CJ Fillmore, and JB Lowe. 1998. The berkeley framenet project. *Proceeding ACL '98 Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, pages 86–90.

Niranjan Balasubramanian, Stephen Soderland, and OE Mausam. 2013. Generating Coherent Event Schemas at Scale. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*.

Breck Baldwin and Thomas S Morton. 1998. Dynamic Coreference-Based Summarization. *Proceedings of the Third Conference on Empirical Methods in Natural Language Processing (EMNLP-3)*, pages 1–6.

Regina Barzilay and Mirella Lapata. 2008. Modeling Local Coherence: An Entity-Based Approach. In *Proceedings of the 43rd Annual Meeting of the ACL*, volume 34, pages 1–34.

Susan Windisch Brown, Claire Bonial, Leo Obrst, and Martha Palmer. 2017. The Rich Event Ontology. In *Proceedings of the Events and Stories in the News Workshop*, pages 87–97.

Nathanael Chambers, Taylor Cassidy, Bill McDowell, and Steven Bethard. 2014. Dense Event Ordering with a Multi-Pass Architecture. *Transactions of the Association for Computational Linguistics*, 2:273–284.

Nathanael Chambers and Dan Jurafsky. 2008. Unsupervised Learning of Narrative Event Chains. In *ACL '08 Meeting of the Association for Computational Linguistics*, pages 789–797.

Jonathan Chang, Sean Gerrish, Chong Wang, and David M Blei. 2009. Reading Tea Leaves: How Humans Interpret Topic Models. *Advances in Neural Information Processing Systems 22*, pages 288–296.

Pengxiang Cheng and Katrin Erk. 2018. Implicit Argument Prediction with Event Knowledge. In *NAACL 2018*, 2012.

JC Kit Cheung, H Poon, and Lucy Vanderwende. 2013. Probabilistic Frame Induction. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL/HLT 2013)*.

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 2018*.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46.

Dipanjan Das and Noah Smith. 2011. Semi-Supervised Frame-Semantic Parsing for Unknown Predicates. In *HLT '11 Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, volume 1, pages 1435–1444.

Quang Xuan Do, Yee Seng Chan, and Dan Roth. 2011. Minimally supervised event causality identification. In *EMNLP '11 Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 294–303.

Quang Xuan Do, Wei Lu, and Dan Roth. 2012. Joint Inference for Event Timeline Construction. *EMNLP-CoNLL '12 Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, (July):677–687.

M. Dojchinovski, D. Reddy, T. Kliegr, T. Vitvar, and H. Sack. 2016. Crowdsourced corpus with entity salience annotations. In *Proceedings of the 10th International Conference on Language Resources and Evaluation, LREC*, pages 3307–3311.

Jesse Dunietz and Daniel Gillick. 2014. A New Entity Salience Task with Millions of Training Examples. In *Proceedings of the European Association for Computational Linguistics*, pages 205–209.

Günes Erkan and Dragomir R Radev. 2004. LexRank : Graph-based Lexical Centrality as Salience in Text Summarization. *Journal of Artificial Intelligence Research*, 22:457–479.

Paolo Ferragina and Ugo Scaiella. 2010. Tagme: on-the-fly annotation of short text fragments (by wikipedia entities). In *CIKM 2010*.

Joseph Evans Grimes. 1975. *The Thread of Discourse*. New York.

Michael Halliday and Ruqaiya Hasan. 1976. *Cohesion in English*.

Diederik P. Kingma and Jimmy Lei Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR 2015*.

Qi Li, Heng Ji, and Liang Huang. 2013. Joint Event Extraction via Structured Prediction with Global Features. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL2013)*.

Tie-Yan Liu. 2009. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3(3):225–331.

Zhengzhong Liu, Jun Araki, Eduard Hovy, and Teruko Mitamura. 2014. Supervised Within-Document Event Coreference using Information Propagation. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 4539–4544, Reykjavik, Iceland.

Jing Lu and Vincent Ng. 2017. Joint Learning for Event Coreference Resolution. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 90–101.

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. *Proceedings of 52nd Annual Meeting of the ACL: System Demonstrations*, pages 55–60.

Daniel Marcu. 1999. Discourse Trees are Good Indicators of Importance in Text. *Advances in Automatic Text Summarization*, pages 123–136.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 2th Advances in Neural Information Processing Systems 2013 (NIPS 2013)*, pages 3111–3119.

Teruko Mitamura, Zhengzhong Liu, and Eduard Hovy. 2015. Overview of TAC KBP 2015 Event Nugget Track. In *TAC KBP 2015*, pages 1–31.

Jane Morris and Graeme Hirst. 1991. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics*, 17:21–48.

Thien Huu Nguyen and Ralph Grishman. 2015. Event Detection and Domain Adaptation with Convolutional Neural Networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 365–371.

Haoruo Peng, Yangqi Song, and Dan Roth. 2016. Event Detection and Co-reference with Minimal Supervision. In *EMNLP 2016*.

Karl Pichotta and Raymond J. Mooney. 2016. Using Sentence-Level LSTM Language Models for Script Inference. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 279–289.

Marco Ponza, Paolo Ferragina, and Francesco Piccinno. 2018. SWAT: A System for Detecting Salient Wikipedia Entities in Texts.

James Pustejovsky, Patrick Hanks, Roser Saurí, Andrew See, Robert Gaizauskas, Andrea Setzer, Beth Sundheim, David Day, Lisa Ferro, and Dragomir. 2002. The TIMEBANK Corpus. *Natural Language Processing and Information Systems*, 4592:647–656.

Marta Recasens, Matthew Can, and Daniel Jurafsky. 2013. Same Referent, Different Words: Unsupervised Mining of Opaque Coreferent Mentions. *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, (June):897–906.

Marta Recasens, Eduard Hovy, and M. Antònia Martí. 2010. A Typology of Near-Identity Relations for Coreference (NIDENT). *7th International Conference on Language Resources and Evaluation (LREC-2010)*, (i):149–156.

Rachel Rudinger, Vera Demberg, Ashutosh Modi, Benjamin Van Durme, and Manfred Pinkal. 2015. Learning to predict script events from domain-specific text. In *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics*, pages 205–210.

Evan Sandhaus. 2008. The New York Times Annotated Corpus.

Roger C Schank and Robert P Abelson. 1977. *Scripts, Plans, Goals and Understanding*. Lawrence Erlbaum Associates.

E Skorochod'ko. 1971. Adaptive Method of Automatic Abstracting and Indexing. In *Proceedings of the IFIP Congress 71*.

Zeno Vendler. 1957. Verbs and times. *The Philosophical Review*, 66(2):143–160.

Piek Vossen and Tommaso Caselli. 2015. Storylines for structuring massive streams of news. In *Proceedings of the First Workshop on Computing News Story Lines*, pages 40–49.

Chenyan Xiong, Zhengzhong Liu, Jamie Callan, and Tie-Yan Liu. 2018. Towards Better Text Understanding and Retrieval through Kernel Entity Salience Modeling. In *SIGIR 2018*.

Chenyang Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. 2017. End-to-End Neural Ad-hoc Ranking with Kernel Pooling. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 55–64.

Congle Zhang, Stephen Soderland, and Daniel S Weld. 2015. Exploiting Parallel News Streams for Unsupervised Event Extraction. volume 3, pages 117–129.