Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 1747–1752
Marseille, 11–16 May 2020

Towards Few-Shot Event Mention Retrieval: An Evaluation Framework and A Siamese Network Approach

Bonan Min, Yee Seng Chan, Lingjun Zhao
Raytheon BBN Technologies
10 Moulton Street, Cambridge, MA 02138
{bonan.min, yeeseng.chan, lingjun.zhao}@raytheon.com

Abstract

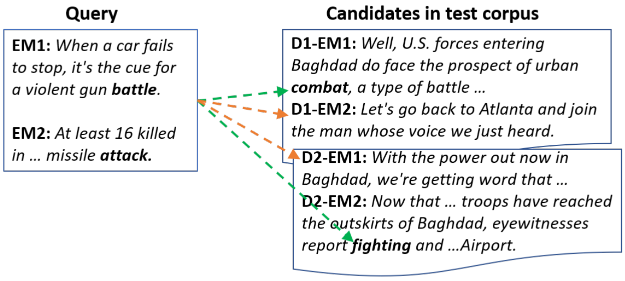
Automatically analyzing events in a large amount of text is crucial for situation awareness and decision making. Previous approaches treat event extraction as "one size fits all" with an ontology defined a priori. The resulted extraction models are built just for extracting those types in the ontology. These approaches cannot be easily adapted to new event types nor new domains of interest. To accommodate personalized event-centric information needs, this paper introduces the few-shot Event Mention Retrieval (EMR) task: given a user-supplied query consisting of a handful of event mentions, return relevant event mentions found in a corpus. This formulation enables "query by example", which drastically lowers the bar of specifying event-centric information needs. The retrieval setting also enables fuzzy search. We present an evaluation framework leveraging existing event datasets such as ACE. We also develop a Siamese Network approach, and show that it performs better than ad-hoc retrieval models in the few-shot EMR setting.

Keywords: Information Extraction, Information Retrieval, Evaluation Methodologies

1. Introduction

Tracking events are vital for situation awareness and decision making. The past two decades have witnessed an exponential growth of unstructured text, in which events are abundant. It is difficult, if not impossible, for a human user to keep up with the extremely high volume of events emerging every day.

Event extraction comes to rescue. It aims at automatically extracting mentions of events [1] from text. However, previous tasks, e.g., MUC (Grishman and Sundheim, 1996), ACE (Doddington et al., 2004b), and TAC-KBP (Freedman and Gabbard, 2014; Mitamura et al., 2015), treat event extraction with "one size fits all" approaches, in which an event ontology needs to be defined a priori, and then event extractors are built just for the types in the ontology. These approaches often require extensive expertise from domain experts and linguists, in order to define event types rigorously [2]. It will not work for personalized information needs that we frequently encounter nowadays. For instance, the ontology might be focused on socio-political events (e.g., *Conflict* and *ProvideAid*), but a user might be interested in events related to corporate finance (e.g., *Merge* and *Acquisition*), disease outbreaks, or sports. Furthermore, there are often disagreements among even experts on what they mean by certain event types (e.g., whether a verbal conflict counts as a *Conflict* event) and the level of granularity of interest (*ProvideAid* or more specifically, *ProvideMilitaryAid*).



EM1: *When a car fails to stop, it's the cue for a violent gun **battle**.*

EM2: *At least 16 killed in ... missile **attack.***

D1-EM1: *Well, U.S. forces entering Baghdad do face the prospect of urban **combat**, a type of battle ...*
D1-EM2: *Let's go back to Atlanta and join the man whose voice we just heard.*

D2-EM1: *With the power out now in Baghdad, we're getting word that ...*
D2-EM2: *Now that ... troops have reached the outskirts of Baghdad, eyewitnesses report **fighting** and ...Airport.*

Figure 1: An example illustrating the few-shot EMR task setting. The green dashed lines with arrows indicate that a candidate event mention is relevant to the query, while the orange dashed lines indicate that the candidate is not related to the query.

To satisfy the ever-increasing on-demand, personalized event-centric information needs, we introduce the few-shot Event Mention Retrieval (EMR) task: given a query consists of a handful of event mentions, return all relevant event mentions in a corpus. An example is illustrated in Figure 1, in which a user supplies a query consisting of two event mentions (EM) EM1 and EM2, the system is expected to return all relevant event mentions (D1-EM1 in document D1 and D2-EM2 in document D2) among event mentions in this two-document corpus. In this example, the event mentions are relevant if they share the same event type (e.g., the type *Attack* as defined by the two event mentions in the query).

This formulation has two advantages over previous extraction-based approaches:

- It enables a **query-by-example** paradigm, in which a user is only required to define new types of events via specifying a handful of examples. This drastically lowers the bar for specifying a diverse range of personalized event-centric information needs.

- In contrast to the rigid classification setting, this allows **fuzzy search**: it allows *relevant* instances but not nec-

[1] An event mention (EM) is an event with surrounding context (text), that are often triggered by a key word or phrase. An example is *John **attacked** a bear*.

[2] For example, ACE defines 33 event types with a 77-page guideline in order to achieve high inter-annotator agreement. The ACE guideline is available at `https://www.ldc.upenn.edu/sites/www.ldc.upenn.edu/files/english-events-guidelines-v5.4.3.pdf`.

essarily *exact matches* to be returned. Although in this paper we define a pair of event mention being relevant if they share the same event type, the EMR setting opens the door to many variations of relevancy such as "same-event" (event coreference) and "same-actor/location" (e.g., what are *Taliban*'s movements/actions in *Iran*?).

To evaluate few-shot EMR, we present an information-retrieval-based evaluation framework, leveraging existing event annotation datasets such as the event annotation corpus from ACE. We will describe details in Section 3. In addition, we develop two approaches for EMR: an ad-hoc retrieval model, and a trainable Siamese Network (Koch et al., 2015) model. We demonstrate that the Siamese Network model performs better well than ad-hoc retrieval, especially under the few-shot EMR setting.

In this paper, we first describe related work, and then define the EMR task and present the two models we developed for EMR. We will then present experiment results and conclusion.

2. Related Work

Event extraction and tracking. Event extraction is often formulated as a two-stage (Ahn, 2006) classification (trigger classification then argument identification) problem. Prior works either use high-level features (Huang and Riloff, 2012; Ji and Grishman, 2008) or are Neural Network models (Chen et al., 2015). In need of labeled datasets for training models and evaluation, datasets such as MUC (Grishman and Sundheim, 1996), ACE (Doddington et al., 2004a), ERE (Song et al., 2015), TAC-KBP events (Freedman and Gabbard, 2014; Mitamura et al., 2015) and Situation Frames (Strassel et al., 2017) have been developed. There are also datasets created for specialized domains. An example is the GENIA biomedical event annotation (Kim et al., 2008). In event extraction using limited training data, (Nguyen et al., 2016) proposed a two-stage NN model for event type extension. Given a new event type with a small set of seed examples, they leverage examples from other event types. (Peng et al., 2016) developed a minimally supervised approach to event trigger extraction by leveraging trigger examples gathered from the ACE annotation guidelines.

Our EMR formulation differs from the extraction setting: the information retrieval setting only requires returning a ranked list of related event mentions, thus facilitating fuzzy search. This will enable a broad range of new applications. A related task is the Topic Detection and Tracking (TDT) Evaluation (Allan, 2012). TDT aims at detecting the appearance of new event-like topics and tracking their reappearance and evolution. The topic may be aggregated at one or more documents. In contrast, our work focuses on event mentions, which are instances of events triggered by words and phrases.

Our work is also related to the Coreferent Mention Retrieval (Sankepally et al., 2018) task, in which the goal is to return coreferent entity mentions given a query mention.

Deep contextualized language models Recently, deep language models such as BERT (Devlin et al., 2019) have been shown to be useful for event extraction (Yang et al., 2019) and Information Retrieval (IR) (Dai and Callan, 2019), because (1) contextualized word embeddings, generated by BERT trained with large corpora, captured word meanings that reflects its context, (2) BERT is shown to capture syntactic and semantic information (Tenney et al., 2019; Clark et al., 2019; Jawahar et al., 2019) that may be useful for modeling IR.

3. The EMR Task Definition, A Dataset and An Evaluation Framework

Task definition: Let there be m event types and a corpus $\mathcal{C}$ of n event mentions per each event type and n' $None$ event mentions that do not belong to any of the m types of interest. We define the k-shot m-way Event Mention Retrieval (EMR) task as follows: given a query consists of k event mention per type (a.k.a., k-shot), find the n event mentions, that share the same event type as defined by the query, among the $m \cdot n + n'$ candidate event mentions in $\mathcal{C}$. In other words, the goal is to return a ranked list of all event mentions in $\mathcal{C}$, such that the relevant n event mentions are ranked higher than the rest in the set.

Dataset To evaluate EMR, we need a dataset which contains ground truth annotation on whether two event mentions has the same event type. We use the ACE event dataset (Doddington et al., 2004b) [3], which consists of 599 documents with event mentions of 33 event types exhaustively annotated by hand. To construct a test dataset (described as $\mathcal{C}$ above) for evaluating system performance, we randomly sample $n = 50$ event mentions per each event type and an additional $n' = 2000$ event mentions of the $None$ class [4].

To make sure we have sufficient amount of event mentions per event type for training, development and test, we filter the event types that have fewer than 100 event mentions. This results in 15 event types (the full list is in Table 1). We choose a threshold of 100 so that we have 50 event mentions to be used in the test dataset, a maximum of 10 event mentions to be used in a query (a.k.a., a maximum of 10-shot), and the remaining 40 event mentions to be used in the development dataset.

Evaluation framework We evaluate a EMR system S with the k-shot m-way retrieval setting: given a query (consists of k event mentions), the goal of S is to rank the $m \cdot n + n'$ candidate event mentions in the test dataset $\mathcal{C}$, according to whether they share the same event type with the query. S can use the k event mention as a query in an ad-hoc retrieval setting, or additionally for training a model such as the Siamese Network model to be described in Section 4.2.

Metric: We use Mean Average Precision (MAP) as the metric. Given a set of queries Q evenly distributed among m event types, the task is to rank all candidate event mentions among which R is the subset of the relevant event mentions. MAP is defined as

$$\frac{1}{|Q|} \sum_{q \in Q} \frac{\sum_r P@r}{|R|}$$

[3] Our approach is also applicable to other event datasets that contain mention-level event annotation.

[4] The ACE documents are exhaustively labeled with events for the 33 types, so verb or noun mentions without labels belong to the $None$ class.

in which $P@r$ is the precision of the top-r-ranked event mentions.

4. Event Mention Retrieval Models

We developed two approaches for EMR. Both approach output a relevancy score $r(e, e')$, given a pair of event mentions e and e' as input. Therefore, the final relevancy score for a candidate $e' \in \mathcal{C}$ given query q (consists of k event mentions) is $r(q, e') = \frac{1}{k}\sum_{e_i \in q} r(e_i, e')$. All candidate event mentions in $\mathcal{C}$ are ranked according to its relevancy score $r(q, e')$ to the query q. In this section, we focus on describing the two approaches in terms of how each approach produces $r(e, e')$ for a pair of event mentions e and e'.

The first approach is an ad-hoc retrieval model that simply calculates a similarity score for a pair of event mentions using their continuous, dense vector representations, and then use the similarity score as the relevancy score. The second approach is a trainable Siamese Network model which takes a pair of event mentions as input and predicts how likely they are relevant.

Representing event mentions with BERT For both approaches, we apply BERT (Devlin et al., 2019) to each sentence where the event mention appears to generate contextualize word ebmeddings. Prior work (Jawahar et al., 2019) shows that different layers in BERT capture different information, e.g., word meaning, syntactic dependency, that are useful for event extraction. Let $W_{-4}(x)$,... $W_{-1}(x)$ be the last 4 layers of BERT that produce 4 vectors for a word x in $\mathbb{R}^d$. To capture the syntactic and semantic information generated by BERT for x, we concatenate its BERT representation for all last 4 layers into a vector $W(x)$ in $\mathbb{R}^{4d}$: $W(x) = [W_{-4}(x), W_{-3}(x), W_{-2}(x), W_{-1}(x)]$. We represent an event mention X using a window of words $x_{-2}, x_{-1}, x_0, x_1, x_2$ around the trigger word x_0, and then generated its representation $W(X) = W(x_{-2}), W(x_{-1}), W(x_0), W(x_1), W(x_2)$. This BERT represenation lookup process is illustrated in the bottom half of Figure 2.

4.1. Model 1: Ad-hoc BERT-based EMR model

Given a pair of event mentions X_1 and X_2 as input, the ad-hoc BERT-based model calculates the relevancy score in the following steps:

- It first generates BERT-based representation $W(X_1)$ and $W(X_2)$ for X_1 and X_2 respectively.
- It then calculates the relevancy as $r_1(X_1, X_2) = cosine(W(X_1), W(X_2))$

This approach calculates the cosine similarity of the BERT representations for the pair of event mentions, and use that as the relevancy score.

4.2. Model 2: A Siamese Network EMR Model

The ad-hoc retrieval model does not take advantage of the relationships between event mentions in the query (i.e., they have the same type), nor the relationships between an event mention in the query and the rest of the event mentions, i.e., with high probability, they do not share the same type with the query, given the sparsity of events.

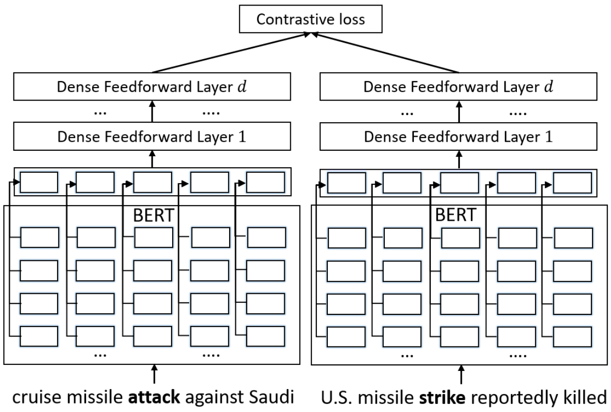

Contrastive loss
Dense Feedforward Layer d
... .
Dense Feedforward Layer 1
BERT
... .
Dense Feedforward Layer d
... .
Dense Feedforward Layer 1
BERT
... .
cruise missile attack against Saudi
U.S. missile strike reportedly killed

Figure 2: A Siamese network for EMR. Two event mentions (left and right) are shown as the input. We pass them into a BERT lookup layer, and then a sequence of densely-connected, feedforward layers, before feeding their learnt representations into the contrastive loss function. The BERT and feedforward layers on the left and right share exactly the same architecture and parameter weights, following the Siamese Network design (Koch et al., 2015).

Based on this observation, we develop a Siamese Network model for EMR, that is trained from event pairs with automatically-generated *same class(1)* and *not in same class (0)* labels. Figure 2 shows the model architecture.

Similar to the ad-hoc retrieval model and as illustrated in Figure 2, it first generates BERT representation $W(X_1)$ and $W(X_1)$ for X_1 and X_2, respectively. It then passes the BERT representations into a sequence of d layers of densely connected feedforward layers $f_1, ..., f_d$ to generate hidden representation $F(X) = f_d(f_{d-1}(...(f_1(W(X)))))$ for X_1 and X_2.

Then the Siamese Network calculates the cosine similarity:

$$S(X_1, X_2) = \frac{< F(X_1), F(X_2) >}{||F(X_1)||\,||F(X_2)||}$$

and then passes it into a contrastive loss function, which uses the *same class* ($y = 1$) or *not in same class* ($y = 0$) as the training signal:

$$L(X_1, X_2, y) = y \cdot \max(1 - S(X_1, X_2), 0)^2$$
$$+(1 - y)S(X_1, X_2)^2$$

The training objective function is to minimize the total loss over a training dataset $D = \{X_1^i, X_2^i, y^i\}$ of N pairs:

$$\mathcal{L}(D) = \sum_{i=1}^{N} L(X_1^i, X_2^i, y^i)$$

Generating labeled training examples: To generate event mention pairs with the *same class* and *not in same class* labels for training the Siamese Network model in the k-shot setting, we randomly sample pairs as follows [5]:

[5] We sample up to 200 *same class* pairs. We sample up to 1000000 *not in same class* pairs to avoid computational inefficiency.

- Sample pairs that are both in the query, and assign them the *same class* label.
- Sample pairs such that one of them is in the query but the other is not, and assign this pair the *not in same class* label.

5. Experiments

As described in Section 3., we use a dataset constructed from the ACE event annotated corpus for evaluation. We evaluate the following models:

- **Ad-hoc BERT-based model**: the ad-hoc BERT-based EMR model described in Section 4.1.
- **BERT+Siamese Network**: This is the Siamese Network model which also uses BERT as input representation. The model is described in Section 4.2.

Experimental settings: We evaluate the models with the k-shot m-way EMR setting, described in Section 3. For each event type, we generated 10 queries, each consists of k event mentions (a.k.a., k-shot), and run the retrieval experiments for 10 times and take an average on the MAP scores reported. For the BERT+Siamese Networks model, the k event mentions are used to generate labeled pairs for training the Siamese Network model, as well as the query for ranking mentions in the test set. We experimented with varying k values, ranging from $k = 2$ to 10.

For training the Siamese Network, we performed grid search over the parameter space for hyper-parameters, using a held-out development dataset (also generated from ACE). In the final experiments, we used the Adam optimizer with a learning rate of $5e^{-5}$, a batch size of 50, and an epoch number of 50. We also tuned the number of densely connected layers and numbers of hidden units, and we found that using 3 layers with 768 hidden units each performs the best on the development set.

5.1. Experimental Results

Overall performance. Figure 3 shows MAP scores of k-shot retrieval for $k = 2, 3, 4, ..., 10$ for both models. It shows increasing performance for the Siamese Network model, when k increases. In contrast, the ad-hoc retrieval model does not increase when k increases. The Siamese Network model performs increasingly better than the ad-hoc model when k gets higher, as it is able to leverage more automatically labeled event pairs as k increases. With limited observation of event mentions (as little as 5), both the ad-hoc model and the Siamese model model are still effective. At k as little as 5, the Siamese Network model outperforms the ad-hoc model by 20 points in MAP.

Retrieval performance by event type Table 1 shows the k-shot ($k = 2, 3, 4, 5, 10$) MAP scores by event type for the Siamese Network model. In general, a similar trend is observed: retrieval performances are significantly improved when more event mentions are used in each query; $k = 5$ consistently show decent results across most event types. Results vary by event type since some events are expressed by a small set of trigger words (e.g., *"elect"* for the type *Personnel.Elect*) but others may have a much larger variation in how they are expressed (e.g., *Conflict.Attack*, and *Personnel.Start/End-Position*).

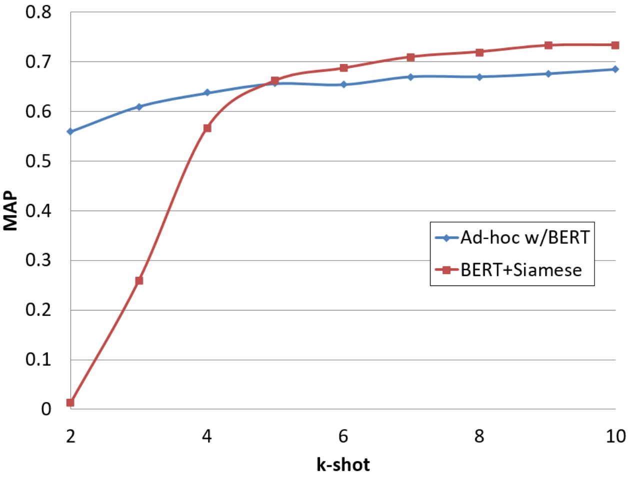


Figure 3: Retrieval performance of the ad-hoc EMR model (Ad-hoc w/ BERT) and the BERT+Siamese Network (BERT+Siamese) models. The scores are MAP in k-shot ($k = 2, 3, 4, ..., 10$) settings in which k event mentions are used in each query. The scores are averaged over 10 queries.

Event type	k**-shot**				
	2	3	4	5	10
Conflict.Attack	0.014	0.13	0.386	0.508	0.615
Contact.Meet	0.015	0.311	0.714	0.785	0.822
Contact.Phone-Write	0.018	0.405	0.575	0.612	0.623
Justice.Arrest-Jail	0.01	0.037	0.43	0.671	0.799
Justice.Charge-Indict	0.01	0.33	0.768	0.884	0.883
Justice.Sentence	0.01	0.3259	0.746	0.841	0.88
Justice.Trial-Hearing	0.01	0.333	0.745	0.942	0.941
Life.Die	0.012	0.259	0.714	0.746	0.842
Life.Injure	0.014	0.394	0.712	0.665	0.794
Movement.Transport	0.017	0.051	0.237	0.365	0.5
Personnel.Elect	0.014	0.76	0.842	0.828	0.825
Personnel.End-Position	0.014	0.177	0.499	0.629	0.802
Personnel.Start-Position	0.015	0.087	0.304	0.394	0.596
Txn.Transfer-Money	0.015	0.19	0.443	0.517	0.55
Txn.Transfer-Ownership	0.012	0.076	0.389	0.525	0.533

Table 1: k-shot retrieval performance by event type for the Siamese Network model. "Txn" is short for "Transaction". The scores are averaged over 10 queries.

5.2. Analysis and Discussion

We further inspect the retrieved event mentions for each query. Table 2 shows some interesting examples, which are among the top-20 retrieved event mentions with the highest relevancy score for each query type. Most of them have a human-annotated ACE label different from the corresponding query event type. The mismatch between query event type and the ACE event labels of these top-ranked event mentions can be divided into three categories:

Some ACE event types are broadly defined: for example, a query for the event type *Transaction.Transfer-Ownership* contains both *U.S. forces drove though portions of the Iraqi capital, **seizing** Iraqi tanks...* and *U.S. special forces and Kurdish militiamen **captured** the town of Bardarash and...*. These two event mentions also expressed (or at lease, are very similar to) another event type *Conflict.Attack*. This

ID	Query Type	ACE Label	Event Mention
1	Movement.Transport	None	*...we can squeeze in running by Christmas Eve on the **way** to my parents...*
2	Movement.Transport	Txn.TO	*...an estimated 3,000 troops. ... they also **took** a prison where they found...*
3	Movement.Transport	None	*...the traveler said. "I was just **driving** by and looking at all your pigs,...*
4	Contact.Phone-Write	None	*...hypocrite would be a word present in many **e-mails**. hypocrite. barbara,...*
5	Contact.Phone-Write	None	*...was in better spirits as he departed for the airport with a final **message**.*
6	Conflict.Attack	Conflict.Attack	*...the power grid. Tracer rounds **lit** the night sky and artillery boomed...*
7	Conflict.Attack	Life.Die	*...minimize civilian casualties in the current Iraq **war**, at least 130 Iraqi...*
8	Txn.TO	None	*...apparently they're for **sale**. we'll have to see about that.*
9	Txn.TO	None	*...have committed to **sell** their shares to Barclays.*
10	Txn.TO	Conflict.Attack	*...on a street in Fallujah during the U.S. **assault** on...*

Table 2: Interesting examples in top-20 retrieved event mentions by query event type. The "ACE Label" shows the event type of the event mention, that are annotated in the ACE dataset. "Txn.TO" is short for "Transaction.Transfer-Ownership".

broad definition of *Transaction.Transfer-Owner* leads to a relatively lower k-shot MAP score (especially when k is small) comparing to other event types, as shown in Table 1. This also led to retrieving a *Conflict.Attack* event mention (shown as the event mention #10 in Table 2.

The model discovered missing annotation: Some retrieved top-ranked event mentions are relevant to the query, but they are missed by ACE annotation. Examples #1, 3, 4, 5, 7, 8, 9 are all in this category. Retrieving these examples requires the model to understand wider context. This shows the effectiveness of the Siamese Network model.

Ambiguous triggers require better modeling of context By applying BERT over the context window around each event trigger, our model captures context for event mentions. For example in event mention #6, *lit* is an ambiguous verb, but the model correctly retrieved it for *Conflict.Attack*. However, in event mention #2, the model fails to capture the context information well enough for predicting *took* correctly. This requires better modeling of context, which we will explore as future work.

6. Conclusion and Future Work

This paper introduces the few-shot Event Mention Retrieval (EMR) task: its task definition, an evaluation framework and a test dataset. We also develop two approaches for EMR, and present experiments that show the Siamese Network approach performs better than ad-hoc retrieval.

Our next step is to broaden the definition of relevancy by introducing a sub-task of retrieving coreferent event mentions, given a query describing a specific event.

7. Acknowledgements

This research is based upon work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via IARPA Contract No. 2019-19051600006, and DARPA/I2O and U.S. Army Research Office Contract No. W911NF-18-C-0003 under the World Modelers program. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, the Department of Defense, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein. This document does not contain technology or technical data controlled under either the U.S. International Traffic in Arms Regulations or the U.S. Export Administration Regulations.

8. Bibliographical References

Ahn, D. (2006). The stages of event extraction. In ARTE, pages 1–8, Sydney, Australia, July. Association for Computational Linguistics.

Allan, J. (2012). Topic detection and tracking: event-based information organization, volume 12. Springer Science & Business Media.

Chen, Y., Xu, L., Liu, K., Zeng, D., and Zhao, J. (2015). Event extraction via dynamic multi-pooling convolutional neural networks. In ACL-IJCNLP2-2015, pages 167–176, Beijing, China, July. Association for Computational Linguistics.

Clark, K., Khandelwal, U., Levy, O., and Manning, C. D. (2019). What does bert look at? an analysis of bert's attention. *arXiv preprint arXiv:1906.04341.*

Dai, Z. and Callan, J. (2019). Deeper text understanding for ir with contextual neural language modeling. *arXiv preprint arXiv:1905.09217.*

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (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.

Doddington, G. R., Mitchell, A., Przybocki, M. A., Ramshaw, L. A., Strassel, S. M., and Weischedel, R. M. (2004a). The automatic content extraction (ace) program - tasks, data, and evaluation. In LREC. European Language Resources Association.

Doddington, G. R., Mitchell, A., Przybocki, M. A., Ramshaw, L. A., Strassel, S. M., and Weischedel, R. M. (2004b). The automatic content extraction (ace) program-tasks, data, and evaluation. In Lrec, volume 2, page 1. Lisbon.

Freedman, M. and Gabbard, R. (2014). Overview of the event argument evaluation. In Proceedings of TAC KBP 2014 Workshop, National Institute of Standards and Technology, pages 17–18.

Grishman, R. and Sundheim, B. (1996). Message understanding conference-6: A brief history. In COLING 1996

Volume 1: The 16th International Conference on Computational Linguistics.

Huang, R. and Riloff, E. (2012). Modeling textual cohesion for event extraction. In AAAI-CAI, AAAI'12, pages 1664–1670. AAAI Press.

Jawahar, G., Sagot, B., Seddah, D., Unicomb, S., Iñiguez, G., Karsai, M., Léo, Y., Karsai, M., Sarraute, C., Fleury, É., et al. (2019). What does bert learn about the structure of language? In 57th Annual Meeting of the Association for Computational Linguistics (ACL), Florence, Italy.

Ji, H. and Grishman, R. (2008). Refining event extraction through cross-document inference. In ACL-HLT-2008, pages 254–262, Columbus, Ohio, June. Association for Computational Linguistics.

Kim, J.-D., Ohta, T., and Tsujii, J. (2008). Corpus annotation for mining biomedical events from literature. *BMC bioinformatics*, 9(1):10.

Koch, G., Zemel, R., and Salakhutdinov, R. (2015). Siamese neural networks for one-shot image recognition. In ICML deep learning workshop, volume 2.

Mitamura, T., Liu, Z., and Hovy, E. H. (2015). Overview of tac kbp 2015 event nugget track. In TAC.

Nguyen, T. H., Fu, L., Cho, K., and Grishman, R. (2016). A two-stage approach for extending event detection to new types via neural networks. In Proceedings of the 1st Workshop on Representation Learning for NLP, pages 158–165, Berlin, Germany, August. Association for Computational Linguistics.

Peng, H., Song, Y., and Roth, D. (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.

Sankepally, R., Chen, T., Van Durme, B., and Oard, D. W. (2018). A test collection for coreferent mention retrieval. In The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, pages 1209–1212. ACM.

Song, Z., Bies, A., Strassel, S., Riese, T., Mott, J., Ellis, J., Wright, J., Kulick, S., Ryant, N., and Ma, X. (2015). From light to rich ere: annotation of entities, relations, and events. In Proceedings of the the 3rd Workshop on EVENTS: Definition, Detection, Coreference, and Representation, pages 89–98.

Strassel, S. M., Bies, A., and Tracey, J. (2017). Situational awareness for low resource languages: the lorelei situation frame annotation task. In SMERP@ ECIR, pages 32–41.

Tenney, I., Das, D., and Pavlick, E. (2019). Bert rediscovers the classical nlp pipeline. *arXiv preprint arXiv:1905.05950*.

Yang, S., Feng, D., Qiao, L., Kan, Z., and Li, D. (2019). Exploring pre-trained language models for event extraction and generation. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 5284–5294.