Extracting Events from Industrial Incident Reports

Nitin Ramrakhiyani Swapnil Hingmire Sangameshwar Patil
Alok Kumar Girish K. Palshikar
{nitin.ramrakhiyani, swapnil.hingmire, sangameshwar.patil}@tcs.com
{k.alok9@tcs.com,gk.palshikar}@tcs.com
TCS Research, India

Abstract

Incidents in industries have huge social and political impact and minimizing the consequent damage has been a high priority. However, automated analysis of repositories of incident reports has remained a challenge. In this paper, we focus on automatically extracting events from incident reports. Due to absence of event annotated datasets for industrial incidents we employ a transfer learning based approach which is shown to outperform several baselines. We further provide detailed analysis regarding effect of increase in pre-training data and provide explainability of why pre-training improves the performance.

1 Introduction

The industrial revolution[1] has had a profound effect on the socio-political fabric of the world. Economic progress of societies has been highly correlated with their degree of industrialization. However, one of the flip sides of this progress has been the cost of large industrial accidents in terms of injuries to workers, damage to material and property as well as the irreparable loss of innocent human lives. Such major industrial incidents have had large social and political impacts and have prompted policy makers to devise multiple regulations towards prevention of such incidents. As an instance, the huge social uproar after the Bhopal Gas Leakage tragedy[2] had many political ramifications and resulted in creation of many new acts, rules and institutions in India and internationally.

Governmental agencies in-charge of industrial safety (OSHA; MINERVA) as well as the industrial enterprises themselves try and minimize the possibility of recurrence of industrial incidents. For this purpose, they carry out detailed investigations of incidents that have previously occurred to identify root causes and suggest preventive actions. In most cases, reports summarizing the incidents as well as their investigation are maintained in incident document repositories[3]. For example, Table 1 shows a sample incident report summary in the construction domain.

On February 1, 2014, at approximately 11:37 a.m., a 340 ft.-high guyed telecommunication tower, suddenly **collapsed** during **upgrading activities**. Four employees were **working** on the tower **removing** its diagonals. In the process, no temporary supports were installed. As a result of the tower 's **collapse** , two employees were **killed** and two others were badly **injured**.

Table 1: Sample Incident Report summary from Construction Domain

However, most of these investigative studies are carried out manually. There is little work towards automated processing of repositories of incident reports. Automated processing of incident reports requires us to solve multiple sub-problems such as identification of domain-specific entities, events, different states or conditions, relations between the events, resolving coreferences etc. As an example, we show the entities, events and states marked in red, blue and green respectively in Table 1. In this paper, we focus on an important stage from the above pipeline - extraction of events from incident reports. Event identification is central to the automated processing of incident reports because they pithily capture what exactly happened during an incident. Identification of events is also an important task required for down the line applications such as narrative understanding and visualization through knowledge representations such as Message Se-

[1] https://en.wikipedia.org/wiki/Industrial_Revolution

[2] https://en.wikipedia.org/wiki/Bhopal_disaster

[3] https://www.osha.gov/data

Proceedings of the 4th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE) , pages 58–67
August 5–6, 2021, ©2021 Association for Computational Linguistics

quence Charts (MSC)(Palshikar et al., 2019; Hingmire et al., 2020) and event timelines(Bedi et al., 2017). Further, most of the work in event detection has focused on events in general domain such as ACE (Linguistic Data Consortium, 2005) and ECB (Bejan and Harabagiu, 2010). Little attention has been paid in the literature towards automated event extraction and analysis from industrial incident reports. To the best of our knowledge, there is no dataset of incident reports comprising of annotations for event identification (spans and attributes). This motivates us to experiment with unsupervised or weakly supervised approaches. In addition to experimenting with unsupervised baselines, we propose a transfer learning approach to extract events which first learns the nature of events in general domain through pre-training and then requires post-training with minimal training data in the domain of incidents.

We consider incident reports from two industries - civil aviation and construction and focus on identifying events involving risk-prone machinery or vehicles, common causes, human injuries and casualties and remedial measures, if any. We show that on both domains, the proposed transfer learning based approach outperforms several unsupervised and weakly supervised baselines. We further supplement the results with detailed analysis regarding effect of increase in pre-training data and explainability of pre-training through a novel clustering based approach.

We discuss relevant related work in Section 2. In Section 3, we cover the event extraction process detailing the annotation guidelines and proposed approach. In Section 4, we explain the experimental setup, evaluation and analysis. We finally conclude in Section 5.

2 Related Work

This section discusses important related work on two important aspects - automated analysis of textual incident reports/descriptions and unsupervised or weakly supervised event extraction approaches. As per the best of our knowledge, this is the first work on labelling and predicting events (a token level object) from incident report text. However, there are multiple papers which analyze incident reports at the document or sentence level for various tasks such as classification, cause-effect extraction and incident similarity. Tanguy et al.(2016) use NLP techniques to analyze aviation safety reports. The authors focus on classification of reports into different categories as well as use probabilistic topic models to analyze different aspects of incidents. The authors also propose the *timePlot* system to identify similar incident reports. Similar to (Tanguy et al., 2016), (Pence et al., 2020) perform text classification of event reports in nuclear power plants. However, both (Tanguy et al., 2016) and (Pence et al., 2020) do not focus on extraction of specific events from incident reports. Dasgupta et al. (2018) use neural network techniques to extract occupational health and safety related information from News articles related to industrial incidents. Specifically, they focus on extraction of target organization, safety issues, geographical location of the incident and penalty mentioned in the article.

In the context of event extraction approaches, multiple state-of-the-art supervised approaches have been proposed in the literature recently. However, the complex neural network architectures demand significant amounts of training data which is not available in the current scenario of event extraction in incident reports. Hence, we discuss two event extraction approaches which are weakly supervised in nature. In (Palshikar et al., 2019), the authors propose a rule based approach which considers all past tense verbs as events with a WordNet based filter retaining only “action” or “communication” events. There is no support for extraction of nominal events proposed by the authors. (Araki and Mitamura, 2018) propose an Open Domain Event Extraction approach which uses linguistic resources like WordNet and Wikipedia to generate training data in a distantly supervised manner and then train a BiLSTM based supervised event detection model using this data. Wang et al.(2019) propose a weakly supervised approach for event detection. The authors first construct a large-scale event-related candidate set and then use an adversarial training mechanism to identify events. We use the first two approaches - (Palshikar et al., 2019) and (Araki and Mitamura, 2018) as our baselines and discuss them in detail in Section 4. The third approach (Wang et al., 2019) based on adversarial training is evaluated on closed-domain datasets and hence it would be difficult to tune it and use it as a baseline for an open-domain event extraction task like ours.

3 Event Extraction in Incident Reports

Events are specific occurrences that appear in the text to denote happenings or changes in states of the involved participants. Multiple guidelines defining events and their extents in text are proposed in the literature (Linguistic Data Consortium, 2005; Mitamura et al., 2017). It is important to note that no event annotated data is available for any incident text dataset and this compels us to consider event extraction approaches which are either unsupervised or involve minimal training data. We make a two fold contribution in this regard. Firstly, we annotate a moderately sized incident text dataset[4] for evaluation and weak supervision. Secondly, we propose a transfer learning approach based on the standard BiLSTM sequence labelling architecture and compare with three baselines from literature.

3.1 Describing and Annotating Events in Incidents Reports

For incident reports, we define events to be specific verbs and nouns which describe pre-incident, incident and post-incident happenings. Though the semantics of the events are specific to this domain, the nature and function of verbs and nouns representing events in standard domains is preserved. In this paper, we focus on extraction of event triggers i.e. the primary verb/noun token indicative of an event, as against an event phrase spanning multiple tokens. Identification of the event triggers is pivotal to the event extraction problem and once an event trigger is identified it is straightforward to construct an event span by collecting specific dependency children of the trigger. We present a set of examples of sentences and their event triggers we focus on extracting in Table 2.

The pilot <EVENT>**pulled**</EVENT> the collective to <EVENT>**control**</EVENT> the <EVENT>**descent**</EVENT>.
The helicopter <EVENT>**crashed**</EVENT> in the field and <EVENT>**sustained**</EVENT> substantial <EVENT>**damage**</EVENT>.

Table 2: Examples of event triggers

Keeping in mind the domain specific semantics of the events, we choose the Open Event extraction guidelines proposed by (Araki, 2018). We differ with these guidelines at a few places and suitably modify them before guiding our annotators for the task. The details of the differences are described as follows:

- (Araki, 2018) suggests labelling of individual adjectives and adverbs as events. Based on our observations of incident text data, we rarely find adjectives or adverbs being "eventive". Hence, we restrict our events to be either verbs (verb-based) or nouns (nominal).

- (Araki, 2018) suggests labelling of states and conditions as events. In the current work, we only focus on extraction of instantaneous events and do not extract events describing long-going state-like situations or general factual information. For example, we do not extract `had` in the sentence `The plane had three occupants` as an event as it only gives information about the plane but we extract all events such as `crashed` in the sentence `The plane crashed in the sea.`

- (Araki, 2018) suggests considering light verb constructions (such as "make a turn") as a single combined event. However, we saw a need to consider more such combined verb formulations. As an example, consider the events `scheduled` and `operate` in the sentence `The plane was scheduled to operate a sight seeing flight.`
  To better capture the complete event semantics, we do not consider these words as separate events but as a single combined event `scheduled to operate.`

3.2 Proposed Transfer Learning approach

Event extraction can be posed as a supervised sequence labelling problem and a standard BiLSTM-CRF based sequence labeller (Lample et al., 2016) can be employed. However, we reiterate that, as a large event annotated dataset specific to the domain of incident reports is not available, it would be difficult to train such a sequence labeller with high accuracy. We hypothesize that pre-training the BiLSTM-CRF sequence labeller with event labelled data from the general domain would help the network know about the general nature of verb-based and nominal events ("eventiveness"). Later as part of a transfer learning procedure (Yang et al., 2017), post-training of the network on a small event labelled dataset in incidents will provide us with an enriched incident event labeller. The proposed approach is based on this hypothesis and the transfer learnt model is then used to predict event triggers while testing.

[4] the dataset can be obtained through an email request to the authors

	#Reports	#Events
Training subset		
AVIATION	10	182
CONSTRUCTION	15	107
Test subset		
AVIATION	30	560
CONSTRUCTION	30	224

Table 3: Annotated Dataset Statistics

4 Experimentation and Evaluation

4.1 Dataset

We base our experimentation on incidents from two domains - AVIATION and CONSTRUCTION. To develop the AVIATION dataset, we crawled all the 54 reports about civil aviation incidents[5] recorded in India between 2003 and 2011. For the CONSTRUCTION dataset, we crawled 67 incident report summaries[6] of some major construction incidents in New York (May 1990 to July 2019). We annotate 40 incident reports from AVIATION and 45 from the CONSTRUCTION dataset for both events and event temporal ordering. We treat 10 reports in AVATION and 15 in CONSTRUCTION as a small labelled training dataset. The annotated dataset statistics are presented in Table 3.

4.2 Baselines

As the first baseline (B1), we consider the approach proposed in (Palshikar et al., 2019). The authors extract Message Sequence Charts (MSC) from textual narratives which depict messages being passing between actors (entities) in the narrative. Their message extraction approach forms the basis for this event extraction baseline. The approach first identifies past tense verbs and then considers flowing the past tense to its children present tense verbs. It then classifies all identified verbs as either an "action" or "communication" using WordNet hypernyms of the verb itself or its nominal forms and ignores all verbs which are neither actions nor communications (mental events such as `thought`, `envisioned`). The approach doesn't extract nominal events, so we supplement this baseline with a simple nominal event extraction technique. We first consider a NomBank (Meyers et al., 2004) based approach which checks each noun for its presence in the NomBank and if found marks it as a nominal event. We also consider another approach based on the deverbal technique proposed by Gurevich et al. (Gurevich et al., 2008), which checks if a candidate noun is the deverbal of any verb in the VerbNet (Palmer et al.). It tags the noun as a nominal event, if such a verb is found. We take a union of the output of the two approaches and filter it using the WordNet to remove obvious false positives (such as entities, etc.) and obtain a final set of nominal events from the given incident report.

As the second baseline (B2), we consider on Open Domain Event Extraction technique proposed in (Araki and Mitamura, 2018). Most prior work on extraction of events is restricted to (i) closed domains such as ACE 2005 event ontology and (ii) limited syntactic types. In this paper, the authors highlight a need for open-domain event extraction where events are not restricted to a domain or a syntactic type and hence this becomes a suitable baseline. The authors propose a distant supervision method to identify events. The method comprises of two steps: (i) training data generation, and (ii) event detection. In the first step of distantly supervised data creation, candidate events are identified and filtered using WordNet to disambiguate for their eventiveness. Further, Wikipedia is used to identify events mentioned using proper nouns such as "Hurricane Katrina". Both these steps help to generate lots of good quality (but not gold) training data. In the second step, BiLSTM based supervised event detection model is trained on this distantly generated training data. The experimental results show that the distant supervision improves event detection performance in various domains, without any need for manual annotation of events.

As the third baseline (B3), we use the standard BiLSTM based sequence labelling neural network (Lample et al., 2016) employed frequently in information extraction tasks such as Named Entity Recognition (NER). We use the small labelled training dataset to train this BiLSTM based sequence labeller for event identification and use it to extract events while testing.

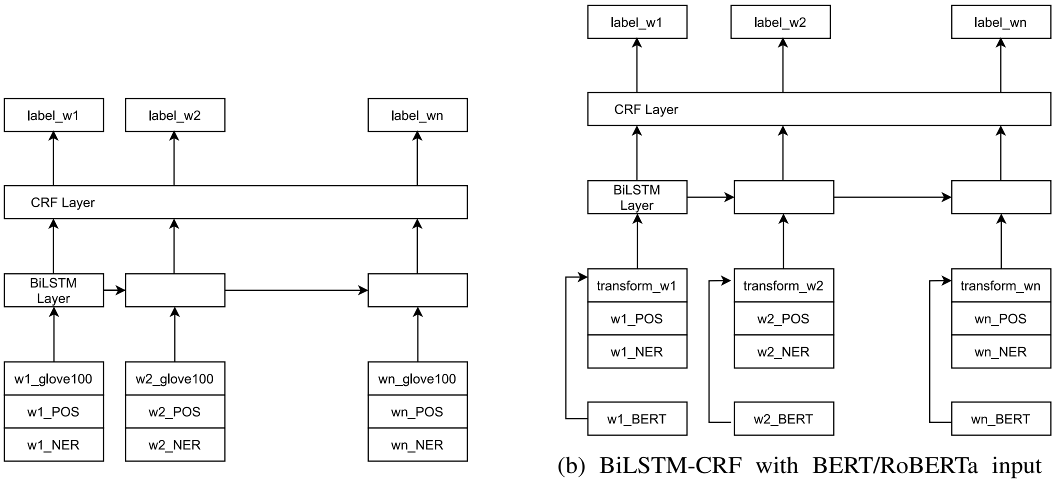
4.3 Experimentation Details

4.3.1 Word Embeddings

For representing the text tokens as input in the proposed neural network approaches, we experiment with the standard static embeddings (GloVe (Pennington et al., 2014)) and the more recent con-

[5] `https://dgca.gov.in/digigov-portal/?page=IncidentReports`

[6] `https://www.osha.gov/construction/engineering`



(a) Standard BiLSTM-CRF architecture

(b) BiLSTM-CRF with BERT/RoBERTa input transform layer

Figure 1: BiLSTM-CRF network models

textual embeddings (BERT (Devlin et al., 2018) and RoBERTa (Liu et al., 2019)). We consider 100-dimensional GloVe embeddings and 768-dimensional contextual BERT and RoBERTa representations for the experiments.

4.3.2 Neural Network Design and Tuning

The neural network architecture we use for baseline B3 and the proposed transfer learning approach is based on the BiLSTM-CRF architecture proposed by (Lample et al., 2016) for sequence labelling. It is shown in the Figure 1a. As part of the input we concatenate the word embeddings by 20 dimensional learnable POS and NER embeddings. We store these learnt embeddings alongwith the model and reload them during inference.

An important aspect to note is that large amount of training data is not available and hence the number of parameters which the network needs to learn should be as minimum as possible to avoid high bias. In particular the connection between the input layer which is 140 dimensional (in case of GloVe embeddings, $100 + 20\ POS + 20\ NER$) and the BiLSTM layer (with hidden units 140) is $140 \times 140 \times 2$. In case of 768-dimensional BERT/RoBERTa based representations it blows up about 6 times to $768 \times 768 \times 2$, assuming the LSTM hidden units are also 768. The network fails to learn while training using the limited data in case of 768-dimensional embeddings. So we devise a small change to the input layer to support learning in this case. We introduce a dense layer just after the 768-dimensional BERT/RoBERTa input with a linear activation function to map the 768-dimensional input into a smaller dimensional space, as shown in Figure 1b. Due to the linear activation, this layer behaves like a linear transformation of a high dimensional input vector to a lower dimensional input vector. Additionally, we concatenate the previously mentioned POS and NER learnable embeddings to the transformed input embeddings as the final input to the network.

We employ 5-fold cross-validation on the small training dataset for tuning the hyperparameters of the neural network separately for both domains and embedding types. We found minimal difference in hyperparameter values across both Aviation and Construction datasets and hence, we use similar parameters in both cases. The tuned hyperparameters with their values are shown in Table 4.

Hyperparameter	**GloVe based model (Fig. 1a)**	**BERT/ RoBERTa based model (Fig. 1b)**
input_word_embedding_dimension	100	768
input_word_transform_dimension	NA	200
input_pos_embedding_dimension	20	20
input_ner_embedding_dimension	20	20
bilstm_hidden_units	140	240
bilstm_recurrent_dropout	0.3	0.3
crf_input_dimension	70	120
optimizer	adam	adam
epochs	20	30
batch_size	8	16
pre-training_epochs	20	20
pre-training_batch_size	16	16

Table 4: Tuned Hyperparameters

4.3.3 Implementation

Baseline B1 is unsupervised and is implemented and used directly. Code for baseline B2 is made available by the authors[7] and we install and use it without any change. The BiLSTM-CRF sequence labelling networks, used for baseline (B3) and the transfer learning approach, is implemented using keras in python 3. These approaches are trained on the small training data shown in Table 3. To handle randomness in neural network weight initialization and to ensure robustness of the results, we run every neural network experiment (both hyperparameter tuning as well as final test experiments) five times and report an average of the five runs. We were able to observe standard deviation in the precision, recall and F1 of these runs to be as low as 1-2%. With respect to the pre-training data for the transfer learning approach, we use the event annotations from the ECB dataset (Bejan and Harabagiu, 2010). It is a dataset for Event Coreference tasks and has comprehensive event annotations (about 8.8K labelled events in about 9.7K sentences).

4.4 Evaluation and Analysis

As we can observe in Table 5, the proposed transfer learning approach (TL) outperforms the other baselines (B1, B2 and B3) in performance irrespective of static or contextual embeddings. Further, as expected the BiLSTM based baseline B3 shows lower recall than the transfer learning approach in which we see significantly improved recall particularly for the Construction dataset for all embedding types. We observe a similar boost in recall particularly for BERT representations on the Aviation dataset. An important point to note here is that the amount of pre-training data, leading to best results, varies between 40% to 60% for combinations of dataset and embedding type. In Table 5, we report the performance for best amount of pre-training data and present a detailed analysis on effect on increasing pre-training data in Section 4.4.1.

As part of the analysis, we first measure the effect of increase in the amount of pre-training data in the transfer learning approach and find out what amount of pre-training leads to the best results. Secondly, we try to explain why the pre-training works through a novel clustering methodology over the BiLSTM learnt context representations of the input embeddings. And thirdly, we present an ensemble approach considering a practical standpoint of using these systems in real-life use cases.

	AVIATION			CONSTRUCTION		
	P	**R**	**F1**	**P**	**R**	**F1**
B1	0.67	0.83	0.74	0.63	0.8	0.7
B2	0.71	0.89	0.79	0.64	0.95	0.77
$\mathbf{B3}_{\mathbf{GloVe100}}$	0.83	0.83	0.83	0.87	0.69	0.77
$\mathbf{TL}_{\mathbf{GloVe100}}$	0.86	0.84	0.85	0.91	0.75	0.82
$\mathbf{B3}_{\mathbf{BERT}}$	0.84	0.79	0.82	0.84	0.63	0.72
$\mathbf{TL}_{\mathbf{BERT}}$	0.87	0.83	0.85	0.9	0.73	0.81
$\mathbf{B3}_{\mathbf{RoBERTa}}$	0.87	0.83	0.85	0.8	0.63	0.71
$\mathbf{TL}_{\mathbf{RoBERTa}}$	0.86	0.85	0.86	0.85	0.79	0.82
ENS	0.90	0.83	0.86	0.95	0.75	0.84

Table 5: Evaluation - Event Extraction

4.4.1 Amount of pre-training data

As an important part of the analysis, we measure what is the effect of increase in pre-training data in the transfer learning approach. We hypothesize that the performance would rise till a certain point with increasing pre-training data and would then stabilize and change minimally. This is based on the notion that pre-training positions the network weights in a better space from where the training on domain specific data should begin. However, beyond a certain amount of pre-training the initialization may not lead to any better initial values for the weights.

To check the validity of this hypothesis, we pre-trained the network with varied amounts of pre-training data (1%, 5%, 10%, 20%, 30%, ..., 100%) and checked the performance on test data. Figure 2 and Figure 3 show the obtained F1 curves for these pre-training settings for Aviation and Construction datasets respectively. As with other experiments, each point in the graphs is an average of performance for 5 runs of training and testing.

It can be seen that with increasing pre-training data, the performance improves and reaches a peak between 30% to 70% of pre-training data available, varying for different input embedding types. We observe a small dip in performance when amounts near complete pre-training data are used. Interestingly, BERT based representations start showing promise with even 1% of pre-training data for the Aviation dataset.

4.4.2 Explanability of Pre-training

To explain why the pre-training is helping, we need to have an understanding of what the network is learning about the input embeddings of the tokens and their context from the bidirectional LSTM. It would be helpful if one could analyze the token-

[7] https://bitbucket.org/junaraki/coling2018-event

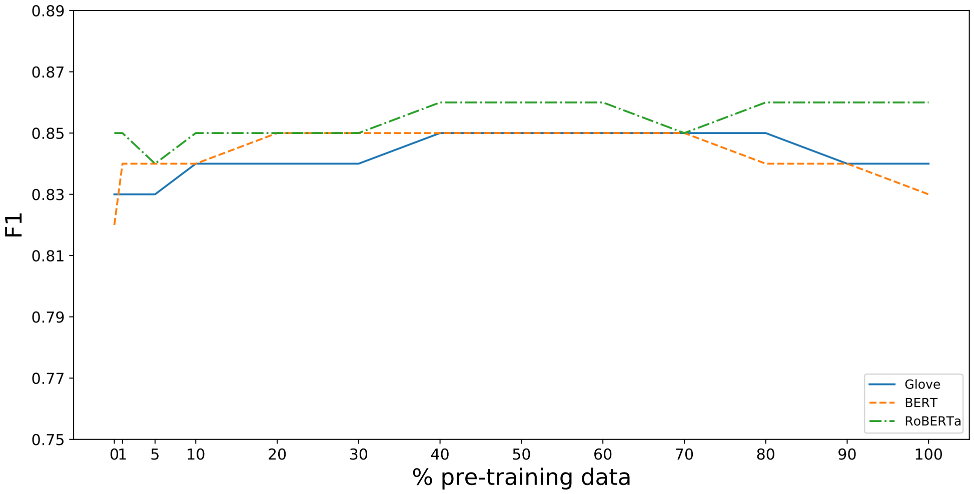


Figure 2: Increase in Pre-training Data - Aviation

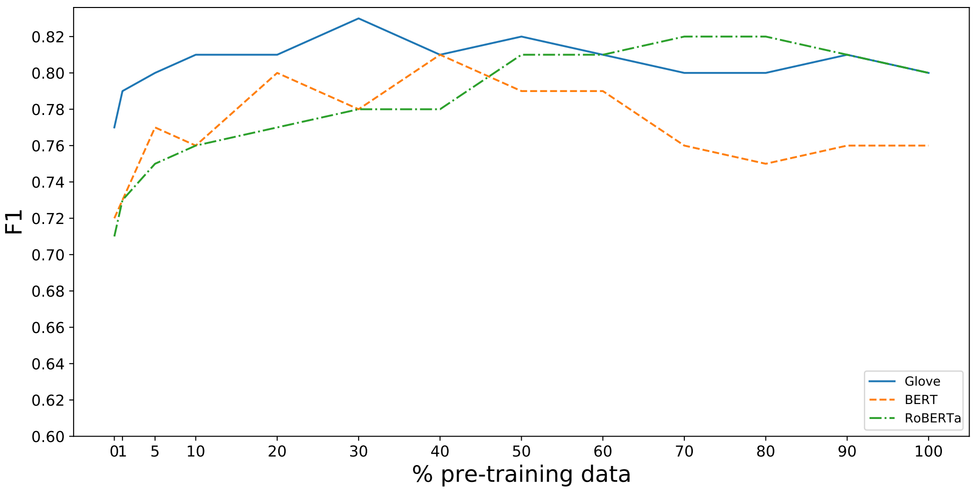


Figure 3: Increase in Pre-training Data - Construction

wise output of the BiLSTM layer, which incorporates both the input embeddings and the context information and feeds these representations to the CRF layer as features for sequence learning/inference (See Figure 1a). However, internal representations in a neural network are a set of numbers not comprehensible in a straightforward manner and would require an indirect observation to decipher what is captured by them. One such indirect analysis of these internal representations involves performing their clustering and observing if representations with similar semantics cluster together and rarely cluster with dissimilar representations. In this case, the desired semantics would mean capture of the "eventiveness" property in event tokens. We perform such a clustering based analysis on extractions in the Construction dataset.

Token	Gold Label	TL Prediction	B3 Prediction
t1	EVENT	EVENT	EVENT
t2	EVENT	EVENT	O
t3	EVENT	O	O
t4	O	O	O

Table 6: Example Tokens and Predictions

We consider all tokens which are marked as events in the gold and are also correctly predicted as events by the transfer learnt model (TL) such as tokens t1 and t2 in Table 6. We obtain the BiLSTM output representations for these tokens by passing their sentences through the TL model truncated till the input of the CRF layer and collect these representations (r^{t1}_{TL} and r^{t2}_{TL}) in a set R_{TL}. As observed from the results, the baseline model B3 has a lower recall than the TL model and for tokens such as t1 and t2, we can categorize the predictions of the B3 model into either 'correctly predicted as events' or 'missed and marked as non-events'. We divide these tokens into the correct and incorrect sets as per their baseline model predictions. We obtain the BiLSTM output representations for these tokens from the B3 model in the similar way as earlier and respectively collect these representations (r_{B3}^{t1} and r_{B3}^{t2}) in two sets R_{B3C} (B3 corrects) and R_{B3I} (B3 incorrects). We hypothesize that all the representations which lead to a correct event prediction should belong to a subspace of "eventive" representations and should be far from the representations which lead to an incorrect prediction. Hence, representations in the set R_{TL} and R_{B3C} should cluster differently from the representations in the set R_{B3I}. So, in the context of the example tokens of Table 6, representations r_{TL}^{t1}, r_{TL}^{t2} and r_{B3}^{t1} should cluster differently from r_{B3}^{t2}.

On performing agglomerative clustering on the above representations with a maximum distance of 0.3 (standard similarity of 0.7), we find that the representations R_{TL} and R_{B3C} belong to multiple clusters which are highly separate from clusters housing the representations in R_{B3I}. This validates our hypothesis and highlights positioning of R_{TL} and R_{B3C} representations closer to the required "eventiveness" subspace and far from the R_{B3I} representation which lead to incorrect predictions. We further strengthen the claim by computing purity (Manning et al., 2008) of the representation clusters. The purity of a clustering gives a measure of the extent to which clusters contains instances of a single class. In case of predictions based on GloVe embeddings models, we observe a purity of 0.9781 and in case of BERT embeddings models, we observe a purity of 0.9832.

4.4.3 Practical standpoint

We also performed a detailed analysis with regard to the errors in verb-based and nominal event predictions. It was observed that the deep learning approaches miss important verb-based events leading to low recall particularly for the verb-based events, but identify nominal events correctly in most cases. The rule based baseline B1, captures all the verb-based events mostly as it designates most past tense verbs as events. However, the rule based approach fails to identify nominal events correctly as it doesn't observe the context of a noun while deciding its event nature. This observation prompted us to perform a novel ensemble where we create a union of all verb-based event predictions of the rule based approach and all nominal event predictions of the transfer learning based approach using glove embeddings. We believe this ensemble approach holds value from a practical standpoint in two ways. Firstly, using GloVe embeddings eases compute and maintenance requirements in deployment environments, which are higher for handling BERT/RoBERTa based contextual models. Further, as seen from the results in Table 5, GloVe embeddings perform at par with contextual representations. Secondly, when showing a user predictions of events from an incident report, she might get perturbed more because of incorrect nominal events than some extra verbal events. As seen in Table 5, this ensemble approach (row marked as ENS) shows a respectable increase in precision over the Transfer learning approach in both datasets and may be useful to employ in real life incident event identification systems.

5 Conclusion and Future Work

In this paper we focused on extracting events from reports on incidents in Aviation and Construction domains. As there is no dataset of incident reports comprising of annotations for event extraction, we contributed by proposing modifications to a set of existing event guidelines and accordingly preparing a small annotated dataset. Keeping in mind the limited data settings, we proposed a transfer learning approach over the existing BiLSTM-CRF based sequence labelling approach and experimented with different static and contextual embeddings. We observed that pretraining improves performance of event extraction for all combinations of domains and embeddings. As part of the analysis, we showed the impact of employing varying amounts of pretraining data. We also performed a novel clustering based analysis to explain why pretraining improves performance of event extraction. We also propose a novel ensemble approach motivated from a practical viewpoint.

As future work, we plan to pursue other important stages of the incident report analysis pipeline such as (i) entity/actor identification which involves finding the important participants in an incident, (ii) event argument identification which involves finding participants which are agents or experiencers of the event, (iii) state/condition identification which involve finding expressions describing long-running state-like conditions and (iv) event-event relation identification which involves establishing of relation links between events.

References

Jun Araki. 2018. *Extraction of Event Structures from Text*. Ph.D. thesis, Carnegie Mellon University.

Jun Araki and Teruko Mitamura. 2018. Open-Domain Event Detection using Distant Supervision. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING)*, pages 878–891, Santa Fe, NM, USA.

Harsimran Bedi, Sangameshwar Patil, Swapnil Hingmire, and Girish Palshikar. 2017. Event timeline generation from history textbooks. In *Proceedings of the 4th Workshop on Natural Language Processing Techniques for Educational Applications (NLPTEA 2017)*, pages 69–77.

Cosmin Bejan and Sanda Harabagiu. 2010. Unsupervised event coreference resolution with rich linguistic features. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1412–1422, Uppsala, Sweden. Association for Computational Linguistics.

Tirthankar Dasgupta, Abir Naskar, Rupsa Saha, and Lipika Dey. 2018. Extraction and visualization of occupational health and safety related information from open web. In *2018 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2018, Santiago, Chile, December 3-6, 2018*, pages 434–439. IEEE Computer Society.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805*.

Olga Gurevich, Richard Crouch, Tracy Holloway King, and Valeria De Paiva. 2008. Deverbal nouns in knowledge representation. *Journal of Logic and Computation*, 18(3):385–404.

Swapnil Hingmire, Nitin Ramrakhiyani, Avinash Kumar Singh, Sangameshwar Patil, Girish Keshav Palshikar, Pushpak Bhattacharyya, and Vasudeva Varma. 2020. Extracting Message Sequence Charts from Hindi Narrative Text. In *Proceedings of the First Joint Workshop on Narrative Understanding, Storylines, and Events, NUSE@ACL 2020, Online, July 9, 2020*, pages 87–96. Association for Computational Linguistics.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California. Association for Computational Linguistics.

Linguistic Data Consortium. 2005. ACE (Automatic Content Extraction) English Annotation Guidelines for Events.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692*.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, USA.

A. Meyers, R. Reeves, C. Macleod, R. Szekely, V. Zielinska, B. Young, and R. Grishman. 2004. The NomBank Project: An Interim Report. In *HLT-NAACL 2004 Workshop: Frontiers in Corpus Annotation*, pages 24–31, Boston, Massachusetts, USA. Association for Computational Linguistics.

MINERVA. The MINERVA Portal of European Commission. `https://minerva.jrc.ec.europa.eu/en/minerva/about`. [Online; accessed 26-Apr-2021].

Teruko Mitamura, Zhengzhong Liu, and Eduard H. Hovy. 2017. Events detection, coreference and sequencing: What's next? overview of the TAC KBP 2017 event track. In *Proceedings of the 2017 Text Analysis Conference, TAC 2017, Gaithersburg, Maryland, USA, November 13-14, 2017*. NIST.

OSHA. Occupational Safety and Health Administration. `https://www.osha.gov/Publications/3439at-a-glance.pdf`. [Online; accessed 26-Apr-2021].

Martha Palmer, Claire Bonial, and Jena Hwang. Verbnet. In *The Oxford Handbook of Cognitive Science*.

Girish Palshikar, Sachin Pawar, Sangameshwar Patil, Swapnil Hingmire, Nitin Ramrakhiyani, Harsimran Bedi, Pushpak Bhattacharyya, and Vasudeva Varma. 2019. Extraction of message sequence charts from narrative history text. In *Proceedings of the First Workshop on Narrative Understanding*, pages 28–36, Minneapolis, Minnesota. Association for Computational Linguistics.

Justin Pence, Pegah Farshadmanesh, Jinmo Kim, Cathy Blake, and Zahra Mohaghegh. 2020. Data-theoretic approach for socio-technical risk analysis: Text mining licensee event reports of u.s. nuclear power plants. *Safety Science*, 124:104574.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Ludovic Tanguy, Nikola Tulechki, Assaf Urieli, Eric Hermann, and Cline Raynal. 2016. Natural language processing for aviation safety reports: From classification to interactive analysis. *Computers in Industry*, 78:80–95. Natural Language Processing and Text Analytics in Industry.

Xiaozhi Wang, Xu Han, Zhiyuan Liu, Maosong Sun, and Peng Li. 2019. Adversarial training for weakly supervised event detection. 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 998–1008, Minneapolis, Minnesota. Association for Computational Linguistics.

Zhilin Yang, Ruslan Salakhutdinov, and William W Cohen. 2017. Transfer learning for sequence tagging with hierarchical recurrent networks. *arXiv preprint arXiv:1703.06345*.