

Event Detection Using Frame-Semantic Parser

Evangelia Spiliopoulou
Carnegie Mellon University
espiliop@cs.cmu.edu

Eduard Hovy
Carnegie Mellon University
hovy@cmu.edu

Teruko Mitamura
Carnegie Mellon University
teruko@cs.cmu.edu

Abstract

Recent methods for Event Detection focus on Deep Learning for automatic feature generation and feature ranking. However, most of those approaches fail to exploit rich semantic information, which results in relatively poor recall. This paper is a small & focused contribution, where we introduce an Event Detection and classification system, based on deep semantic information retrieved from a frame-semantic parser. Our experiments show that our system achieves higher recall than state-of-the-art systems. Further, we claim that enhancing our system with deep learning techniques like feature ranking can achieve even better results, as it can benefit from both approaches.

1 Introduction

Automatic Event Detection is an important and challenging task in Natural Language Processing and Information Extraction. According to the ACE 2005 Evaluations (ACE, 2005), an Event is defined as a specific occurrence that describes a change of state, the **Event Nugget**, and it involves a set of participants, the **Event Arguments**. The term Event Nugget (TAC, 2014) refers to a semantically meaningful unit of text that denotes some action (event), while the Event Arguments are Entity mentions or temporal expressions related to the Event Nugget. In this work, we focus on the task of Event Nugget Detection and its classification to types and subtypes of Events, according to the ACE 2005 guidelines.

Current Event Detection methods that achieve state-of-the-art results are based on Deep Learning techniques using shallow lexical features and word embeddings (Chen et al., 2015), (Nguyen and Gr-

ishman, 2015). Although these approaches open the door to automatically extracted features, they fail to exploit deeper semantic information. This results in a limited number of detected events and, consequently, in low recall/ high precision systems.

In this work, we investigate a different approach on the Event Detection task, that achieves higher recall by generating a large set of candidate events using a semantic-frame parser. Semantic-frame parsers output a variety of linguistic structures, including events, relations and entities. Similar to the approach followed by Liu et al. (2016), we exploit the similarities in structure between FrameNet and the ACE Ontology to create a mapping from the former to the latter. We use this mapping to refine the parser’s output and classify the linguistic structures as event mentions. In this paper we show that this approach results in a high recall system (72.6%) which, if combined with a deep learning model, can achieve better recall without loss in precision.

2 Background

2.1 The ACE Dataset

According to the ACE 2005 Evaluations (ACE, 2005), an Event contains two spans: the Event Nugget and the Event Arguments. Although there are several types of events, the ACE annotations include only events that can be defined under a certain ontological structure. This structure contains 8 event types followed by a total of 33 event subtypes. The event types are: LIFE, MOVEMENT, TRANSACTION, BUSINESS, CONFLICT, CONTACT, PERSONNEL and JUSTICE. In this work, we focus on the Event Nugget detection and its classification to one type/subtype pair, as defined by the ACE guidelines.

2.2 FrameNet

FrameNet (Baker et al., 1998) is a taxonomy of more than 1,200 manually identified semantic frames, deriving from a corpus of 200,000 annotated sentences. The aim of the FrameNet semantic frames is to capture information about the type of a linguistic structure, which can be an event, entity or relation, and its participants. This type is called **Frame** and the participants are called **Frame Elements**. Each Frame is linked to a set of words that may trigger the Frame (**Lexical Units**).

Following the definition of FrameNet semantic frames and the ACE 2005 guidelines, it seems natural to assume a good correspondence between the two resources. This property implies that a mapping from FrameNet Frames to ACE types and subtypes can be extremely helpful in Event Detection (Liu et al., 2016).

2.3 Semafor

Semafor (Das et al., 2014) is a semantic frame parser based on the FrameNet taxonomy. Semafor follows a semi-supervised approach to detect words that are FrameNet triggers, which evoke some semantic frame(s). Semafor’s output contains a set of FrameNet semantic frames, their trigger and their Frame Elements.

Since Semafor is based on FrameNet, its triggers can be events, entities or relations. This implies that Semafor cannot be directly applied on the Event Detection problem, since it has extremely low precision. However, in this paper we will present how Semafor can be used as an additional resource to enhance Event Detection recall.

2.4 Related Work

Recent research that achieves state-of-the-art results is primarily based on deep learning techniques. Chen et al. (2015) propose a dynamic multi-pooling convolutional neural network (DM-CNN), which automatically induces lexical-level and sentence-level features from text, achieving state-of-the-art results. Nguyen and Grishman (2015)’s work focuses on CNNs using word embeddings in order to achieve a more generalizable event detection system. Other approaches include Ghaeini et al. (2016)’s FBRNN, which is a modification of RNNs using word and branch embeddings, and Liu et al. (2016)’s ANN & Random ANN, which exploits the direct relationship between the FrameNet and the ACE Ontology in or-

der to construct an out-domain ANN model. Peng et al. (2016) showed that it is feasible to achieve state-of-the-art results with minimal supervision. In their approach, they use only a few examples and the SRL of a candidate event in order to construct a structured vector representation, which maps the event to an ontology.

3 Approach

In this paper, we present a system that uses a semantic-frame parser in order to generate event candidates, which are then filtered according to a mapping between ontologies. The main motivation behind this approach is that most systems based on deep learning methods do not exploit rich semantic information and therefore miss non-surface-level equivalences, which results in low recall. Furthermore, we claim that a combination of a semantically rich system with a deep learning approach can result in better overall performance than both traditional semantic-based approaches and pure deep learning methods.

3.1 Using Semafor

In order to generate a list of candidate events, we need a system with very high recall that contains semantic information about the event. A semantic-frame parser like Semafor, extracts a variety of semantic structures, as events, entities and relations. Furthermore, since it is based on FrameNet, it provides semantic information (Frame), which is essential for the classification of the structure as an event. In order to test the performance of Semafor on the ACE dataset and whether it is a reasonable choice for the system, we run experiments on the Newswire dataset and report the following: *Recall* 82.53%, *Precision* 6.8% and F_1 12.6%.

3.2 Defining Events

Based on the ACE 2005 guidelines, we define an event as a nominal or verbal phrase that can be mapped to a subtype of the ACE 2005 Ontology. Utilizing the structural similarity of ACE and FrameNet, we construct a mapping from a subset of FrameNet frames to ACE subtypes. We decided to create two different mappings, according to the POS tag of the trigger. This is because Event Nuggets can be either nominal or verbal phrases, each triggered by different sets of Frames.

In Table 1 we present the mapping of FrameNet Frames to ACE types for verbal mentions. Fur-

ther, for a small number of frames, we use a set of lexically-based disambiguation rules to find the correct subtype. An example of such a rule is that the frame *Verdict* may correspond to both *Convict* and *Acquit* subtypes. Because FrameNet LUs do not include several words, this mapping does not cover the ACE Ontology. Thus, additional disambiguation rules may even further increase the precision and recall of the current model.

ACE Type	FrameNet Frame
Conflict	Invading, Attack, Explosion, Destroying, Hostile encounter, Use firearm, Shoot projectiles, Downing, Explosion, Destroying, Protest, Political actions
Life	Giving birth, Being born, Death, Killing, Forming relationships, Cause harm, Personal relationship, Cause harm, Dead or alive
Movement	Self motion, Inhibit movement, Travel, Departing, Arriving, Visiting, Motion, Cause motion, Bringing
Transaction	Import export scenario, Commerce buy, Commerce sell, Getting, Commerce pay, Borrowing, Giving
Business	Activity start, Conquering, Endeavor failure, Intentionally create, Business closure, Locale closure
Contact	Meet with, Discussion, Come together, Communication, Contacting, Communication means, Text creation, Request
Personnel	Take place of, Get a job, Hiring, Appointing, Removing, Firing, Quitting, Choosing, Becoming a member, Change of leadership
Justice	Arrest, Imprisonment, Detaining, Extradition, Breaking out captive, Try defendant, Pardon, Appeal, Verdict, Sentencing, Fining, Execution, Releasing, Notification of charges

Table 1: Mapping of FrameNet verbs to ACE Ontology.

3.3 System Architecture

We first use Semafor to generate a set of candidate Event Nuggets, their FrameNet frame and their

Frame Elements. Then we use the POS tagger from Stanford CoreNLP (Manning et al., 2014) in order to distinguish the candidate events to verbal events and nominal events. For every trigger in the candidate events, we use the output FrameNet Frame in order to decide whether it is an event or not. If the Frame is in the domain of the FrameNet to ACE mapping, then it means that it corresponds to some subtype of the ACE Ontology and, thus, we accept it as an event. Furthermore, according to the mapping, we assign the type and subtype of the event. In Figure 1 we see an example output of the system for one article. The events are represented with green, red and black color if they are true positives, false positives and false negatives, respectively.

Type	Subtype	Event Nugget
Life	Die	killed
Life	Injure	wounded
Conflict	Attack	blast
Life	Die	death
Contact	Phone-Write	Radio
Life	Die	death
Transaction	Transfer-Money	giving
Life	Die	deaths
Life	Injure	wounded
Life	Injure	injured
Contact	Phone-Write	list
Conflict	Attack	bomb
Conflict	Attack	tore
Conflict	Attack	explosion
Conflict	Attack	hit
Life	Injure	injuries
Conflict	Attack	blast
Conflict	Attack	bomb
Conflict	Attack	exploded
Contact	Phone-Write	radio
Movement	Transport	carting
Life	Injure	wounded

Figure 1: Example output of Event Nuggets, Types and Subtypes.

Source: ACE 2005, Newswire, AFP_ENG_20030304.0250

4 Experiments

4.1 Dataset

The test dataset used for experiments is subset of the ACE 2005 corpus. It contains 106 Newswire articles and a total of 1557 event mentions. The methods we compare with are tested on a randomly selected subset of those articles (40 articles), as they needed to use some development data. Since our approach does not require training, we tested on the entire Newswire ACE 2005 corpus. In this way, our results show a more complete picture of the performance of our system on

the corpus.

4.2 System Evaluation

Method	Recall	Precision	F1
Liu’s ANN (2016)	60.7	79.5	68.8
Liu’s Random ANN (2016)	49.5	81.0	61.5
Chen’s DMCNN (2015)	67.7	80.4	73.5
Peng’s MSEP (2016)	69.8	75.6	72.6
Proposed Model	72.6	43.3	54.2

Table 2: Evaluation on Event Nugget Detection.

For the Event Nugget Detection task, we report and compare the *Recall*, *Precision* and F_1 with the state-of-the-art methods discussed in Section 2.4. Out of a total of 1557 event mentions, the proposed system correctly recognizes 1131. As we see in Table 2, although the proposed system has relatively low precision, it achieves significantly higher recall than current state-of-the-art systems. This indicates that our model is a good candidate for integration with other systems, a hypothesis further discussed in the subsequent sections.

A second metric of evaluation is the classification of the Event Nuggets to types and subtypes. Out of a total of 1557 types and 1557 subtypes, our system correctly recognizes 1044 types and 1018 subtypes. This highlights that our system solves the problem of Event Detection simultaneously with the event classification to types and subtypes. According to our results, 92.3% and 90.0% of the events that were correctly identified by our system were also correctly classified to types and subtypes, respectively. In Table 3, we report the *Recall*, *Precision* and F_1 measure of the ACE subtypes, viewed as a classification task without prior information about the Event Nuggets. We observe that our system still has the highest recall amongst the compared methods. Since precision on the Event Nugget Detection task was low, prior errors are also propagated to the event subtype classification.

Method	Recall	Precision	F1
Peng’s MSEP (2013)	65.0	70.4	67.6
Chen’s DMCNN (2015)	63.6	75.6	69.1
Proposed Model	65.4	39.0	48.35

Table 3: Evaluation on Event Subtype Classification.

4.3 Further Experiments

We claim that merging a high recall system based on rich semantic information with a deep learning classifier may achieve better results than current approaches on Event Detection, since it can benefit from both techniques. In a preliminary exploration of this hypothesis, we construct a dataset of candidate events based on our system’s output and run classification experiments on them for the Event Nugget Detection task (binary classification). As described in previous approaches, we randomly split the ACE Newswire articles into 60% train and 40% test set. Each instance on those sets represents an extracted Event Nugget of our system for the corresponding article. The features of each instance are:

- **Shallow features:** Event Nugget textual representation & lemma, Part-of-Speech tag, Right Context & Left Context (one position away)
- **FrameNet Frame:** the Frame that Semafor extracted for this Event Nugget.
- **Frame Elements:** a list of the Frame Elements roles (eg Agent, Target) that Semafor extracted for this Event Nugget.
- **Predicted Type and Subtype:** the type and subtype that the our system predicted.

In table 4 we show the results of a Random Forest and a vanilla Neural Net (15 hidden layers) on this dataset. We compare those results with our system’s output on the test set. Since the dataset contains only our system’s output, the recall upper bound of any classifier is 73.43% (our system’s recall).

Classifier	Recall	Precision	F1
None (system output)	73.43	39.2	51.11
Random Forest	52.8	72.6	61.1
Neural Net	51.53	65.0	57.49

Table 4: Classification of Events on our system’s output.

Overall the classifiers behave in a similar way, since both of them show a drop in recall and a significant gain on precision. Further, we observe a significant increase on the F_1 score, which indicates that there is an actual system improvement instead of tweaking the precision/recall tradeoff.

A second interesting observation is that the Random Forest classifier gives better results than the Neural Net. We have identified two reasons for that. First, our dataset is extremely small (train: 1550 instances, test: 1045 instances), which results in insufficient training of the Neural Net. The second reason is the nature of the two classifiers. In general, Neural Nets are very strong at discovering new features, which is extremely important when there is a great bulk of hidden information not included as dataset features. On the other hand, Trees select and rank actual dataset features that give maximum entropy. Since each instance in our dataset has a small set of features that are either nominal classes or unigrams, Neural Nets fail to discover good new features. Instead, a good feature reranker as Random Forests, results in better feature selection and, consequently, higher precision.

5 Conclusion and Future Work

Our experiments indicate that our system achieves higher recall than current state-of-the-art systems, while maintaining a reasonable precision. This illustrates that semantically rich features give a great boost in recall that deep learning methods alone cannot reach, due to the inadequacy of shallow linguistic features to capture deep semantic information. On the other hand, deep learning methods are very good at automatic feature extraction and feature ranking, which leads in extremely high precision. We claim that merging the two approaches in one system can solve the current trade-off between precision and recall, as the new sys-

tem will benefit from both techniques.

As our preliminary experiments show, integrating our system with a classifier results in significantly better system performance. The fact that this is achieved with vanilla models which are not widely used for the Event Detection task, is a strong indicator that we can further improve the results by using more suitable models. Our next step is to investigate our system integration with a more sophisticated deep learning classifier instead of off-the-shelf vanilla models. Further, we plan to use an enlarged version of the dataset by including instances that our system did not recognize as Event Nuggets (e.g. all verbal and nominal mentions) with a lesser weight. In that way, we can reduce the previously unavoidable drop in recall, since we will add a bias but not enforce the candidate events to be part of our system’s output.

An alternative approach to the system integration involves the construction of a collective output of multiple Event Detection systems with their corresponding confidence scores, if available. We plan to use this as input to a deep learning model, in a similar fashion with the approach discussed earlier. We claim that this classifier will capture more information about events, since it can learn from the strengths and weaknesses of the multiple Event Detection systems involved.

Acknowledgments

This research was supported in part by DARPA grant FA8750-12-2-0342 funded under the DEFT program.

References

- 2005. *ACE (Automatic Content Extraction) English Annotation Guidelines for Events*. Linguistic Data Consortium, 5.4.1 edition.
- 2014. *TAC KBP Event Detection Annotation Guidelines*, 1.7 edition.
- Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The berkeley framenet project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 1*.
- Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling 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 1: Long Papers).

Dipanjan Das, Desai Chen, André FT Martins, Nathan Schneider, and Noah A Smith. 2014. Frame-semantic parsing. *Computational Linguistics* .

Reza Ghaeini, Xiaoli Z Fern, Liang Huang, and Prasad Tadepalli. 2016. Event nugget detection with forward-backward recurrent neural networks. In *The 54th Annual Meeting of the Association for Computational Linguistics*.

Shulin Liu, Yubo Chen, Shizhu He, Kang Liu, and Jun Zhao. 2016. Leveraging framenet to improve automatic event detection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*.

Thien Huu Nguyen and Ralph Grishman. 2015. Event detection and domain adaptation with convolutional neural networks. In *ACL (2)*.

Haoruo Peng, Yangqiu Song, and Dan Roth. 2016. Event detection and co-reference with minimal supervision. In *EMNLP*.