Event Extraction as Frame-Semantic Parsing

Alex Judea and **Michael Strube**
Heidelberg Institute for Theoretical Studies gGmbH
Schloss-Wolfsbrunnenweg 35
69118 Heidelberg, Germany
(alex.judea|michael.strube)@h-its.org

Abstract

Based on the hypothesis that frame-semantic parsing and event extraction are structurally identical tasks, we retrain SEMAFOR, a state-of-the-art frame-semantic parsing system to predict event triggers and arguments. We describe how we change SEMAFOR to be better suited for the new task and show that it performs comparable to one of the best systems in event extraction. We also describe a bias in one of its models and propose a feature factorization which is better suited for this model.

1 Introduction

Event Extraction is a task in information extraction where mentions of predefined events are extracted from texts. We follow the task definition of the Automatic Content Extraction (ACE) program of 2005. It defines 33 event types, organized in eight categories. Each event type has associated *roles*, e.g., ATTACK has the roles *attacker*, *target*, and *instrument*, whereas DIE has the roles *agent*, *victim*, and *instrument*. The roles *place* and *time* are shared by all event types.

ACE events occur only within sentences. Each event is indicated by a word, the *trigger*. The roles associated with the respective event type are filled by zero or more *arguments*. Most arguments are mentions of entities, e.g. persons, locations, or organizations. Some arguments are mentions of points in time, amounts of money, etc. Arguments may be shared by multiple events and may play different roles in each of them.

Figure 1 illustrates an example. The sentence contains two events, DIE and ATTACK, triggered by "died" and "fired", respectively. For DIE, the roles *victim*, *instrument*, and *place* are filled with the arguments "cameraman", "American tank", and "Baghdad", respectively. For ATTACK, the role *target* has two arguments, namely "cameraman" and "Palestine hotel", the roles *instrument*, and *place* have the arguments, "American tank", and "Baghdad", respectively. Three arguments are shared. One of them, "cameraman", plays different roles in the events, namely *victim* of DIE and *target* of ATTACK.

Frame-semantic parsing is the task of extracting semantic predicate-argument structures from texts. It is built on the theory of frame semantics and FrameNet (Fillmore et al., 2003; Das et al., 2014). As in event extraction, frames occur within sentences and have triggers and roles (called lexical units and frame elements).

Our hypothesis is that the two tasks are structurally identical. From a computational point of view, they differ only in feature types. We can use the same approach and infrastructure to tackle both. Based on this hypothesis, we retrain a frame-semantic parsing system, SEMAFOR, for event extraction.

We describe differences between frame-semantic parsing and event extraction and the adaptions needed to better prepare SEMAFOR for the new task. We also describe a bias in the trigger classification model which affects frame-semantic parsing as well as event extraction and propose a new factorization of features which is better suited for this

*Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics (*SEM 2015)*, pages 159–164,
Denver, Colorado, June 4–5, 2015.

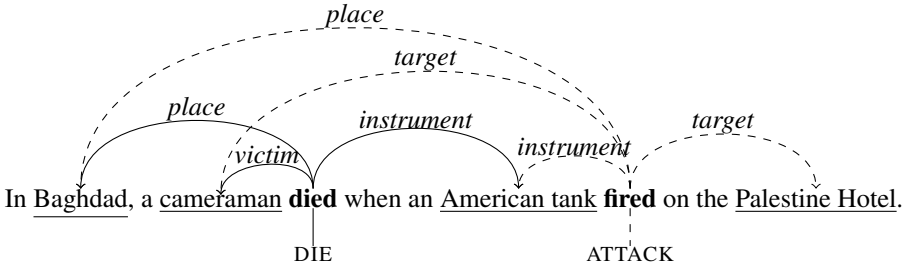

Figure 1: A sentence with two event instances, a DIE event triggered by the word "died", and an ATTACK event triggered by "fired". Three arguments are shared by both events.

model. Finally, we evaluate the retrained system on the ACE 2005 data (Walker et al., 2006).

2 Related Work

Many approaches to event extraction do not cross sentence boundaries, e.g. Grishman et al. (2005), Ahn (2006), Lu and Roth (2012), Li et al. (2013) and Li et al. (2014). Only few approaches, like Ji and Grishman (2008) and Liao and Grishman (2010) go beyond sentences and even beyond documents in order to exploit richer context for the extraction of events.

While early systems usually predict triggers and arguments independently, more recent work employs joint inference, i.e., predicts triggers and arguments (or only arguments) jointly, e.g., Lu and Roth (2012), Li et al. (2013), and Li et al. (2014).

3 Approach

We make use of SEMAFOR, a state-of-the-art frame-semantic parsing system (Das et al., 2010)[1]. We retrain it to predict ACE events, i.e., triggers with event types and arguments for their roles, and make adaptions to better prepare it for event extraction. We call the new system SEMAFOR_E.

3.1 Trigger Classification

In order to classify triggers (single or multiple tokens), the original SEMAFOR uses a log-linear model. To cope with unknown triggers the model includes a latent variable iterating over triggers seen in training (called *hidden units*). At inference time, hidden units serve as prototypes for unknown words. The model is defined as

$$e_i = \operatorname*{argmax}_{e \in E_i} \sum_{l \in L_e} p_\theta(e, l \mid t_i, x). \qquad (1)$$

e_i is the best event type for trigger t_i according to the model. E_i is the set of observed event types for t_i. L_e is the set of triggers observed during training for event e. All $l \in L_e$ are called hidden units. $p_\theta(e, l \mid t_i, x)$ gives the probability of e and a hidden unit l given the trigger t_i and a sentence x. This probability is modeled as

$$p_\theta(e, l \mid t_i, x) = \frac{1}{Z} \exp \theta^\top g(e, l, t_i, x). \qquad (2)$$

This is a conditional log-linear model with a normalization constant Z, weights θ, and a vector-valued feature function g.

The model is biased towards classes with many hidden units. In order to illustrate this, imagine there is only one feature which does not depend on hidden units, e.g., if there is a named entity in the sentence. During inference, the sum in Equation 1 is computed. As a constant, Z is ignored during inference. The named entity feature would be active for every hidden unit, having the same weight in every iteration, because features are always evaluated inside the sum. Then, the sum is not meaningful anymore, because the event with the most hidden units wins. This bias affects both, frame-semantic parsing and event extraction.

In order to weaken the bias we propose to separate features which actually depend on hidden units, e.g., because they capture lexical similarity to some of them, from features which do not, like the named entity feature. Then, inference is performed as

$$e_i = \operatorname*{argmax}_{e \in E_i} \sum_{l \in L_e} \exp \theta^\top g'(e, l, t_i, x) + \exp \theta^\top g^\star(e, t_i, x). \qquad (3)$$

g' is a function for features depending on hidden units, $g^\star$ is a function for the remaining features. In this way, activation frequencies of features become meaningful. However, the model is still biased towards events with many hidden units. This is problematic, because the distribution of triggers over events is diverse and arbitrary. The number of hidden units does not necessarily correlate with occurrence probabilities of events. On the other

[1] `http://www.ark.cs.cmu.edu/SEMAFOR/`; we use version 2.1, without semi-supervised extensions or dual decomposition.

hand, the idea of known triggers being prototypes for events is appealing, therefore we did not change this part of the model.

3.2 Argument Classification

The argument model predicts the best argument A_i for every role r_k of an event e_i given a set of spans S. In our experiments, spans are extents of gold mentions, including the empty span. The argument-role mapping is defined as

$$A_i(r_k) = \operatorname*{argmax}_{s \in S} p_\psi(s \mid r_k, e_i, t_i, x). \quad (4)$$

Again, a conditional log-linear model with weights ψ, a normalization constant Z, and a feature function h is used to model p_ψ:

$$p_\psi = \frac{1}{Z} \exp \psi^\top h(s, r_k, e_i, t_i, x). \quad (5)$$

3.3 Adaptions

Based on our hypothesis that event extraction is structurally identical to frame-semantic parsing, we retrain SEMAFOR to predict ACE events. While the structure of the tasks may be identical, their behavior is not. It does not suffice to convert the ACE data to the right format and retrain the model.

There are two important differences between frame-semantic parsing and event extraction. First, in frame-semantic parsing, there is no 'null class' for triggers. A trigger may indicate multiple frames, but it always invokes one of them. In event extraction, we have potential triggers, which may or may not invoke events. Second, most event arguments are defined based on entity types. ACE distinguishes between the entity types person, organization, geopolitical entity, location, facility, vehicle, and weapon. For frame-semantic parsing, no such restriction in entity type exists. Thus, we need to introduce entity type features to tackle argument classification for event extraction. Such features are also useful for the trigger model.

One way to allow potential triggers to be classified as non-triggers is to introduce a null class to the event types. Each trigger in the training data also becomes a trigger of the null class (or *null event*). If a null event is triggered, we filter it out. Note that having a class with so many triggers biases our model towards it (Section 3.1). A less biased way would be to introduce a 'null version' of every event type, having the same triggers. However, we would have to predict twice as much classes (66 instead of 33). Having only one null class better exploits the limited training data. Furthermore, biasing SEMAFOR$_{\text{E}}$ towards null events is acceptable because there are considerably more null events than events.

Allowing all triggers from the training data in prediction hurts performance, mainly due to triggers which coincide with high-frequency words like "be". In order to prune the trigger set we compute a score for each trigger, catching its distribution among events and non-events: $s(t) = f_e/(f_e+f_n)d$. f_e is the frequency of t as an event trigger, f_n is the frequency of t in non-events, and d is the number of events t is a trigger of. The measure prefers triggers which are frequently triggers for only a few events. We filter all triggers with $s < 0.01$[2].

Finally, we changed the learning algorithm from the maximum entropy to the perceptron framework. This was done because the perceptron gives better performance for SEMAFOR$_{\text{E}}$ and is considerably faster, e.g., the argument model can be trained in a few seconds instead of several hours. The new models have a simpler form because we do not have to compute probabilities anymore. The new trigger model is defined as

$$e_i = \operatorname*{argmax}_{e \in E_i} \sum_{l \in L_e} \theta^\top g'(e, l, t_i, x) + \theta^\top g^\star(e, t_i, x). \quad (6)$$

The new argument model is defined as

$$A_i(r_k) = \operatorname*{argmax}_{s \in S} \psi^\top h(s, r_k, e_i, t_i, x). \quad (7)$$

Weights θ and ψ are learned using a variant of the averaged perceptron (Collins, 2002), where we store feature vectors only after each pass through the training data.

3.4 Features

For the trigger model, SEMAFOR's features include lemmas (of trigger tokens and of the head governor), dependencies of the head, if the head is equal to or has semantic relations with any hidden unit, as well

[2]The threshold was determined on development data.

as the type of these relations[3]. Additionally, we include unigrams and bigrams around the trigger in a window of two. Following Li et al. (2013), we also look at the mention nearest to the trigger. We include its entity type and its string representation as features.

Potential triggers are compared to hidden units by semantic relations. We extend this by incorporating measures of semantic similarity. We compare tokens in the actual sentence with tokens of all sentences the actual hidden unit appeared in (in the training data) and with tokens of all sentences all hidden units of the actual frame appeared in. The comparison is made in terms of cosine similarity.

SEMAFOR's features for the argument model characterize the actual span (its length, tokens, and head dependencies), the voice and string representation of the trigger, and the dependency path between span and trigger heads. Additionally, we include the token before the argument and its part-of-speech, and all tokens and parts-of-speech between argument and trigger as features. Following Li et al. (2013), we also use as features the type of the entity the actual span represents, if it is the only mention of its entity type, or the nearest to the trigger.

4 Experiments

We trained SEMAFOR$_{\text{E}}$ on the English ACE 2005 data. We followed Li et al. (2014) and removed the two smallest and most informal parts of the data, namely 'conversational telephone speech' and 'Usenet newsgroups'. From the remaining 511 documents, 351 are used for training, 80 for development, and 80 for testing.

We follow standard evaluation procedures for triggers and events (Ji and Grishman, 2008). A trigger is correct, if its span and event type match a reference trigger. An argument is correct, if its span, event type, and role match a reference argument.

Table 1 summarizes results for SEMAFOR$_{\text{E}}$ and a state-of-the-art system for event extraction (Li et al., 2013). To make a fair comparison, we report the numbers of their pipeline version, i.e., predicting trigger and arguments sequentially, as we do. Both systems use gold mentions and gold entity types. For SEMAFOR$_{\text{E}}$, we excluded all nested mentions of the same type: From "said [president [Obama]]", the inner span would be excluded.

SEMAFOR$_{\text{E}}$'s recall is comparable to Li et al. (2013). However, their system gives a higher precision for both subtasks. We believe that the higher precision of their argument model comes from the higher precision of their trigger model. Similarly, the lower precision of SEMAFOR$_{\text{E}}$'s argument model is due to the lower precision of its trigger model. Because of this, SEMAFOR$_{\text{E}}$ is a few F_1 points below Li et al. (2013).

We note that there is only a minor drop in performance when comparing numbers for the development and test sets. This indicates that SEMAFOR$_{\text{E}}$'s performance is robust.

The biggest error source for trigger classification is missing triggers. The second biggest error source is confusion of events with null events. Consider the following example: "Saba hasn't delivered yet". SEMAFOR$_{\text{E}}$ predicted a null event for the trigger "delivered" instead of the right BE-BORN event. The context it had to analyze did not suffice to overcome its bias towards null events. Even for humans it seems hard to infer the right event type here. One would need to know that "Saba" refers to a pregnant woman, which could be inferred from the document. However, the sentence alone does not provide enough information.

The biggest error source for argument classification is error propagation from the trigger model. The second major error source is the local prediction of arguments. It seems better to predict triggers and arguments jointly in order to weaken error propagation (Li et al., 2013; Li et al., 2014). For example, SEMAFOR$_{\text{E}}$ finds a START-ORG event for the trigger "set up" in the following sentence: "At the site, equipment has been set up to test conventional explosives [...]". In such cases, the model would need to know that the argument "equipment" cannot fill the *org* role of START-ORG because it is no organization. Inferring triggers and arguments jointly would enable SEMAFOR$_{\text{E}}$ to better prevent such errors.

5 Conclusions and Future Work

Based on the hypothesis that frame-semantic parsing and event extraction are structurally identical, we retrained a state-of-the-art frame-semantic pars-

[3] Semantic relations come from WordNet (Fellbaum, 1998)

	Triggers			Arguments		
	P	R	F_1	P	R	F_1
$SEMAFOR_E$ dev	65.8	57.8	61.6	57.0	32.4	41.3
$SEMAFOR_E$ test	62.6	56.8	60.0	53.5	33.3	41.0
Li et al. (2013)	74.5	59.1	65.9	65.4	33.1	43.9

Table 1: Evaluation results for $SEMAFOR_E$ on the development and test sets compared to a state-of-the-art system.

ing system for event extraction. We presented the adaptions in prediction classes and features needed to make the system better suited for the more restrictive task of event extraction. We also described a bias in the trigger classification model and proposed a feature factorization which is better suited for this model. As the evaluation shows, the retrained system can rival the state-of-the-art in event extraction.

For future work, we plan to incorporate mention detection into $SEMAFOR_E$. SEMAFOR's segmentation approach is not suited for event extraction because it produces too many argument candidates. Furthermore, error analysis and evaluation suggest that we need to predict triggers and arguments jointly. We also plan to go beyond sentences and search for larger contexts which may be relevant for event extraction. These changes may also be beneficial for frame-semantic parsing.

Acknowledgments

This work has been funded by the Klaus Tschira Foundation, Heidelberg, Germany. The first author has been supported by a HITS Ph.D. scholarship. We would like to thank our colleagues Sebastian Martschat, Daraksha Parveen, Nafise Moosavi, Yufang Hou and Mohsen Mesgar who commented on earlier drafts of this paper.

References

David Ahn. 2006. The stages of event extraction. In *Proceedings of the Workshop on Annotating and Reasoning about Time and Events,* Sydney, Australia, 23 July 2006, pages 1–8.

Michael Collins. 2002. Discriminative training methods for Hidden Markov Models: Theory and experiments with perceptron algorithms. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing,* Philadelphia, Penn., 6–7 July 2002, pages 1–8.

Dipanjan Das, Nathan Schneider, Desai Chen, and Noah A. Smith. 2010. Probabilistic frame-semantic parsing. In *Proceedings of Human Language Technologies 2010: The Conference of the North American Chapter of the Association for Computational Linguistics,* Los Angeles, Cal., 2–4 June 2010, pages 948–956.

Dipanjan Das, Desai Chen, André F. T. Martins, Nathan Schneider, and Noah A. Smith. 2014. Frame-semantic parsing. *Computational Linguistics*, 40(1):9–56.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, Mass.

Charles J. Fillmore, Christopher R. Johnson, and Miriam R. L. Petruck. 2003. Background to FrameNet. *International Journal of Lexicography*, 16(3):235–250.

Ralph Grishman, David Westbrook, and Adam Meyers. 2005. NYU's English ACE 2005 system description. Technical report, Department of Computer Science, New York University, New York, N.Y.

Heng Ji and Ralph Grishman. 2008. Refining event extraction through cross-document inference. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies,* Columbus, Ohio, 15–20 June 2008, pages 254–262.

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 (Volume 1: Long Papers),* Sofia, Bulgaria, 4–9 August 2013, pages 73–82.

Qi Li, Heng Ji, Yu Heng, and Sujian Li. 2014. Constructing information networks using one single model. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing,* Doha, Qatar, 25–29 October 2014, pages 1846–1851.

Shasha Liao and Ralph Grishman. 2010. Using document level cross-event inference to improve event extraction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics,* Uppsala, Sweden, 11–16 July 2010, pages 789–797.

Wei Lu and Dan Roth. 2012. Automatic event extraction with structured preference modeling. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers),* Jeju Island, Korea, 8–14 July 2012, pages 835–844.

Christopher Walker, Stephanie Strassel, Julie Medero, and Maeda Kazuaki. 2006. ACE 2005 multilingual training corpus. LDC2006T06, Philadelphia, Penn.: Linguistic Data Consortium.