Exploiting a more global context for event detection through bootstrapping

Dorian Kodelja, Romaric Besancon, Olivier Ferret

Exploiting a More Global Context for Event Detection through Bootstrapping*

Dorian Kodelja ✉, Romaric Besançon, and Olivier Ferret

CEA, LIST, Laboratoire Analyse Sémantique Texte et Image,
Gif-sur-Yvette, F-91191 France.
{dorian.kodelja,romaric.besancon,olivier.ferret}@cea.fr

Abstract. Over the last few years, neural models for event extraction have obtained interesting results. However, their application is generally limited to sentences, which can be an insufficient scope for disambiguating some occurrences of events. In this article, we propose to integrate into a convolutional neural network the representation of contexts beyond the sentence level. This representation is built following a bootstrapping approach by exploiting an intra-sentential convolutional model. Within the evaluation framework of TAC 2017, we show that our global model significantly outperforms the intra-sentential model while the two models are competitive with the results obtained by TAC 2017 participants.

Keywords: Information extraction · Event detection · Global context

1 Introduction

In some domains, such as journalism, the notion of event is particularly important and can be a central dimension for guiding search among documents [7]. Detecting events from texts is a necessary step for implementing such an approach. In this article, we consider supervised event detection, which consists in identifying in texts the mentions of *a priori* known event types, *i.e.* the word or the sequence of words indicating the presence of a particular type of events. Most of the current approaches for this task are based on neural models, either convolutional [2,17], recurrent [16] or mixing the two kinds of models [4]. Moreover, the best systems of the recent evaluation campaigns for this task, such as TAC Event Nugget 2017, are based on such models. These models successfully identify a significant part of event mentions but still fail when the local context is too ambiguous for discriminating between two types of events or deciding if an event mention is actually present. For instance, in the following example:

"[...] according to **leaked documents**. I don't trust them AT ALL [...], so I will have to read these **cables**$_{[broadcast]}$ myself."

the local sentence context is not sufficient for disambiguating the word *cables* as a trigger for a *Broadcast* event while looking at previous sentences would show that *cables* is related to the expression *leaked documents*, which is more directly linked to a *Broadcast* event. Performing such disambiguation requires

* Work partly supported by ANR under project ASRAEL (ANR-15-CE23-0018).

exploiting contexts beyond the scope of sentences. This perspective has already been explored by [3] by adding to the input of a BiLSTM model for trigger extraction the representation of the overall document computed by the method of [10]. The underlying hypothesis is that integrating such representation accounts for the fact that a document related to the topic of war is more likely to contain *Die* or *Attack* events than *Divorce* events. However, the document representation, in that case, is general. Very recently, [20] has extended this approach in a more integrated way by exploiting a hierarchical document embedding. Similarly, our approach aims at building a document representation specifically linked to the target task but we adopt a simpler approach by relying on bootstrapping: a model focusing on a very local context is first applied to the considered document; then, its local predictions are aggregated for building a document context vector. This vector is finally exploited by a new extraction model we define. Previously, [11] introduced a global classifier to apply a second pass on the input corpus and detect ambiguous triggers missed by the local classifier. The global classifier only used the candidate word and a binary vector informing about the detection of at least one event of each event type by the local classifier. On the opposite, our system, which uses the more informative estimated distribution of the number of events for each type, is not only able to detect missed event but also to reject previously detected spurious triggers.

Our experiments on the TAC Event Nugget 2017 data show that this new model significantly outperforms our state-of-the-art local model.

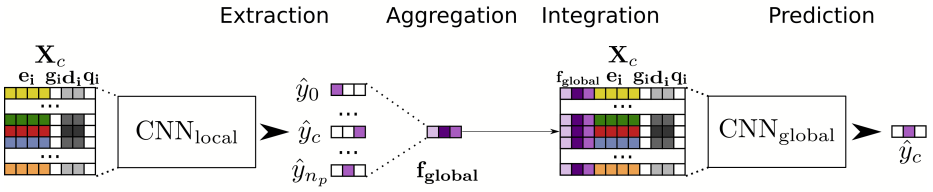

Fig. 1. Generation and integration of the global context representation

2 Method

In this article, we aim at detecting event mentions (triggers) in text and categorize them into predefined types. We consider the 38 event types defined in the DEFT Rich ERE taxonomy [1] used in the Event Nugget evaluation of the TAC campaigns [15]. Since most of the annotated triggers are single tokens [19], we only consider mono-token triggers. While this simplification does not affect performance significantly, it makes the model simpler and allows the introduction of a positional vector, which has a significant impact on results [18].

Figure 1 gives an overview of our integration of a global context in a convolutional neural network (CNN) through bootstrapping. First, a local model $\mathrm{CNN}_{\mathrm{local}}$ is trained to predict an event label for each word of a document. These

labels are then aggregated at a specific level (in Figure 1, labels are aggregated at the document level) and integrated into a new model. The following sections present the local and global models in more detail.

2.1 Local Event Detection Model

At the local level, our event detection model relies on a CNN based on the architecture introduced in [18]. We successively consider each token in each sentence as a candidate mention. This mention is represented by a fixed-size local context centered on the mention. We perform padding to complete the sequence when the local context goes beyond sentence boundaries. Let i_c be the index of the candidate mention and w the window size. We define $\mathbf{i_c} = [i_{c-w}, i_{c-w+1}, \ldots, i_c, \ldots, i_{c+w-1}, i_{c+w}]$ as the index vector centered on i_c. This vector is then transformed into a real-valued matrix $\mathbf{X}_c = [\mathbf{x_{c-w}}, \mathbf{x_{c-w+1}}, \ldots, \mathbf{x_c}, \ldots, \mathbf{x_{c+w-1}}, \mathbf{x_{c+w}}]$ by replacing each index i with its vector representation $\mathbf{x_i} = [\mathbf{e_i}, \mathbf{d_i}, \mathbf{g_i}, \mathbf{q_i}]$ using the concatenation of the following representations:

Word embedding $\mathbf{e_i}$ This distributed representation of token t_i at position i is pre-trained on a large corpus to capture its semantic and syntactic properties [14].

Position embedding $\mathbf{d_i}$ This vector encodes the relative distance from the token t_i to the candidate t_{i_c}. This embedding matrix is initialized randomly.

Dependency vector $\mathbf{g_i}$ The size of this vector corresponds to the number of considered dependencies[1]. If a dependency of a given type is found between t_i and t_{i_c}, the corresponding value is set to 1.

Chunk embedding $\mathbf{q_i}$ This vector encodes the type of syntactic chunk containing the token t_i, using a BIO encoding scheme: the chunks are computed by a *chunker*[2] from the syntactic tree provided by Stanford CoreNLP. This embedding matrix is initialized randomly.

A convolution layer is applied to the input matrix $\mathbf{X}_c$, made of multiple filters of different sizes. A global max-pooling is performed to get a single value for each filter. This provides a representation of the candidate in its local context, learned by the convolutional neural network. This local representation $\mathbf{f_{softmax}} = [\mathbf{f_{pooling}}]$ is then fed into a softmax layer for computing the probability distribution of the different event classes for the candidate. Finally, the highest probability class $\hat{y}_c$ is taken as prediction. To improve generalization, a dropout is applied between the embedding and the convolutional layers.

2.2 From Local to Global Model

As mentioned in Section 2, our objective is to improve the performance of our local model by integrating a representation of a more global context. Moreover, we propose to generate such global representation in connection with our target task by using bootstrapping: we first apply the $\text{CNN}_{\text{local}}$ model presented above to a document. The prediction $\hat{y}_c$ for each token is then extracted and aggregated

[1] We use the basic dependencies provided by par Stanford CoreNLP [13].

[2] `https://github.com/mgormley/concrete-chunklink`

at a given level of context through *sum-pooling*, leading to a histogram of the detected event types used as the representation $\mathbf{f_{global}}$ of the global context.

Two main factors have to be defined for implementing this approach: the level of context to take into account and the place in the neural network where the representation of this context is integrated. Three levels are considered for the first factor: sentence-wide (*sentence*), a three sentence window centered on the current sentence (*wide*) or document-wide (*doc*). We use the following notation to refer to these three aggregation levels: $\mathbf{f_{global}} = \mathbf{f_{[doc/wide/sentence]}}$. Concerning the second factor, the global context representation can be integrated by concatenation either to the input matrix $\mathbf{X}_c$ by redefining $\mathbf{x_i} = [\mathbf{e_i}, \mathbf{d_i}, \mathbf{g_i}, \mathbf{q_i}, \mathbf{f_{global}}]$ or before the softmax layer: $\mathbf{f_{softmax}} = [\mathbf{f_{pooling}}, \mathbf{f_{global}}]$. Finally, 6 model configurations can be distinguished by choosing the aggregation and integration levels, with the following notation: $\text{CNN}_{\text{[doc/wide/sentence]-[input/softmax]}}$.

3 Experiments and Evaluation

Parameters and Resources In our experiments, we use the 300 dimension word embeddings pre-trained on Google News using *word2vec* that we modify during training. The size of the chunk and position embeddings is set to 50 and the dropout probability to 0.8, based on preliminary experiments. For each window size (2,3,4,5), 150 filters are used. We apply a hyperbolic tangent non-linearity to the resulting 600 filters. Following [8], our models are trained by stochastic gradient descent (SGD) using the Adadelta optimizer, a gradient clipping of the *l2* norm equal to 3 and a mini-batch size set to 50. The number of epochs is determined by early stopping on the development set. The results are averaged micro F1 scores, computed by the TAC 2017 scorer, on 10 runs.

Our training set is built by merging the DEFT_RICH_ERE_R2_V2 (LDC2015E68), DEFT_RICH_ERE_V2 (LDC2015E29) and TAC 2015 (LDC2017E02) datasets. Our development set comes from the TAC 2016 Event Nugget campaign (LDC2017E02) and we test our model on the data of the TAC 2017 Event Nugget campaign (LDC2017E02). Starting from TAC 2016, the datasets are only focused on the most difficult event types, which reduces the number of possible labels from 38 to 19. The datasets also contain few occurrences of mentions annotated with multiple distinct events types. Since most of these cases correspond to one configuration among three – *Attack/Die*, *Transfer-Money/Transfer-Ownership*, *Attack/Injure* – we introduce 3 new hybrid event types to avoid dealing with a multi-label classification task. We train our model with 42 classes (*other* class and hybrid classes included) but we skip the predictions of the removed types during validation and test. Similarly, the global vector only aggregates the predictions from the test types. Finally, the results we present rely on the best normalization of the global context vector for each configuration, namely no normalization for the $\mathbf{f_{[wide/sentence]}}$ vectors while the $\mathbf{f_{doc}}$ was reduced and centered prior to training.

Influence of the Aggregation Level Our first experiments concern the aggregation level used for the global representation. Aggregating the predictions at the sentence level could help to reduce intra-sentence ambiguities while a larger

Table 1. Performance on the TAC 2016 development set depending on the aggregation level. Results are averaged over 10 runs. ‡ indicates models that are significantly better than $\text{CNN}_{\text{local}}$ ($p < 0.01$ for a bilateral t-test over the 10 runs)

methods	P	R	F
$\text{CNN}_{\text{doc-input}}$	**52.71**	47.95	**50.2** ‡
$\text{CNN}_{\text{wide-input}}$	52.00	47.6	49.69
$\text{CNN}_{\text{sentence-input}}$	49.83	49.49	49.66
$\text{CNN}_{\text{local}}$	46.42	**52.04**	49.06
$\text{CNN}_{\text{doc-input-gold}}$	**54.85**	51.02	**52.83** ‡
$\text{CNN}_{\text{sentence-input-gold}}$	54.21	47.58	50.68 ‡

Table 2. Performance on the TAC 2017 test set. † indicates ensemble models. ‡ indicates in the lower part of the table models that are significantly better than $\text{CNN}_{\text{local}}$ ($p < 0.01$ for a bilateral t-test over the 10 runs)

Methods	max			average over 10 runs		
	P	R	F	P	R	F(std)
BiLSTM CRF (Jiang) †	**56.83**	**55.57**	**56.19**	-	-	-
BiLSTM-SMO (Makarov) †	52.16	48.71	50.37	-	-	-
CNN (Kodelja)	54.23	46.59	50.14	-	-	-
$\text{CNN}_{\text{local}}$	52.21	49.55	50.84	51.90	48.92	50.36 (0.33)
$\text{CNN}_{\text{doc-input}}$	**59.13**	45.37	51.34	**58.07**	45.43	50.95 (0.41) ‡
$\text{CNN}_{\text{doc-softmax}}$	52.87	**50.35**	**51.58**	53.12	**49.61**	**51.30**(0.22) ‡
$\text{CNN}_{\text{doc-input\_softmax}}$	55.72	47.08	51.04	57.62	45.09	50.58 (0.49)
$\text{CNN}_{\text{PV-DM}}$	53.20	47.40	50.10	53.54	46.92	49.98 (0.41)

context could be beneficial for inter-sentence ambiguities. Table 1 compares results for different sizes while integrating this representation at the input level.

We observe that each configuration yields an improvement compared to $\text{CNN}_{\text{local}}$ but this improvement is significant only for $\text{CNN}_{\text{doc-input}}$. Since the local model used for building the global representation is not perfect, one possible interpretation of this finding is that errors tend to dilute when the local model is applied to a wider context. We ran a complementary experiment using the gold event mentions to generate the global representation (see the last two lines of Table 1) and observed that the document level aggregation is also the best choice in this configuration, confirming that this level intrinsically leads to a better global representation for the event extraction task.

Comparison to the State-of-the-Art Our last experiments, reported in Table 2, compare the different options for the integration of the global representation (*input*/*softmax*) to the 3 best models of the event detection track of TAC 2017:

1. **BiLSTM CRF ensemble**: [6] use an ensemble of 10 BiLSTM combined by a voting strategy. Since their neural models tend to have a good recall at the expense of precision, they combine this ensemble with a Conditional Random Field classifier to improve precision. For the BiLSTM, only word embeddings

are used while the CRF use multiple features such as tokens, lemmas, roots, named entities, and POS tags.
2. **BILSTM-SMO**: [12] introduce a BiLSTM with a softmax margin objective [5]. This objective aggressively penalizes false negatives to counterbalance the scarcity of positive samples in training data. An ensemble of 5 networks is used as well as hybrid types.
3. **CNN**: the model of [9] is similar to our local model, *i.e.* a CNN using word, position and chunk embeddings and syntactic dependencies as inputs. The main difference is the absence of hybrid types for modeling multi-type tokens.

We also compare our approach to the integration of a generic document vector in the local model, following [3] (noted $\mathbf{CNN_{PV\text{-}DM}}$). This vector of size 100 is generated using the PV-DM model [10]. Unlike our global representation, it is not specific to the task. We optimize the same integration hyperparameters as for our model, namely the level of integration and the choice of normalization. The best configuration integrates reduced and centered vectors at the softmax level.

It is difficult to compare our contributions to [6] and [12] since they are ensemble methods while we use a single model approach. [6] is even a rather complex ensemble method based on two different architectures combined with a specific heuristic. Furthermore, only the best score of the two models is available while average scores over several runs are more reliable [19]. However, we can note that $\text{CNN}_{\text{doc-input}}$ and $\text{CNN}_{\text{doc-softmax}}$ significantly outperform not only our local model but also the ensemble method of [12], with an advantage of *softmax* over *input* for integrating the global representation. The breakdown analysis of our gain between trigger span detection (+0.65) and trigger classification (+0.89) indicates that our representation mostly helps to filter out ambiguous non-triggers while marginally improving the classification part. Finally, we can observe that the integration of the representation proposed by [3] leads to a decrease in performance. The absence of correlation between the representation and the task is a possible explanation of this observation.

4 Conclusion and Perspectives

In this article, we propose a new representation to exploit a more global context for event extraction. This method is based on bootstrapping and more specifically on the aggregation of local predictions for building a document representation exploited by a global model. We show on the TAC 2017 evaluation data that integrating such global representation significantly increases the results of our initial state-of-the-art local model and can even outperform a BiLSTM ensemble model. We also show that a document representation linked to the target task is more effective than relying on a general document representation. While this model only exploits the output of an initial model, our work could be extended by integrating richer context representations such as internal representations produced by our initial CNN model or a document representation built on a related task trained from a large set of data following a multi-task perspective.

References

1. Bies, A., Song, Z., Getman, J., Ellis, J., Mott, J., Strassel, S., Palmer, M., Mitamura, T., Freedman, M., Ji, H., O'Gorman, T.: A Comparison of Event Representations in DEFT. In: Fourth Workshop on Events. pp. 27–36. San Diego, California (June 2016), http://www.aclweb.org/anthology/W16-1004
2. Chen, Y., Xu, L., Liu, K., Zeng, D., Zhao, J.: Event Extraction via Dynamic Multi-Pooling Convolutional Neural Networks. In: 53^{rd} Annual Meeting of the Association for Computational Linguistics and 7^{th} International Joint Conference on Natural Language Processing (ACL-IJCNLP 2015). pp. 167–176. Beijing, China (2015)
3. Duan, S., He, R., Zhao, W.: Exploiting Document Level Information to Improve Event Detection via Recurrent Neural Networks. In: Eighth International Joint Conference on Natural Language Processing (IJCNLP 2017). pp. 352–361. Taipei, Taiwan (2017), https://aclanthology.coli.uni-saarland.de/papers/I17-1036/i17-1036
4. Feng, X., Huang, L., Tang, D., Ji, H., Qin, B., Liu, T.: A Language-Independent Neural Network for Event Detection. In: 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016). pp. 66–71. Berlin, Germany (2016). https://doi.org/10.18653/v1/P16-2011, https://aclanthology.coli.uni-saarland.de/papers/P16-2011/p16-2011
5. Gimpel, K., Smith, N.: Softmax-Margin CRFs: Training Log-Linear Models with Cost Functions. In: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL HLT 2010). pp. 733–736. Los Angeles, California (2010)
6. Jiang, S., Li, Y., Qin, T., Meng, Q., Dong, B.: SRCB Entity Discovery and Linking (EDL) and Event Nugget Systems for TAC 2017. In: Text Analysis Conference (TAC) (2017)
7. Jorge, A.M., Campos, R., Jatowt, A., Nunes, S. (eds.): First Workshop on Narrative Extraction From Text (Text2Story 2018). Grenoble, France (2018)
8. Kim, Y.: Convolutional Neural Networks for Sentence Classification. In: EMNLP (2014)
9. Kodelja, D., Besançon, R., Ferret, O., Le Borgne, H., Boros, E.: CEA LIST Participation to the TAC 2017 Event Nugget Track. In: Text Analysis Conference (TAC) (2017)
10. Le, Q., Mikolov, T.: Distributed Representations of Sentences and Documents. In: 31st International Conference on International Conference on Machine Learning (ICML 2014). pp. 1188–1196. Beijing, China (2014)
11. Liao, S., Grishman, R.: Using Document Level Cross-Event Inference to Improve Event Extraction. In: ACL (2010)
12. Makarov, P., Clematide, S.: UZH at TAC KBP 2017: Event Nugget Detection via Joint Learning with Softmax-Margin Objective. In: Text Analysis Conference (TAC) (2017)
13. Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S.J., McClosky, D.: The Stanford CoreNLP Natural Language Processing Toolkit. In: 52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014), system demonstrations. pp. 55–60 (2014)
14. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed Representations of Words and Phrases and Their Compositionality. In: 26th International Conference on Neural Information Processing Systems (NIPS 2013). pp. 3111–3119. Lake Tahoe, Nevada (2013)

15. Mitamura, T., Liu, Z., Hovy, E.: Events Detection, Coreference and Sequencing: What's next? Overview of the TAC KBP 2017 Event Track. In: Text Analysis Conference (TAC) (2017)
16. Nguyen, T.H., Cho, K., Grishman, R.: Joint Event Extraction via Recurrent Neural Networks. In: 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT 2016). pp. 300–309. San Diego, California (2016)
17. Nguyen, T.H., Grishman, R.: Event Detection and Domain Adaptation with Convolutional Neural Networks. In: 53^{rd} Annual Meeting of the Association for Computational Linguistics and 7^{th} International Joint Conference on Natural Language Processing (ACL-IJCNLP 2015). pp. 365–371. Beijing, China (2015)
18. Nguyen, T.H., Grishman, R., Meyers, A.: New York University 2016 System for KBP Event Nugget: A Deep Learning Approach. In: Text Analysis Conference (TAC) (2016)
19. Reimers, N., Gurevych, I.: Reporting Score Distributions Makes a Difference: Performance Study of LSTM-networks for Sequence Tagging. In: 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017). pp. 338–348. Copenhagen, Denmark (2017)
20. Zhao, Y., Jin, X., Wang, Y., Cheng, X.: Document Embedding Enhanced Event Detection with Hierarchical and Supervised Attention. In: 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018). pp. 414–419. Association for Computational Linguistics (2018)