Better exploiting BERT for few-shot event detection

Aboubacar Tuo, Romaric Besancon, Olivier Ferret, Julien Tourille

Better Exploiting BERT for Few-shot Event Detection*

Aboubacar Tuo (✉), Romaric Besançon, Olivier Ferret, and Julien Tourille

Université Paris-Saclay, CEA, List, F-91120, Palaiseau, France
{aboubacar.tuo,romaric.besancon,olivier.ferret,julien.tourille}@cea.fr

Abstract. Recent approaches for event detection rely on deep supervised learning, which requires large annotated corpora. Few-shot learning approaches, such as the meta-learning paradigm, can be used to address this issue. We focus in this paper on the use of prototypical networks with a BERT encoder for event detection. More specifically, we optimize the use of the information contained in the different layers of a pre-trained BERT model and show that simple strategies for combining BERT layers can outperform the current state-of-the-art for this task.

Keywords: Few-shot Event Detection · Meta-learning · BERT.

1 Introduction

Event extraction aims to automatically extract structured information about events from text. It can be compared to filling a form with entities, each field of this form corresponding to an argument of the event. Earlier methods for event extraction were based on handcrafted rules [1]. These methods were gradually replaced by machine learning algorithms with the development of statistical learning and neural networks in recent years. In this context, [16] proposes a structured prediction model based on numerous lexico-syntactic features, [20] uses convolutional networks to exploit contextual information, [19] defines models based on recurrent networks, and [18], [21], and [31] exploit graph convolution models to capture syntactic dependencies between different parts of sentences.

While the objective of event extraction is to identify all arguments connected to a particular event, datasets since ACE 2005 [30] have introduced the concept of *event trigger*, designating the word or group of words that indicate as clearly as possible the presence of an event in a sentence. The intention is to define a lexical anchor to help in the search for arguments. We focus in this paper on the detection and the classification of those triggers according to a restricted set of predefined types, a task generally called *Event Detection* or *Trigger Detection*.

Supervised learning methods are costly since they require large corpora that are manually annotated. Hence, a current challenge is to investigate methods

* This publication was made possible by the use of the FactoryIA supercomputer, financially supported by the Île-de-France Regional Council.

that reduce the development cost of these systems. In this context, we investigate Few-Shot Learning (FSL) for trigger detection.

Few Shot Event Detection (FSED) has been recently the focus of several studies with various configurations: generalization of models to new types of events using keyword lists [2,14], data enrichment with external resources [7], Zero-Shot Learning with the use of class descriptions or external resources [33], and FSL [26,3,4]. Other studies have also focused on Few-Shot Event Classification, which restricts FSED to assigning an event type to a candidate trigger already identified in a sentence [13,15,6].

In this paper, our contribution focuses on a better exploitation of the BERT language model representations for FSED, more specifically by studying the importance of these different representations and by evaluating different ways to associate them.

2 Method

2.1 Problem Formulation

We cast FSED as a sequence labeling task [23], using the IOB format (*Inside Outside Beginning*), which can be addressed as a multi-class classification task.

Recent studies in FSL use meta-learning, which is often defined as *learning to learn*. The main idea behind meta-learning is to train models on several tasks, each with a limited number of instances, so that the learned model can quickly perform similar tasks on new data. The methods that have emerged in recent years for solving FSL tasks with the help of meta-learning fall into three main categories: model-based methods [32,25], optimization-based algorithms [10,22,24], and metric-based methods [29,28,27]. Among the latter, we adopted prototypical networks [27] in our study, similarly to most works about FSED.

We adopt the standard N-way, K-shot episodic formulation, as described in [29]. An episode $\mathcal{E}$ is composed of a *support set* and an associated *query set*. During each episode, the model is defined by relying on a subset $\mathcal{S}$ of the available labeled data, the support set, which contains N types of events and K annotated instances per type (K being generally small, e.g. 1, 5, or 10):

$$\mathcal{S} = \{(x_1^1, t_1^1, y^1), \ldots, (x_k^1, t_k^1, y^1), \ldots, (x_1^N, t_1^N, y^N), \ldots, (x_k^N, t_k^N, y^N)\}$$

where $x_i^n = \{w_1, \ldots, w_L\}$ is a sequence of tokens of length L containing a trigger of type n, t_i^n, the position of the trigger, and y^n, the corresponding sequence of labels. The training in this context is done by updating the weights of the model based on the prediction on the instances of the query set, which has the same structure as the support set.

2.2 Model Architecture

We use a Prototypical Network model with episodic learning to combine meta-learning and FSL. An overview of our model, composed of three modules, is presented in Figure 1.

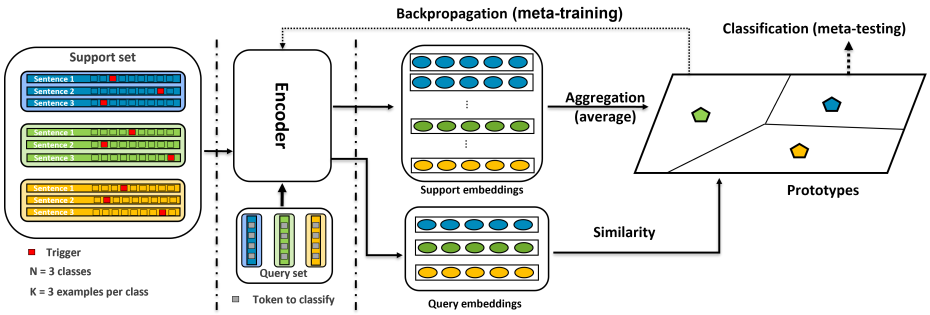


Fig. 1: Our few-shot event detection model based on prototypical networks.

Encoder module. Given a sentence $x = \{w_1, \ldots, w_L\}$ of length L, the objective of this module is to construct a representation e_i of each word $w_i \in x$ in a d-dimensional space. Importantly, during the episodic learning phase, only the encoder part of the model is actually updated. We use the BERT [8] language model as our encoder, with $H_i = [h_i^1, h_i^2, \ldots, h_i^{12}]$, its representations for the word w_i and $h_i^j \in \mathbb{R}^d$. The main objective of our work is to study different options for selecting and combining the 12 layers of BERT to obtain more relevant token representations for the event detection task. More precisely:

- **Average**: embedding e_i of the word w_i = average of the representations of m consecutive layers. $e_i = \frac{1}{m}\sum_{k=1}^{m} h_i^k$ or $e_i = \frac{1}{m}\sum_{k=12-m+1}^{12} h_i^k$ depending on whether the layers are aggregated from the first or the last layer.
- **Max-pool**: max-pooling on each dimension p for m consecutive layers. The p-th element of the embedding e_i is given by: $(e_i)_p = \max((h_i^1)_p, \ldots, (h_i^m)_p)$
- **Concat**: concatenation of m consecutive BERT layers $e_i = [h_i^1 \,||\, h_i^2 \,||\, \ldots ||h_i^m]$ or $e_i = [h_i^{12} \,||\, h_i^{11} \,||\, \ldots \,||\, h_i^{12-m+1}]$
- **Weighted**: linear combination of the 12 BERT layers. $e_i = \sum_{k=12}^{12} \alpha^k h_i^k$, where the α^k are randomly initialized and learned.
- **ATT**: linear combination of BERT layers for each dimension using an attention mechanism. The objective is to identify the most important layers for each dimension. The p-th element of e_i is given by $(e_i)_p = \sum_{k=1}^{12} \alpha^k (h_i^k)_p$ where the α^k are learnt from a linear combination of the 12 layers and a softmax normalization.

Prototypical module. The objective of this module is to build a prototype for each class and then, to classify new examples according to their similarity to these prototypes. We take the average of the examples in the support set as prototypes for each class, as proposed by [27]. Since we are using the IOB format, we build a prototype for classes B and I, as well as for class O (the *null* class for all event types), which refers to words that are not triggers of any event. Hence, we obtain $2N + 1$ prototypes for an episode composed of N types.

Classification module. This module classifies the words of a query sequence according to their similarity to the prototypes. The probability of a given word to belong to a particular class is computed according to its similarity to the class' prototype. The model is trained using the cross-entropy loss.

3 Experiments and Results

3.1 Experimental Setup

Evaluation dataset. We experiment on the FewEvent corpus [6] for FSED. This corpus is composed of 70,852 event mentions divided into 100 types. We use the same split as [4] for comparison purposes. This split includes 80 types in the training set, 10 types in the test set, and the remaining 10 types in the validation set.

Model parameters. We use the BERT-base pre-trained model as our encoder. To evaluate the impact of our modifications of this encoder, we rely on two models presented in [4]: **Proto-dot**, a prototypical model based on the dot product for its similarity function, which is our baseline model, and **PA-CRF**, an improvement of the previous model using CRFs (*Conditional Random Field*) [12] to estimate the transition probabilities between different IOB labels as proposed by [11]. The PA-CRF model is the main contribution of [4] and is, to our knowledge, the best performing model for FSED. We take as input 128-word sentences (with padding if needed) and train the model with a learning rate of 10^{-5}.

Evaluation. To test our model, we construct 3,000 episodes $\mathcal{E}^i \triangleq \{\mathcal{S}^i, \mathcal{Q}^i\}$ with N types of events randomly sampled for each episode. We then select K examples per class in the support set and one example per class in the query set. The examples in the support set are used to build prototypes and the examples in the query set are classified based on their similarity to these prototypes. We consider an event trigger to be correct if its type and position in the sentence are correctly predicted, as in previous work about event detection [4,5,17].

3.2 Results and Discussion

We compute the micro F1-score to evaluate the performance and report the means and standard deviations over 5 runs in Table 1 with different values of N and K. For the encoders **Average**, **Concat**, and **Max-pool**, we only consider the last 4 layers (as suggested in [8]) for the results reported in Table 1. We compare our model to our implementation of [4], which gives the best current performance on the same task (**BERT** line in Table 1) and report the results provided in their article (**BERT[Cong]** line in Table 1).

Whatever the model used (Proto-dot or PA-CRF), all the encoder improvements, except the **Concat** configuration for the 10 ways condition of PA-CRF, significantly improve the performance compared to the classical BERT encoder.

Table 1: Results: mean and standard deviation of the micro F1-score over 5 trials. **bold**, the best performance on average; underlined, the second best. * denotes the best statistically significant model compared to the second best using the significance test of [9].

Model	Encoder	5 ways 5 shots	5 ways 10 shots	10 ways 5 shots	10 ways 10 shots
Proto-dot	BERT [Cong]	58.82 ± 0.88	61.01 ± 0.23	55.01 ± 1.62	58.78 ± 0.88
	BERT	61.22 ± 0.90	60.84 ± 1.58	58.14 ± 1.69	59.85 ± 2.01
	average	64.34 ± 1.94	65.37 ± 0.66	61.85 ± 2.05	63.93 ± 1.08
	max-pool	64.10 ± 1.78	65.80 ± 0.91	61.15 ± 1.51	63.37 ± 1.03
	concat	61.99 ± 0.46	61.94 ± 0.97	57.47 ± 0.65	59.02 ± 1.39
	weighted	65.62 ± 1.55	**67.15 ± 0.88***	**62.63 ± 1.18***	**65.22 ± 0.98***
	ATT	**65.64 ± 0.90**	65.63 ± 0.46	62.22 ± 0.52	64.23 ± 0.99
PA-CRF	BERT [Cong]	62.25 ± 1.42	64.45 ± 0.49	58.48 ± 0.68	61.54 ± 0.89
	BERT	63.63 ± 2.01	63.66 ± 1.54	62.11 ± 1.58	62.47 ± 1.29
	average	65.09 ± 0.40	66.70 ± 0.45	62.32 ± 1.51	65.38 ± 1.71
	max-pool	63.95 ± 1.99	66.94 ± 1.20	61.74 ± 1.95	64.77 ± 1.84
	concat	64.30 ± 1.99	64.31 ± 1.80	62.01 ± 1.28	61.88 ± 1.05
	weighted	**66.26 ± 1.16***	**66.97 ± 0.95***	**63.90 ± 1.23***	**67.21 ± 1.27***
	ATT	63.65 ± 1.35	66.40 ± 1.03	62.41 ± 1.73	64.32 ± 1.64

Thus, a better exploitation of the information of the BERT model leads to outperform the improvements brought by the more sophisticated model of [4], representing the current state of the art.

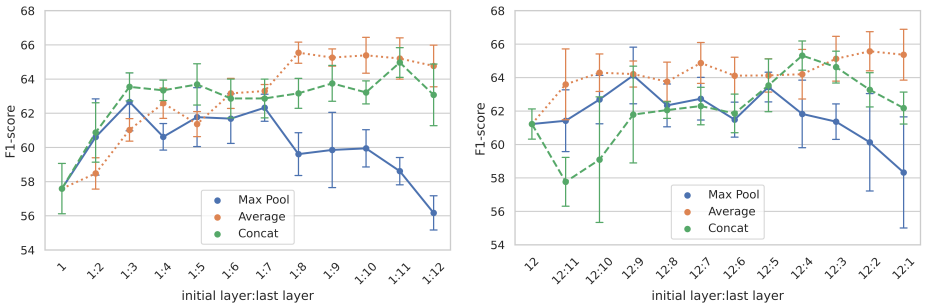
Among all the tested strategies, those allowing the model to learn the weights of the linear combination of BERT's layers generally yield better results, with the **Weighted** strategy proving to be the best in almost all configurations.

Finally, the fact that the gains observed for the Proto-dot model are also found for the PA-CRF model, which is a more elaborate version of the Proto-dot model, shows that the proposed improvements are complementary to those that can be made to the other modules (prototypical and classification modules).

Influence of the episodic formulation N ways k shots. We observe logically that the task is more difficult when the number of types (N) increases and that the results are improved with a larger number of annotated examples (K). Furthermore, we assess the robustness of the evaluation protocol by experimenting with different numbers of evaluation episodes for the encoder **Average** (between 500 and 5,000). The score variation between our different runs remains within a range of ± 0.5 points.

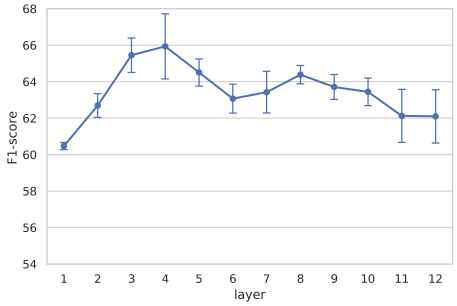
Layer analysis. We perform experiments to determine the influence of the number of layers selected for the encoders **Average**, **Concat**, and **Max-pool**, which do not have the capability to perform this selection on their own, contrary to **Weighted** and **ATT**. We report in Figure 2 the results obtained by these

three models for the *5 ways 5 shots* task, taking into account n successive layers starting from the first layer (Figure 2a) or the last one (Figure 2b).



(a) From first layers to last layers.

(b) From last layers to first layers.



(c) Performance by layer.

Fig. 2: Figures 2a and 2b present the influence of the number of layers selected for the encoders **Average**, **Concat**, and **Max-pool** on the performance of the model. Figure 2c presents the performance of the model for only one single layer.

We observe that the encoder **Average** is more stable than the other two and that its performance tends to increase steadily with the number of layers. Its best results are also competitive with the **Weighted** strategy, which shows that taking into account all the layers is beneficial, even with a very simple strategy such as averaging. We can also observe that the other strategies do not exploit all the information. This is particularly noticeable for **Max-pool**, which probably tends to increasingly smooth out the outstanding dimensions as the number of layers increases. Concerning **Concat**, it seems that the strategy creates large layered representations that are probably not very informative for the model.

In this context, Figures 2a and 2b also show that the combination of the last layers seems more interesting than the combination of the first ones. This observation may reflect the intrinsic presence of more useful information for the task in these layers, or simply be explained by a more important influence of the learning linked to the task at their level because of their closeness to the model

output. Finally, Figure 2c reports the limiting case of considering only one layer, with, from first to last layers, a strong increase of results until layer 4, which reaches a performance comparable to **Average** but with a larger variance, and then, a soft decrease.

4 Conclusions and Perspectives

In this article, we have studied different ways to better exploit the information contained in the BERT pre-trained model for the task of detecting events from a few examples. We have shown that the improvements brought by our proposals outperform the state-of-the-art results on this task. We plan to pursue the improvement of the encoder by studying other representation models and by enriching these representations with external knowledge or lists of examples of triggers. Furthermore, we will study how the improvements of the encoder can be combined with improvements of the prototypical and classification modules.

References

1. Ahn, D.: The stages of event extraction. In: Workshop on Annotating and Reasoning about Time and Events. pp. 1–8. Sydney, Australia (2006)
2. Bronstein, O., Dagan, I., Li, Q., Ji, H., Frank, A.: Seed-Based Event Trigger Labeling: How far can event descriptions get us? In: ACL-IJCNLP. pp. 372–376 (2015)
3. Chen, J., Lin, H., Han, X., Sun, L.: Honey or Poison? Solving the Trigger Curse in Few-shot Event Detection via Causal Intervention. arXiv:2109.05747 (2021)
4. Cong, X., Cui, S., Yu, B., Liu, T., Yubin, W., Wang, B.: Few-Shot Event Detection with Prototypical Amortized Conditional Random Field. In: Findings of ACL-IJCNLP. pp. 28–40. Online (2021)
5. Cui, S., Yu, B., Liu, T., Zhang, Z., Wang, X., Shi, J.: Edge-Enhanced Graph Convolution Networks for Event Detection with Syntactic Relation. In: Findings of EMNLP. pp. 2329–2339. Online (2020)
6. Deng, S., Zhang, N., Kang, J., Zhang, Y., Zhang, W., Chen, H.: Meta-Learning with Dynamic-Memory-Based Prototypical Network for Few-Shot Event Detection. In: WSDM. pp. 151–159. Houston, TX, USA (2020)
7. Deng, S., Zhang, N., Li, L., Hui, C., Huaixiao, T., Chen, M., Huang, F., Chen, H.: OntoED: Low-resource Event Detection with Ontology Embedding. In: ACL-IJCNLP. pp. 2828–2839. Online (2021)
8. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: NAACL-HLT. pp. 4171–4186 (2019)
9. Dror, R., Shlomov, S., Reichart, R.: Deep dominance - how to properly compare deep neural models. In: ACL. pp. 2773–2785 (2019)
10. Finn, C., Abbeel, P., Levine, S.: Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. arXiv:1703.03400 [cs] (2017), arXiv: 1703.03400
11. Hou, Y., Che, W., Lai, Y., Zhou, Z., Liu, Y., Liu, H., Liu, T.: Few-shot slot tagging with collapsed dependency transfer and label-enhanced task-adaptive projection network. In: ACL. pp. 1381–1393. Online (2020)

12. Lafferty, J.D., McCallum, A., Pereira, F.C.N.: Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In: ICML. pp. 282–289 (2001)
13. Lai, V., Dernoncourt, F., Nguyen, T.H.: Learning Prototype Representations Across Few-Shot Tasks for Event Detection. In: EMNLP. pp. 5270–5277 (2021)
14. Lai, V.D., Nguyen, T.: Extending Event Detection to New Types with Learning from Keywords. In: W-NUT 2019. pp. 243–248. Hong Kong, China (2019)
15. Lai, V.D., Nguyen, T.H., Dernoncourt, F.: Extensively Matching for Few-shot Learning Event Detection. In: Workshop NUSE. pp. 38–45 (2020)
16. Li, Q., Ji, H., Huang, L.: Joint event extraction via structured prediction with global features. In: ACL. pp. 73–82. Sofia, Bulgaria (Aug 2013)
17. Liu, S., Cheng, R., Yu, X., Cheng, X.: Exploiting Contextual Information via Dynamic Memory Network for Event Detection. In: EMNLP. pp. 1030–1035 (2018)
18. Liu, X., Luo, Z., Huang, H.: Jointly Multiple Events Extraction via Attention-based Graph Information Aggregation. In: EMNLP. pp. 1247–1256 (2018)
19. Nguyen, T.H., Cho, K., Grishman, R.: Joint Event Extraction via Recurrent Neural Networks. In: NAACL-HLT. pp. 300–309. San Diego, California (2016)
20. Nguyen, T.H., Grishman, R.: Event Detection and Domain Adaptation with Convolutional Neural Networks. In: ACL-IJCNLP. pp. 365–371. Beijing, China (2015)
21. Nguyen, T.H., Grishman, R.: Graph Convolutional Networks with Argument-Aware Pooling for Event Detection. In: AAAI (2018)
22. Nichol, A., Achiam, J., Schulman, J.: On First-Order Meta-Learning Algorithms. arXiv:1803.02999 (2018)
23. Ramshaw, L., Marcus, M.: Text chunking using transformation-based learning. In: Workshop on Very Large Corpora (1995)
24. Ravi, S., Larochelle, H.: Optimization as a model for few-shot learning. In: ICLR (2017)
25. Santoro, A., Bartunov, S., Botvinick, M., Wierstra, D., Lillicrap, T.: Meta-learning with memory-augmented neural networks. In: ICML. pp. 1842–1850 (2016)
26. Shen, S., Wu, T., Qi, G., Li, Y.F., Haffari, G., Bi, S.: Adaptive Knowledge-Enhanced Bayesian Meta-Learning for Few-shot Event Detection. In: Findings of ACL-IJCNLP. pp. 2417–2429. Online (2021)
27. Snell, J., Swersky, K., Zemel, R.: Prototypical networks for few-shot learning. In: Advances in Neural Information Processing Systems. vol. 30 (2017)
28. Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P.H.S., Hospedales, T.M.: Learning to compare: Relation network for few-shot learning. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition pp. 1199–1208 (2018)
29. Vinyals, O., Blundell, C., Lillicrap, T., kavukcuoglu, k., Wierstra, D.: Matching networks for one shot learning. In: NeurIPS. vol. 29 (2016)
30. Walker, C., Strassel, S., Medero, J., Maeda, K.: ACE 2005 Multilingual Training Corpus (2006)
31. Yan, H., Jin, X., Meng, X., Guo, J., Cheng, X.: Event Detection with Multi-Order Graph Convolution and Aggregated Attention. In: EMNLP-IJCNLP. pp. 5766–5770 (2019)
32. Yan, W., Yap, J., Mori, G.: Multi-task transfer methods to improve one-shot learning for multimedia event detection. In: BMVC. pp. 37.1–37.13 (2015)
33. Zhang, H., Wang, H., Roth, D.: Zero-shot Label-Aware Event Trigger and Argument Classification. In: Findings of ACL-IJCNLP. pp. 1331–1340. Online (2021)