Event Role Extraction using Domain-Relevant Word Representations

Emanuela Boroş[†‡] **Romaric Besançon**[†] **Olivier Ferret**[†] **Brigitte Grau**[‡*]
[†]CEA, LIST, Vision and Content Engineering Laboratory, F-91191, Gif-sur-Yvette, France
[‡]LIMSI, rue John von Neumann, Campus Universitaire d'Orsay, F-91405 Orsay cedex
[*]ENSIIE, 1 square de la résistance F-91025 Évry cedex
firstname.lastname@cea.fr firstname.lastname@limsi.fr

Abstract

The efficiency of Information Extraction systems is known to be heavily influenced by domain-specific knowledge but the cost of developing such systems is considerably high. In this article, we consider the problem of event extraction and show that learning word representations from unlabeled domain-specific data and using them for representing event roles enable to outperform previous state-of-the-art event extraction models on the MUC-4 data set.

1 Introduction

In the Information Extraction (IE) field, event extraction constitutes a challenging task. An event is described by a set of participants (*i.e.* attributes or roles) whose values are text excerpts. The event extraction task is related to several subtasks: event mention detection, candidate role-filler extraction, relation extraction and event template filling. The problem we address here is the detection of role-filler candidates and their association with specific roles in event templates. For this task, IE systems adopt various ways of extracting patterns or generating rules based on the surrounding context, local context and global context (Patwardhan and Riloff, 2009). Current approaches for learning such patterns include bootstrapping techniques (Huang and Riloff, 2012a; Yangarber et al., 2000), weakly supervised learning algorithms (Huang and Riloff, 2011; Sudo et al., 2003; Surdeanu et al., 2006), fully supervised learning approaches (Chieu et al., 2003; Freitag, 1998; Bunescu and Mooney, 2004; Patwardhan and Riloff, 2009) and other variations. All these methods rely on substantial amounts of manually annotated corpora and use a large body of linguistic knowledge. The performance of these approaches is related to the amount of knowledge engineering deployed and a good choice of features and classifiers. Furthermore, the efficiency of the system relies on the a priori knowledge of the applicative domain (the nature of the events) and it is generally difficult to apply a system on a different domain with less annotated data without reconsidering the design of the features used. An important step forwards is TIER$_{light}$ (Huang and Riloff, 2012a) that targeted the minimization of human supervision with a bootstrapping technique for event roles detection. Also, PIPER (Patwardhan and Riloff, 2007; Patwardhan, 2010) distinguishes between relevant and irrelevant regions and learns domain-relevant extraction patterns using a semantic affinity measure. Another possible approach for dealing with this problem is to combine the use a restricted set of manually annotated data with a much larger set of data extracted in an unsupervised way from a corpus. This approach was experimented for relations in the context of Open Information Extraction (Soderland et al., 2010) but not for extracting events and their participants to our knowledge.

In this paper, we propose to approach the task of labeling text spans with event roles by automatically learning relevant features that requires limited prior knowledge, using a neural model to induce semantic word representations (commonly referred as *word embeddings*) in an unsupervised fashion, as in (Bengio et al., 2006; Collobert and Weston, 2008). We exploit these word embeddings as features for a supervised event role (multiclass) classifier. This type of approach has been proved efficient for numerous tasks in natural language processing, including named entity recognition (Turian et al., 2010), semantic role labeling (Collobert et al., 2011), machine translation (Schwenk and Koehn, 2008; Lambert et al., 2012), word sense disambiguation (Bordes et al., 2012) or sentiment analysis (Glorot et al., 2011; Socher et al., 2011) but has never been used, to our knowl-

Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1852–1857,
October 25-29, 2014, Doha, Qatar. ©2014 Association for Computational Linguistics

edge, for an event extraction task. Our goal is twofold: (1) to prove that using as only features word vector representations makes the approach competitive in the event extraction task; (2) to show that these word representations are scalable and robust when varying the size of the training data. Focusing on the data provided in MUC-4 (Lehnert et al., 1992), we prove the relevance of our approach by outperforming state-of-the-art methods, in the same evaluation environment as in previous works.

2 Approach

In this work, we approach the event extraction task by learning word representations from a domain-specific data set and by using these representations to identify the event roles. This idea relies on the assumption that the different words used for a given event role in the text share some semantic properties, related to their context of use and that these similarities can be captured by specific representations that can be automatically induced from the text, in an unsupervised way. We then propose to rely only on these word representations to detect the event roles whereas, in most works (Riloff, 1996; Patwardhan and Riloff, 2007; Huang and Riloff, 2012a; Huang and Riloff, 2012b), the role fillers are represented by a set of different features (raw words, their parts-of-speech, syntactic or semantic roles in the sentence).

Furthermore, we propose two additional contributions to the construction of the word representations. The first one is to exploit limited knowledge about the event types (seed words) to improve the learning procedure by better selecting the dictionary. The second one is to use a *max* operation[1] on the word vector representations in order to build noun phrase representations (since slot fillers are generally noun phrases), which represents a better way of aggregating the semantic information born by the word representations.

2.1 Inducing Domain-Relevant Word Representations

In order to induce the domain-specific word representations, we project the words into a 50-dimensional word space. We chose a single layer neural network (NN) architecture that avoids strongly engineered features, assumes little prior knowledge about the task, but is powerful enough to capture relevant domain information. Following (Collobert et al., 2011), we use an NN which learns to predict whether a given text sequence (short word window) exists naturally in the considered domain. We represent an input sequence of n words as $\langle w_i \rangle = \langle w_{i-(n/2)} \ldots, w_i, \ldots w_{i+(n/2)} \rangle$. The main idea is that each sequence of words in the training set should receive a higher score than a sequence in which one word is replaced with a random one. We call the sequence with a random word *corrupted* ($\langle \bar{w_i} \rangle$) and denote as *correct* ($\langle w_i \rangle$) all the sequences of words from the data set. The goal of the training step is then to minimize the following loss function for a word w_i in the dictionary D: $C_{w_i} = \sum_{w_i \in D} max(0, 1 - g(\langle w_i \rangle) + g(\langle \bar{w_i} \rangle))$, where $g(\cdot)$ is the scoring function given by the neural network. Further details and evaluations of these embeddings can be found in (Bengio et al., 2003; Bengio et al., 2006; Collobert and Weston, 2008; Turian et al., 2010). For efficiency, words are fed to our architecture as indices taken from a finite dictionary. Obviously, a simple index does not carry much useful information about the word. So, the first layer of our network maps each of these word indices into a feature vector, by a lookup table operation. Our first contribution intervenes in the process of the choosing the proper dictionary. (Bengio, 2009) has shown that the order of the words in the dictionary of the neural network is not indifferent to the quality of the achieved representations: he proposed to order the dictionary by frequency and select the words for the corrupted sequence according to this order. In our case, the most frequent words are not always the most relevant for the task of event role detection. Since we want to have a training more focused to the domain specific task, we chose to order the dictionary by word relevance to the domain. We accomplish this by considering a limited number of seed words for each event type that needs to be discovered in text (e.g. *attack, bombing, kidnapping, arson*). We then rate with higher values the words that are more similar to the event types words, according to a given semantic similarity, and we rank them accordingly. We use the "Leacock Chodorow" similarity from Wordnet 3.0 (Leacock and Chodorow, 1998). Initial experimental results proved that using this domain-

[1] This max operation consists in taking, for each component of the vector, the max value of this component for each word vector representation.

oriented order leads to better performance for the task than the order by frequency.

2.2 Using Word Representations to Identify Event Roles

After having generated for each word their vector representation, we use them as features for the annotated data to classify event roles. However, event role fillers are not generally single words but noun phrases that can be, in some cases, identified as named entities. For identifying the event roles, we therefore apply a two-step strategy. First, we extract the noun chunks using SENNA[2] parser (Collobert et al., 2011; Collobert, 2011) and we build a representation for these chunks defined as the maximum, per column, of the vector representations of the words it contains. Second, we use a statistical classifier to recognize the slot fillers, using this representation as features. We chose the extra-trees ensemble classifier (Geurts et al., 2006), which is a meta estimator that fits a number of randomized decision trees (*extra-trees*) on various sub-samples of the data set and use averaging to improve the predictive accuracy and control over-fitting.

3 Experiments and Results

3.1 Task Description

We conducted the experiments on the official MUC-4 training corpus that consists of 1,700 documents and instantiated templates for each document. The task consists in extracting information about terrorist events in Latin America from news articles. We classically considered the following 4 types of events: *attack*, *bombing*, *kidnapping* and *arson*. These are represented by templates containing various slots for each piece of information that should be extracted from the document (perpetrators, human targets, physical targets, etc). Following previous works (Huang and Riloff, 2011; Huang and Riloff, 2012a), we only consider the "String Slots" in this work (other slots need different treatments) and we group certain slots to finally consider the five slot types *PerpInd* (individual perpetrator), *PerpOrg* (organizational perpetrator), *Target* (physical target), *Victim* (human target name or description) and *Weapon* (instrument id or type). We used 1,300 documents (DEV) for training, 200 documents (TST1+TST2) for tuning, and 200 documents (TST3+TST4) as the blind test set. To compare with similar works, we do not evaluate the template construction and only focus on the identification of the slot fillers: for each answer key in a reference template, we check if we find it correctly with our extraction method, using head noun matching (e.g., the victim *her mother Martha Lopez Orozco de Lopez* is considered to match *Matha Lopez*), and merging duplicate extractions (so that different extracted slot fillers sharing the same head noun are counted only once). We also took into account the answer keys with multiple values in the reference, dealing with conjunctions (when several victims are named, we need to find all of them) and disjunctions (when several names for the same organization are possible, we need to find any of them). Our results are reported as Precision/Recall/F1-score for each event role separately and averaged on all roles.

3.2 Experiments

In all the experiments involving our model, we established the following stable choices of parameters: 50-dimensional vectors obtained by training on sequences of 5 words, which is consistent with previous studies (Turian et al., 2010; Collobert and Weston, 2008). All the hyper-parameters of our model (e.g. learning rate, size of the hidden layer, size of the word vectors) have been chosen by finetuning our event extraction system on the TST1+TST2 data set. For *DRVR-50* and *W2V-50*, the embeddings were built from the whole training corpus (1,300 documents) and the dictionary was made of all the words of this corpus under their inflected form.

We used the extra-trees ensemble classifier implemented in (Pedregosa et al., 2011), with hyper-parameters optimized on the validation data: forest of 500 trees and the maximum number of features to consider when looking for the best split is $\sqrt{number\_features}$. We present a 3-fold evaluation: first, we compare our system with state-of-the-art systems on the same task, then we compare our domain-relevant vector representations (*DRVR-50*) to more generic word embeddings (*C&W50*, *HLBL-50*)[3] and finally to another

[2] Code and resources can be found at `http://ml.nec-labs.com/senna/`

[3] *C&W-50* are described in (Collobert and Weston, 2008), *HLBL-50* are the Hierarchical log-bilinear embeddings (Mnih and Hinton, 2007), provided by (Turian et al., 2010), available at `http://metaoptimize.com/projects/wordreprs` induced from the Reuters-RCV1

State-of-the-art systems						
	PerpInd	**PerpOrg**	**Target**	**Victim**	**Weapon**	**Average**
(Riloff, 1996)	33/49/40	53/33/41	54/59/56	49/54/51	38/44/41	45/48/46
(Patwardhan and Riloff, 2007)	39/48/43	55/31/40	37/60/46	44/46/45	47/47/47	44/36/40
(Patwardhan and Riloff, 2009)	51/58/54	34/45/38	43/72/53	55/58/56	57/53/55	48/57/52
(Huang and Riloff, 2011)	48/57/52	46/53/50	51/73/60	56/60/58	53/64/58	51/62/56
(Huang and Riloff, 2012a)	47/51/47	60/39/47	37/65/47	39/53/45	53/55/54	47/53/50
(Huang and Riloff, 2012b)	54/57/56	55/49/51	55/68/61	63/59/61	62/64/63	58/60/59
Models based on word embeddings						
C&W-50	80/55/65	64/65/64	76/72/74	53/63/57	85/64/73	68/63/65
HLBL-50	81/53/64	63/67/65	78/72/75	53/63/58	93/64/75	69/62/66
W2V-50	79/57/66	88/71/79	74/72/73	69/75/71	97/65/78	77/68/72
DRVR-50	79/57/66	91/74/81	79/57/66	77/75/76	92/58/81	80/67/73

Table 1: Accuracy of "String Slots" on the TST3 + TST4 test set P/R/F1 (Precision/Recall/F1-Score)

word representation construction on the domain-specific data (*W2V-50*)[4].

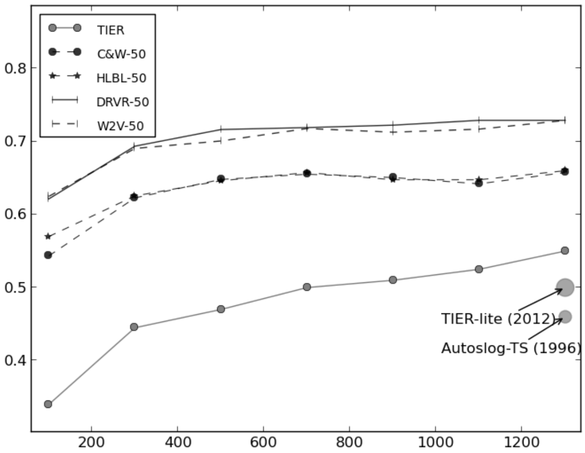

Figure 1: F1-score results for event role labeling on MUC-4 data, for different size of training data, of "String Slots" on the TST3+TST4 with different parameters, compared to the learning curve of TIER (Huang and Riloff, 2012a). The grey points represent the performances of other IE systems.

Figure 1 presents the average F1-score results, computed over the slots PerpInd, PerpOrg, Target, Victim and Weapon. We observe that models relying on word embeddings globally outperform the state-of-the-art results, which demonstrates that the word embeddings capture enough semantic information to perform the task of event role labeling on "String Slots" without using any additional hand-engineered features. Moreover, our representations (*DRVR-50*) clearly surpass the models based on generic embeddings (C&W-50 and HLBL-50) and obtain better results than W2V-50, based the competitive model of (Mikolov et al., 2013a), even if the difference is small. We can also note that the performance of our model is good even with a small amount of training data, which makes it a good candidate to easily develop an event extraction system on a new domain.

Table 1 provides a more detailed analysis of the comparative results. We can see in this table that our results surpass those of previous systems (0.73 vs. 0.59) with, particularly, a consistently higher precision on all roles, whereas recall is smaller for certain roles (Target and Weapon). To further explore the impact of these representations, we compared our word embeddings with other word embeddings (*C&W-50*, *HLBL-50*) and report the results in Figure 1 and Table 1. The results show that our model also outperforms the models using others word embeddings (F1-score of 0.73 against 0.65, 0.66). This proves that a model learned on a domain-specific data set does indeed provide better results, even if its size is much smaller (whereas it is usually considered that neural models require often important training data). Finally, we also achieve slightly better results than *W2V-50* with other word representations built on the same corpus, which shows that the choices made for the word representation construction, such as the use of domain information for word ordering, tend to have a positive impact.

newswire corpus

[4] *W2V-50* are the embeddings induced from the MUC4 data set using the negative sampling training algorithm (Mikolov et al., 2013a; Mikolov et al., 2013b; Mikolov et al., 2013c), available at `https://code.google.com/p/word2vec/`

4 Conclusions and Perspectives

We presented in this paper a new approach for event extraction by reducing the features to only use unsupervised word representations and a small set of seed words. The word embeddings induced from a domain-specific corpus bring improvement over state-of-art models on the standard MUC-4 corpus and demonstrate a good scalability on different sizes of training data sets. Therefore, our proposal offers a promising path towards easier and faster domain adaptation. We also prove that using a domain-specific corpus leads to better word vector representations for this task than using other publicly-available word embeddings (even if they are induced from a larger corpus).

As future work, we will reconsider the architecture of the neural network and we will refocus on creating a deep learning model while taking advantage of a larger set of types of information such as syntactic information, following (Levy and Goldberg, 2014), or semantic information, following (Yu and Dredze, 2014).

References

Yoshua Bengio, Rejean Ducharme, and Pascal Vincent. 2003. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155.

Yoshua Bengio, Holger Schwenk, Jean-Sébastian Senécal, Fréderic Morin, and Jean-Luc Gauvain. 2006. Neural probabilistic language models. In DawnE. Holmes and LakhmiC. Jain, editors, *Innovations in Machine Learning*, volume 194 of *Studies in Fuzziness and Soft Computing*, pages 138–186. Springer Berlin Heidelberg.

Yoshua Bengio. 2009. Learning deep architectures for AI. *Foundations and trends in Machine Learning*, 2(1).

Antoine Bordes, Xavier Glorot, Jason Weston, and Yoshua Bengio. 2012. Joint learning of words and meaning representations for open-text semantic parsing. In *Fifteenth International Conference on Artificial Intelligence and Statistics (AISTATS 2012)*, pages 127–135.

Razvan Bunescu and Raymond J Mooney. 2004. Collective information extraction with relational markov networks. In *42nd Annual Meeting on Association for Computational Linguistics (ACL-04)*, pages 438–445.

Hai Leong Chieu, Hwee Tou Ng, and Yoong Keok Lee. 2003. Closing the gap: Learning-based information extraction rivaling knowledge-engineering methods. In *41st international Annual Meeting on Association for Computational Linguistics (ACL-2003)*, pages 216–223.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *25th International Conference of Machine learning (ICML-08)*, pages 160–167. ACM.

Ronan Collobert, Jason Weston, Léon Battou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537.

Ronan Collobert. 2011. Deep learning for efficient discriminative parsing. In *14th International Conference on Artificial Intelligence and Statistics (AISTATS 2011)*.

Dayne Freitag. 1998. Information extraction from HTML: Application of a general machine learning approach. In *AAAI'98*, pages 517–523.

Pierre Geurts, Damien Ernst, and Louis Wehenkel. 2006. Extremely randomized trees. *Machine Learning*, 63(1):3–42.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *28th International Conference on Machine Learning (ICML-11)*, pages 513–520.

Ruihong Huang and Ellen Riloff. 2011. Peeling back the layers: Detecting event role fillers in secondary contexts. In *ACL 2011*, pages 1137–1147.

Ruihong Huang and Ellen Riloff. 2012a. Bootstrapped training of event extraction classifiers. In *13th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2012)*, pages 286–295.

Ruihong Huang and Ellen Riloff. 2012b. Modeling textual cohesion for event extraction. In *26th Conference on Artificial Intelligence (AAAI 2012)*.

Patrik Lambert, Holger Schwenk, and Frédéric Blain. 2012. Automatic translation of scientific documents in the hal archive. In *LREC 2012*, pages 3933–3936.

Claudia Leacock and Martin Chodorow. 1998. Combining local context and Wordnet similarity for word sense identification. In Christiane Fellbaum, editor, *WordNet: An electronic lexical database.*, pages 265–283. MIT Press.

Wendy Lehnert, Claire Cardie, David Fisher, John McCarthy, Ellen Riloff, and Stephen Soderland. 1992. University of Massachusetts: MUC-4 test results and analysis. In *4th Conference on Message understanding*, pages 151–158.

Omer Levy and Yoav Goldberg. 2014. Dependency-based word embeddings. In *52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014), Short Papers*, pages 302–308, Baltimore, Maryland, June.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *International Conference on Learning Representations (ICLR 20013), workshop track.*

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26 (NIPS 2013)*, pages 3111–3119.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013c. Linguistic regularities in continuous space word representations. In *NAACL-HLT 2013*, pages 746–751.

Andriy Mnih and Geoffrey Hinton. 2007. Three new graphical models for statistical modelling. In *24th International Conference of Machine learning (ICML 2007)*, pages 641–648. ACM.

Siddharth Patwardhan and Ellen Riloff. 2007. Effective information extraction with semantic affinity patterns and relevant regions. In *2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2007)*, pages 717–727.

Siddharth Patwardhan and Ellen Riloff. 2009. A unified model of phrasal and sentential evidence for information extraction. In *2009 Conference on Empirical Methods in Natural Language Processing (EMNLP 2009)*, pages 151–160.

Siddharth Patwardhan. 2010. *Widening the field of view of information extraction through sentential event recognition*. Ph.D. thesis, University of Utah.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Ellen Riloff. 1996. Automatically generating extraction patterns from untagged text. In *AAAI'96*, pages 1044–1049.

Holger Schwenk and Philipp Koehn. 2008. Large and diverse language models for statistical machine translation. In *IJCNLP 2008*, pages 661–666.

Richard Socher, Cliff C Lin, Chris Manning, and Andrew Y Ng. 2011. Parsing natural scenes and natural language with recursive neural networks. In *28th International Conference on Machine Learning (ICML-11)*, pages 129–136.

Stephen Soderland, Brendan Roof, Bo Qin, Shi Xu, Mausam, and Oren Etzioni. 2010. Adapting open information extraction to domain-specific relations. *AI Magazine*, 31(3):93–102.

Kiyoshi Sudo, Satoshi Sekine, and Ralph Grishman. 2003. An improved extraction pattern representation model for automatic ie pattern acquisition. In *41st Annual Meeting on Association for Computational Linguistics (ACL-03)*, pages 224–231.

Mihai Surdeanu, Jordi Turmo, and Alicia Ageno. 2006. A hybrid approach for the acquisition of information extraction patterns. In *EACL-2006 Workshop on Adaptive Text Extraction and Mining (ATEM 2006)*, pages 48–55.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *48th international Annual Meeting on Association for Computational Linguistics (ACL 2010)*, pages 384–394.

Roman Yangarber, Ralph Grishman, Pasi Tapanainen, and Silja Huttunen. 2000. Automatic acquisition of domain knowledge for information extraction. In *18th Internation Conference on Computational Linguistics (COLING 2000)*, pages 940–946.

Mo Yu and Mark Dredze. 2014. Improving lexical embeddings with semantic knowledge. In *52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014), Short Papers*, pages 545–550, Baltimore, Maryland, June.