SPARSE:
Structured Prediction Using Argument-Relative Structured Encoding

Rishi Bommasani
Dept. of Computer Science
Cornell University
rb724@cornell.edu

Arzoo Katiyar
Dept. of Computer Science
Cornell University
arzoo@cs.cornell.edu

Claire Cardie
Dept. of Computer Science
Cornell University
cardie@cs.cornell.edu

Abstract

We propose *structured encoding* as a novel approach to learning representations for relations and events in neural structured prediction. Our approach explicitly leverages the structure of available relation and event metadata to generate these representations, which are parameterized by both the attribute structure of the metadata as well as the learned representation of the arguments of the relations and events. We consider affine, biaffine, and recurrent operators for building hierarchical representations and modelling underlying features.

Without task-specific knowledge sources or domain engineering, we significantly improve over systems and baselines that neglect the available metadata or its hierarchical structure. We observe across-the-board improvements on the BeSt 2016/2017 sentiment analysis task of at least **2.3** (absolute) and **10.6%** (relative) F-measure over the previous state-of-the-art.

1 Introduction

Information extraction has long been an active subarea of natural language processing (NLP) (Onyshkevych et al., 1993; Freitag, 2000). A particularly important class of extraction tasks is ERE detection in which an object, typically an element in a knowledge base, is created for each ENTITY, RELATION, and EVENT identified in a given text (Song et al., 2015). In some variants of ERE detection, *metadata* descriptions and specific *mentions* of the ERE objects also need to be recorded as shown graphically in Figure 1. Subsequent second-order extraction tasks can further build upon first-order ERE information.

In this work, in particular, we consider the second-order structured prediction task studied in the 2016/2017 Belief and Sentiment analysis evaluations (BeSt) (Rambow et al., 2016a): given a document and its EREs (including metadata and mentions) determine the *sentiment* of each ENTITY towards every RELATION and EVENT in the document.[1]

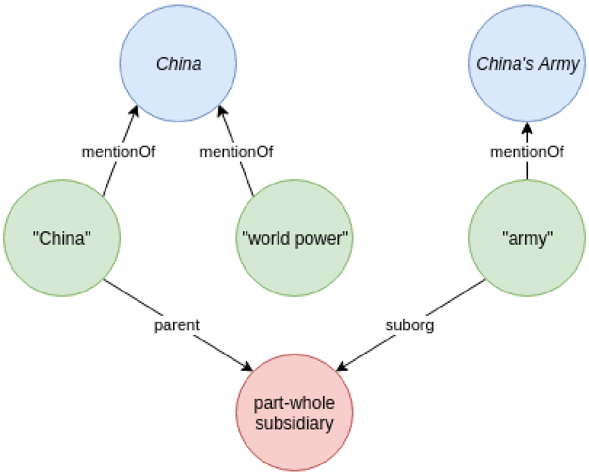

Figure 1: ERE graph for “China, a growing world power, is developing its army.” Entities are denoted in blue, entity mentions in green, and relations in red.

Until quite recently, existing approaches to this type of second-order extraction task have relied heavily on domain knowledge and feature engineering (Rambow et al., 2016b; Niculae et al., 2016; Dalton et al., 2016; Gutirrez et al., 2016). And while end-to-end neural network methods that bypass feature engineering have been developed for information extraction problems, they have largely been applied to first-order extraction tasks (Katiyar and Cardie, 2016; Miwa and Bansal, 2016; Katiyar and Cardie, 2017; Orr et al., 2018). Possibly more importantly, these techniques ignore, or have no access to, the internal structure of relations and events.

We hypothesize that utilizing the metadata-

[1] The BeSt evaluation also requires the identification of sentiment towards entities. For reasons that will become clear later, we do not consider entity-to-entity sentiment here.

Proceedings of 3rd Workshop on Structured Prediction for NLP, pages 13–17
Minneapolis, Minnesota, June 7, 2019. ©2019 Association for Computational Linguistics

induced structure of relations and events will improve performance on the second-order sentiment analysis task. To this end, we propose *structured encoding* as an approach towards learning representations for relations and events in end-to-end neural structured prediction. In particular, our approach not only models available metadata but also its hierarchical nature.

We evaluate the structured encoding approach on the BeSt 2016/2017 dataset. Without sentiment-specific resources, we are able to see significant improvements over baselines that do not take into account the available structure. We achieve state-of-the-art F1 scores of **22.9** for discussion forum documents and **14.0** for newswire documents. We also openly release our implementation to promote further experimentation and reproducible research.

2 Task Formulation

In the BeSt sentiment analysis setting, we are given a corpus $\mathcal{D}$ where every document $d_i \in \mathcal{D}$ is annotated with the set of entities $\mathcal{E}_i$, relation mentions $\mathcal{R}_i$, and event mentions[2] $\mathcal{EV}_i$ present in d_i. For each entity $e_j \in \mathcal{E}_i$, we are additionally given the *span*, or variable-length n-gram, associated with (each of) its mention(s). Similarly, for each relation and event, we are given metadata specifications of its type and subtype as well as the arguments that constitute it. Our task is then the following: for each potential source-target pair, $(src, trg) \in (\mathcal{E}_i \times \mathcal{R}_i) \cup (\mathcal{E}_i \times \mathcal{EV}_i)$, predict the sentiment (one of POSITIVE, NEGATIVE or NONE) of the src entity towards the trg relation/event.

3 Model

Our structured encoding model is depicted in Figure 2. Its goal is to compute representations for the $\mathbf{src}$, $\mathbf{trg}$, and context $\mathbf{c}$; we then pass the concatenated triple into a FFNN for sentiment classification. We describe the model components in the paragraphs below.

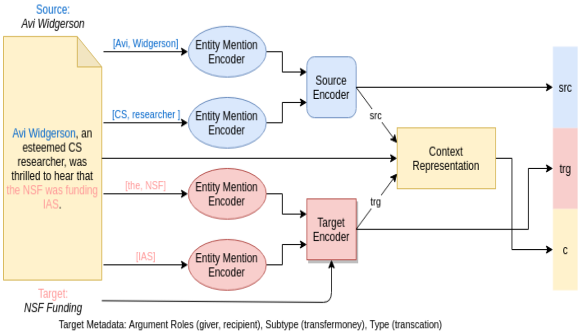

Figure 2: Model for representing the source (blue), target (red), and context (yellow) for the directed sentiment analysis task.

Word and Entity Mention Representation Given a span $s = w_1, \ldots, w_{|s|}$ of the input document, a noncontextual embedding mapping f, and a contextual mapping g, we create the initial word representation $\mathbf{w}_t$ for each word w_t in s by concatenating its noncontextual and contextual embedding. We then pass the initial word representations through a bidirectional RNN to obtain a domain-specific contextual word representation $\mathbf{h}_t$.[3]

$$\begin{aligned} \mathbf{w}_t &= [f(w_t); g(w_t|s)] \\ \mathbf{h}_t &= [\overrightarrow{RNN}(\mathbf{w}_t); \overleftarrow{RNN}(\mathbf{w}_t)] \end{aligned} \tag{1}$$

For each entity mention s, we compute a representation of s as the mean-pooling of $\mathbf{h}_1, \ldots, \mathbf{h}_{|s|}$.

Source Encoding For each source entity e_j we are given its grounded entity mentions $e_j^1, \ldots, e_j^n$. Similar to the representation methodology for entity-mentions, we represent each source as the average of its entity mention representations.

$$\mathbf{e_j} = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{|[e_j^i]|} \sum_{t=1}^{|[e_j^i]|} \mathbf{h}_t \tag{2}$$

The dataset is also annotated with entities corresponding to the authors of articles. To more appropriately handle these entities, we learn two author embeddings. The *post author* embedding represents the source entity if the target occurs in a post written by the source. If this is not the case, we use the *other author* embedding.

Target Encoding To encode targets, we leverage the hierarchical structure provided in the relation and event metadata. Initially, we construct a fixed-length representation for each of the target's *arguments*, which are *(role, entity mention)* pairs, (r, e_j^k), via concatenation (*flat*) or an affine map (*affine*) as follows (where U_r and $\mathbf{v}_r$ are learned):

$$\begin{aligned} \mathbf{arg}^r_{j,k} &= U_r \mathbf{e}^k_j \quad \textit{(affine)} \\ \mathbf{arg}^r_{j,k} &= [\mathbf{v}_r; \mathbf{e}^k_j] \quad \textit{(flat)} \end{aligned} \tag{3}$$

For relations, given that the dataset enforces the constraint of two arguments per relation, we pool

[2] In this work, the distinction between relations/events and relation/event mentions is not considered, so we use 'relation' and 'event' as a shorthand.

[3] This process is not shown in Figure 2.

the relation vectors by concatenating them. On the other hand, for events, as the number of arguments is variable, we propose to pool them with a GRU encoder (Cho et al., 2014). This allows for us to deal with the variable number of arguments and learn compositional relationships between different arguments (arguments are ordered by their semantic roles in the overarching event).

Both relations and events are annotated with their types and subtypes. Based on our hypothesis that hierarchical modeling of structure is important in encoding targets, we consider encoding the subtype and type using *flat* concatenation or learned *affine* maps applied successively in a similar fashion to role encoding for arguments.

Context Representation We encode the source and target independently, capturing the inductive bias that these two components of the input have stand-alone meanings. We find that this is computationally efficient as opposed to approaches that model the source and target on a pair-specific basis. We introduce source-target interaction into the architecture by modeling the textual context in which they appear: first select a source-target specific *context* span; then construct a fixed-length encoding of the *context* using an attention mechanism conditional on the source-target pair. To identify the context span, we identify the closest mention of the source that precedes the target. We begin the context span starting at the first word beyond the end of this mention (w_i). Similarly, we conclude the span at the last word preceding the target (w_j). Denoting the context span as $[w_i, w_j]$, we compute the context vector $\mathbf{c}$ as (where the U maps are learned):

$$\begin{aligned}
&\mathbf{u}_{src} = U_{src}\,\mathbf{src};\ \mathbf{u}_{trg} = U_{trg}\,\mathbf{trg};\ \mathbf{u}_t = U_c\,\mathbf{h}_t \\
&\alpha_t = \mathbf{u}_t^T(\mathbf{u}_{src} \odot \mathbf{u}_{trg}) \\
&\alpha = \mathit{softmax}([\alpha_i, \ldots, \alpha_j]^T) \\
&\mathbf{c} = [\mathbf{h}_i, \ldots, \mathbf{h}_j]\alpha
\end{aligned} \tag{4}$$

We truncate long spans (length greater than 20) to be the 20 words preceding the start of the target ($i = j - 20$).

4 Results and Analysis

Dataset and Evaluation We use the LDCE114 dataset that was used in the BeSt 2016/2017 English competition. The dataset contains 165 documents, 84 of which are multi-post discussion forums (DF) and 81 are single-post newswire articles (NW), and is summarized in Table 1. We observe that the data is extremely sparse with respect to positive examples: only **0.203%** of all source-target pairs are positive examples with 431 positive examples in 211310 candidate pairs in the training set.

	DF	NW
Length (in num. tokens)	750.36	538.96
Relation Pairs	775.34	1474.51
Relation Positive Examples	1.32	1.02
Event Pairs	810.36	1334.53
Event Positive Examples	2.60	3.98

Table 1: Average document statistics

Consistent with the BeSt evaluation framework, we report the microaverage *adjusted* F1 score[4] and treat NONE as the negative class and both POSITIVE and NEGATIVE as the positive classes. The BeSt metric introduces a notion of partial credit to reward predictions that are incorrect but share global properties with the correct predictions. We find that this metric is well-suited for structured prediction treatments of the task that also consider global structure.

Implementation Details We use frozen word representations that concatenate noncontextual 300-dimensional GloVe embeddings (Pennington et al., 2014) and contextual 3072-dimensional ELMo embeddings (Peters et al., 2018). We use a 2-layer bidirectional LSTM encoder (Hochreiter and Schmidhuber, 1997) and consistently use both a hidden dimensionality of 128 and *tanh* nonlinearities throughout the work. Based on results on the development set, we use an attention dimensionality of 32, a dropout (Srivastava et al., 2014) probability in all hidden layers of 0.2, and a batch size of 10 documents. We also find that a single-layer GRU worked best for event argument pooling and interestingly find that a unidirectional GRU is preferable to a bidirectional GRU. The final classifier is a single-layer FFNN. The model is trained with respect to the class-weighted cross entropy loss where weights are the ℓ_1-normalized vector corresponding to inverse class frequences and optimized using ADAM (Kingma and Ba, 2014) with the default parameters in PyTorch (Paszke et al., 2017). All models are trained for 50 epochs.

[4] Complete results can be found in Appendix A.

			DF	NW
Baseline			14.53	7.24
Columbia/GWU			20.69	10.10
Cornell-Pitt-Michigan			19.48	0.70
Target	Arg	(Sub)Type		
Relation	Flat	Flat	15.2	8.5
Event	Flat	Flat	15.5	8.5
Both	Affine	Affine	**22.9**	**14.0**

Table 2: Experimental results on the BeSt dataset. The specified target types are encoded using *structured encoding* with the specified argument and subtype/type encoding methods. The other target type is encoded with affine maps for the arguments, subtypes, and types.

Overall Results In Table 2, the upper half of the table describes the official BeSt baseline system as well as the top systems on the task. The baseline employs a rule-based heuristic that considers pairs for which the source is the article author as potential positive examples and classifies positive examples as NEGATIVE (the more frequent positive class) (Rambow et al., 2016a). The Columbia/GWU system treats second-order sentiment classification as first-order relation extraction and extends a relation extraction system that uses phrase structure trees, dependency trees, and semantic parses with an SVM using tree kernels (Rambow et al., 2016b). The Cornell-Pitt-Michigan system employs a rule-based heuristic to prune candidate pairs in the sense of link prediction and then performs sentiment classification via multinomial logistic regression (Niculae et al., 2016). They find that the link prediction system sometimes predicts spurious links and permit the classifier to overrule the link prediction judgment by predicting NONE.

Note these results include entities as targets and we argue that this makes the task comparatively easier. In particular, sparsity and dataset size are less of a concern for entity targets as positive examples are equally frequent (**0.196%**) and there are significantly more entity pairs (438165 pairs) than relation and event targets combined (211310 pairs).[5]

In the lower half of Table 2, we report results for different *structured encoding* approaches. As we simultaneously predict sentiment towards relation targets and event targets, we find that the encoding method for one target category tends to improve performance on the other due to shared representations (refer to Appendix A). We find that using affine encoders, which is consistent with the inductive bias of hierarchical encoding, performs best in all settings.

Embeddings	Encoder	DF	NW
GloVe + ELMo	Bidi-LSTM	**22.9**	**14.0**
GloVe + ELMo*	Bidi-LSTM	8.5	2.4
GloVe	Bidi-LSTM	12.6	8.3
ELMo	Bidi-LSTM	15.0	9.9
GloVe + ELMo	Uni-LSTM	20.9	13.6
GloVe + ELMo	none	22.1	13.6

Table 3: Experimental results on the effects of word representations. * indicates embeddings are not frozen.

Word Representations To understand the effect of pretraining given its pervasive success throughout NLP as well as the extent to which domain-specific LSTM encoders are beneficial, we perform experiments on the development set. When the LSTM encoder is omitted, we introduce an affine projection that yields output vectors of the same dimension as the original LSTM (128). This ensures that the capacity of subsequent model components is unchanged but means there is no domain-specific context modelling. As shown in Table 3, the combination of contextual and noncontextual embeddings leads to an improvement but the contextual embeddings perform better stand-alone. In doing this comparison, we consider pretrained embeddings of differing dimensionalities however since in most settings the embeddings are frozen, we do not find that this significantly effects the run time or model capacity. When embeddings are learned, we observe a substantial decline in performance which we posit is due to catastrophic forgetting due to noisy and erratic signal from the loss. The results further indicate that a bidirectional task oriented encoder improves performance.

5 Conclusion

In this paper, we propose *structured encoding* to model structured targets in semantic link prediction. We demonstrate that this framework along with pretrained word embeddings can be an effective combination as we achieve state-of-the-art performance on several metrics in the BeSt 2016/2017 Task in English.

[5] Additionally, entities are unstructured targets and therefore are not well-suited for the contributions of this work.

Acknowledgements

We thank Vlad Niculae for his tremendous support and guidance in the early stages of this work and RB especially thanks him for his advice on research and for being the gold standard for code quality. We thank the anonymous reviewers for their insightful feedback.

References

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078.

Adam Dalton, Morgan Wixted, Yorick Wilks, Meenakshi Alagesan, Gregorios Katsios, Ananya Subburathinam, and Tomek Strzalkowski. 2016. Target-focused sentiment and belief extraction and classification using cubism. In *TAC*.

Dayne Freitag. 2000. Machine learning for information extraction in informal domains. *Machine Learning*, 39(2):169–202.

Yoan Gutirrez, Salud Mara Jimnez-Zafra, Isabel Moreno, M. Teresa Martn-Valdivia, David Toms, Arturo Montejo-Rez, Andrs Montoyo, L. Alfonso Urea-Lpez, Patricio Martnez-Barco, Fernando Martnez-Santiago, Rafael Muoz, and Eladio Blanco-Lpez. 2016. Redes at tac knowledge base population 2016: Edl and best tracks. In *TAC*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Arzoo Katiyar and Claire Cardie. 2016. Investigating lstms for joint extraction of opinion entities and relations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 919–929. Association for Computational Linguistics.

Arzoo Katiyar and Claire Cardie. 2017. Going out on a limb: Joint extraction of entity mentions and relations without dependency trees. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 917–928. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using lstms on sequences and tree structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1116. Association for Computational Linguistics.

Vlad Niculae, Kai Sun, Xilun Chen, Yao Cheng, Xinya Du, Esin Durmus, Arzoo Katiyar, and Claire Cardie. 2016. Cornell belief and sentiment system at tac 2016. In *TAC*.

Boyan Onyshkevych, Mary Ellen Okurowski, and Lynn Carlson. 1993. Tasks, domains, and languages for information extraction. In *Proceedings of a workshop on held at Fredericksburg, Virginia: September 19-23, 1993*, pages 123–133. Association for Computational Linguistics.

Walker Orr, Prasad Tadepalli, and Xiaoli Fern. 2018. Event detection with neural networks: A rigorous empirical evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 999–1004. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

Owen Rambow, Meenakshi Alagesan, Michael Arrigo, Daniel Bauer, Claire Cardie, Adam Dalton, Mona Diab, Greg Dubbin, Gregorios Katsios, Axinia Radeva, Tomek Strzalkowski, and Jennifer Tracey. 2016a. The 2016 tac kbp best evaluation. In *TAC*.

Owen Rambow, Tao Yu, Axinia Radeva, Alexander Richard Fabbri, Christopher Hidey, Tianrui Peng, Kathleen McKeown, Smaranda Muresan, Sardar Hamidian, Mona T. Diab, and Debanjan Ghosh. 2016b. The columbia-gwu system at the 2016 tac kbp best evaluation. In *TAC*.

Zhiyi Song, Ann Bies, Stephanie Strassel, Tom Riese, Justin Mott, Joe Ellis, Jonathan Wright, Seth Kulick, Neville Ryant, and Xiaoyi Ma. 2015. From light to rich ere: annotation of entities, relations, and events. In *Proceedings of the the 3rd Workshop on EVENTS: Definition, Detection, Coreference, and Representation*, pages 89–98.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958.