Context-Aware Representations for Knowledge Base Relation Extraction

Daniil Sorokin and **Iryna Gurevych**
Ubiquitous Knowledge Processing Lab (UKP)
Research Training Group AIPHES
Department of Computer Science
Technische Universität Darmstadt
www.ukp.tu-darmstadt.de

Abstract

We demonstrate that for sentence-level relation extraction it is beneficial to consider other relations in the sentential context while predicting the target relation. Our architecture uses an LSTM-based encoder to jointly learn representations for all relations in a single sentence. We combine the context representations with an attention mechanism to make the final prediction.

We use the Wikidata knowledge base to construct a dataset of multiple relations per sentence and to evaluate our approach. Compared to a baseline system, our method results in an average error reduction of 24% on a held-out set of relations.

The code and the dataset to replicate the experiments are made available at https://github.com/ukplab.

1 Introduction

The main goal of relation extraction is to determine a type of relation between two target entities that appear together in a text. In this paper, we consider the sentential relation extraction task: to each occurrence of the target entity pair $\langle e_1, e_2 \rangle$ in some sentence s one has to assign a relation type r from a given set R (Hoffmann et al., 2011). A triple $\langle e_1, r, e_2 \rangle$ is called a *relation instance* and we refer to the relation of the target entity pair as *target relation*. Relation extraction is a fundamental task that enables a wide range of semantic applications from question answering (Xu et al., 2016) to fact checking (Vlachos and Riedel, 2014).

For relation extraction, it is crucial to be able to extract relevant features from the sentential context (Riedel et al., 2010; Zeng et al., 2015). Modern approaches focus just on the relation between the target entities and disregard other relations that might be present in the same sentence (Zeng et al., 2015; Lin et al., 2016). For example, in order to correctly identify the relation type between the movie e_1 and the director e_2 in (1), it is important to separate out the INSTANCE_OF relation between the movie and its type e_3:

(1) [e_1 **Star Wars VII**] is an American [e_3 **space opera epic film**] directed by [e_2 **J. J. Abrams**].

We present a novel architecture that considers other relations in the sentence as a context for predicting the label of the target relation. We use the term *context relations* to refer to them throughout the paper. Our architecture uses an LSTM-based encoder to jointly learn representations for all relations in a single sentence. The representation of the target relation and representations of the context relations are combined to make the final prediction.

To facilitate the experiments we construct a dataset that contains multiple positive and negative relation instances per sentence. We employ a fast growing community managed knowledge base (KB) Wikidata (Vrandečić and Krötzsch, 2014) to build the dataset.

Our main contribution is the new neural network architecture for extracting relations between an entity pair that takes into account other relations in the sentence.

2 Related Work

We employ a neural network to automatically encode the target relation and the sentential context into a fixed-size feature vector. Mintz et al. (2009) and Riedel et al. (2010) have used manually engineered features based on part-of-speech tags and dependency parses to represent the target relations. Recently, Zeng et al. (2015) and Zhao et al. (2015) have shown that one can successfully apply convo-

Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 1785–1790
Copenhagen, Denmark, September 7–11, 2017.

lutional neural networks to extract sentence-level features automatically.

Most of the methods (Riedel et al., 2010; Zeng et al., 2015; Lin et al., 2016) focus on predicting a single relation type based on the combined evidence from all of the occurrences of an entity pair. Hoffmann et al. (2011) and Surdeanu et al. (2012) assign multiple relation types to each entity pair, such that the predictions are tied to particular occurrences of the entity pair. We regard the relation extraction task similarly and predict relation types on the sentence level.

We use a distant supervision approach (Mintz et al., 2009) to construct the dataset. Mintz et al. (2009) and Riedel et al. (2010) have applied it to create relation extraction datasets for a large-scale KB. In contrast to our dataset, their data contains a single relation instance per sentence. That makes it incompatible with our method.

All of the aforementioned approaches consider just the relation between the target entities and disregard other relations that might be present in the same sentence. Our method uses context relations to predict the target relation. One can also use other types of structured information from the nearby context to improve relation extraction. Roth and Yih (2004) have combined named entity recognition and relation extraction in a structured prediction approach to improve both tasks. Later, Miwa and Bansal (2016) have implemented an end-to-end neural network to construct a context representation for joint entity and relation extraction. Finally, Li et al. (2013) have designed global features and constraints to extract multiple events and their arguments from the same sentence.

We don't implement global constraints in our approach, since unlike events and arguments, there are no restrictions as to what relations can appear together. Instead we encode all relations in the same context into fixed-size vectors and use an attention mechanism to combine them.

3 Data generation with Wikidata

Wikidata is a collaboratively constructed KB that encodes common world knowledge in a form of binary relation instances (e.g. CAPITAL:P36 (*Hawaii*:Q782, *Honolulu*:Q18094))[1]. It contains more than 28 million entities and 160 million relation instances.[2] A broad community oversight, similar to Wikipedia, ensures a higher data quality compared to other KBs (Färber et al., 2015).

	Train	Validation	Held-out
# of relation triples	284,295	113,852	287,902
# of relation inst.	578,199	190,160	600,804

Table 1: Statistics of the generated dataset.

We use the complete English Wikipedia corpus to generate training and evaluation data. Wikipedia and Wikidata are tightly integrated which enables us to employ manual wiki annotations to extract high quality data. From each sentence in a complete article we extract link annotations and retrieve Wikidata entity IDs corresponding to the linked articles. There is an unambiguous one-to-one mapping between Wikidata entities and Wikipedia articles. For example:

1: **Input** `Born in [[Honolulu|Honolulu, Hawaii]], Obama is a graduate of [[Columbia University]].`
2: **Links to Wikidata Ids** `Honolulu` ↦ `Q18094`
`Columbia_University` ↦ `Q49088`

For further processing, we filter out sentences that contain fewer than 3 annotated entities, since we need to have multiple relations per sentence for training (see Section 4).

We extract named entities and noun chunks from the input sentences with the Stanford CoreNLP toolkit (Manning et al., 2014) to identify entities that are not covered by the Wikipedia annotations (e.g. `Obama` in the sentence above). We retrieve IDs for those entities by searching through entity labels in Wikidata. We use HeidelTime (Strötgen and Gertz, 2013) to extract dates.

For each pair of entities, we query Wikidata for relation types that connect them. We discard an occurrence of an entity pair if the relation is ambiguous, i. e. multiple relation types were retrieved. For comparison, Surdeanu et al. (2012) report that only 7.5% of entity pairs have more than one corresponding relation type in the distantly supervised dataset of Riedel et al. (2010). The entity pairs that have no relation in the knowledge base are stored as negative instances.

The constructed dataset features 353 different relation types (out of approximately 1700 non-meta relation types in the Wikidata scheme). We split

[1]Unique IDs in Wikidata have a Q-prefix for entities and a P-prefix for relations.

[2]`https://www.wikidata.org/wiki/Special:Statistics`

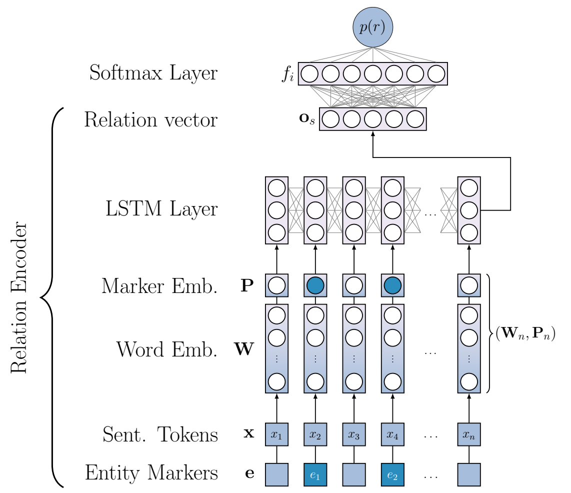

Figure 1: The architecture of the relation encoder

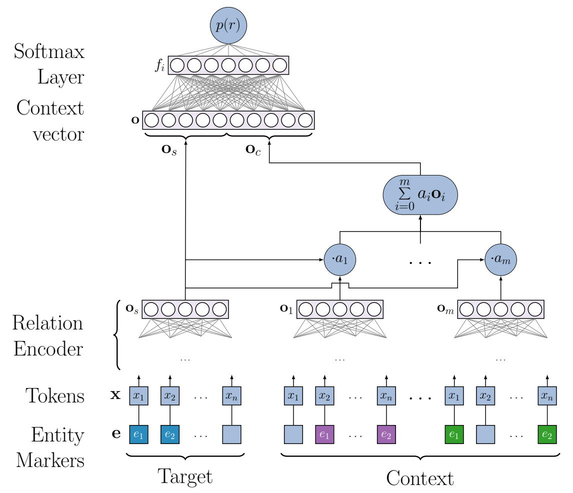

Figure 2: Incorporation of the context relations. For the ContextSum model variant $a_i = 1$.

it into train, validation and held-out sets, ensuring that there is no overlap in either sentences or relation triples between the three sets. Table 1 summarizes the statistics about the dataset. We assessed the quality of the distant supervision set-up on 200 manually verified sentences from the training set: 79.5% of relations in those sentences were correctly labeled with distant supervision (86.9 if one entity is linked, 74.7 if both are linked).

4 Model architecture

4.1 Relation encoder

The relation encoder produces a fixed-size vector representation $\mathbf{o}_s$ of a relation between two entities in a sentence (see Figure 1).

First, each token of the sentence $\mathbf{x} = \{x_1, x_2 \dots x_n\}$ is mapped to a k-dimensional embedding vector using a matrix $\mathbf{W} \in \mathbb{R}^{|V| \times k}$, where $|V|$ is the size of the vocabulary. Throughout the experiments in this paper, we use 50-dimensional GloVe embeddings pre-trained on a 6 billion corpus (Pennington et al., 2014).

Second, we mark each token in the sentence as either belonging to the first entity e_1, the second entity e_2 or to neither of those. A marker embedding matrix $\mathbf{P} \in \mathbb{R}^{3 \times d}$ is randomly initialized (d is the dimension of the position embedding and there are three marker types). For each token, we concatenate the marker embedding with the word embedding: $(\mathbf{W}_n, \mathbf{P}_n)$.

We apply a recurrent neural network (RNN) on the token embeddings. The length n naturally varies from sentence to sentence and an RNN provides a way to accommodate inputs of various sizes. It maps a sequence of n vectors to a fixed-size output vector $\mathbf{o}_s \in \mathbb{R}^o$. We take the output vector $\mathbf{o}_s$ as the representation of the relation between the target entities in the sentence. We use the Long Short-Term Memory (LSTM) variant of RNN (Hochreiter and Schmidhuber, 1997) that was successfully applied to information extraction before (Miwa and Bansal, 2016).

4.2 Model variants

LSTM baseline As the first model variant, we feed the output vector $\mathbf{o}_s$ of the relation encoder to a softmax layer to predict the final relation type for the target entity (see the upper part of Figure 1):

$$p(r|\langle e_1, e_2 \rangle, \mathbf{x}; \theta) = \frac{\exp(f_r)}{\sum_{i=1}^{n_r} \exp(f_i)}, \quad (1)$$
$$f_i = \mathbf{y}_i \cdot \mathbf{o}_s + b_i,$$

where $\mathbf{y}_i$ is a weight vector and b_i is a bias.

ContextSum We argue that for predicting a relation type for a target entity pair other context relations in the same sentence are relevant. Some relation types may tend to co-occur, such as DIRECTED_BY and PRODUCED_BY, whereas others may be restrictive (e. g. one can only have a single PLACE_OF_BIRTH).

Therefore, in addition to the target entity pair, we take other entities from the same sentence that were extracted at the data generation step. We construct a set of context relations by taking each possible pair of entities.[3] Example (2) shows a target entity pair $\langle e_1, e_2 \rangle$ and context entities highlighted in bold.

[3] We limit the maximum number of relations in a sentence to 7 for computational reasons.

(2) [**Swag It Out**] is the official [**debut single**] by [**American singer**] [e_1 **Zendaya**], known for starring in the series [e_2 **Shake It Up**].

We apply the same relation encoder on the target and context relations (see Figure 2). That ensures that representation for target and context relations are learned jointly. We sum the context relation representations: $\mathbf{o}_c = \sum_{i=0}^{m} \mathbf{o}_i$, where each element $\mathbf{o}_i$ is a vector representation of a single context relation. The resulting context representation $\mathbf{o}_c \in \mathbb{R}^o$ is concatenated with the vector representation of the target relation: $\mathbf{o} = [\mathbf{o}_s, \mathbf{o}_c]$. We feed the concatenated vector to the softmax layer in Eq. 1 to predict the final relation type for the target entity pair (see the upper part of Figure 2).

ContextAtt In this variant, we use a weighted sum of the context relation representation at the penultimate step:

$$\mathbf{o}_c = \sum_{i=0}^{m} a_i \mathbf{o}_i, \quad a_i = \frac{\exp(g(\mathbf{o}_i, \mathbf{o}_s))}{\sum_{j=0}^{m} \exp(g(\mathbf{o}_j, \mathbf{o}_s))}, \quad (2)$$

where g_i computes an attention score for a context relation with respect to the target relation: $g(\mathbf{o}_i, \mathbf{o}_s) = \mathbf{o}_i \mathbf{A} \mathbf{o}_s$, and $\mathbf{A}$ is a weight matrix that is learned.

5 Experiments

5.1 Training the models

All models were trained using the Adam optimizer (Kingma and Ba, 2014) with categorical cross-entropy as the loss function. We use an early stopping criterion on the validation data to determine the number of training epochs. The learning rate is fixed to 0.01 and the rest of the optimization parameters are set as recommended in Kingma and Ba (2014): $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon = 1e-08$. The training is performed in batches of 128 instances.

We apply Dropout (Srivastava et al., 2014) on the penultimate layer as well as on the embeddings layer with a probability of 0.5. We choose the size of the layers (RNN layer size $o = 256$) and entity marker embeddings ($d = 3$) with a random search on the validation set.[4]

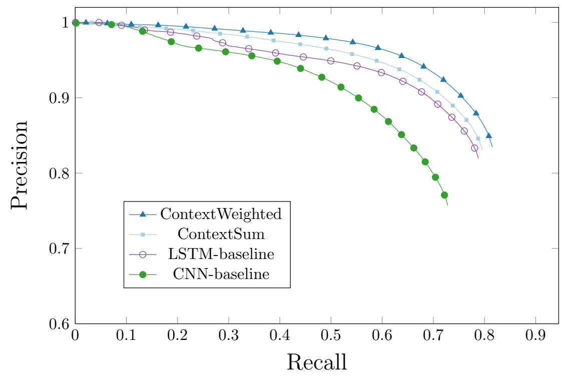

Figure 3: Aggregated precision-recall curves for the implemented models.

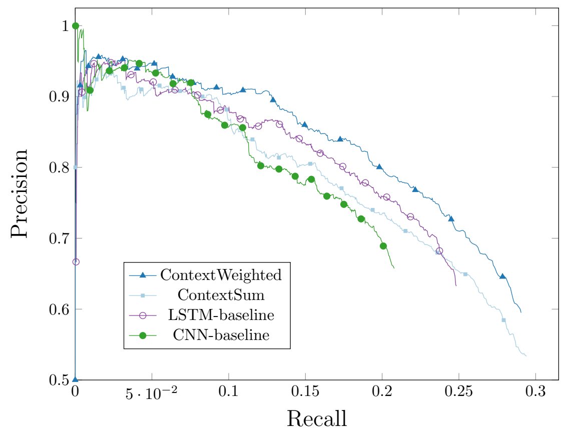

Figure 4: Aggregated macro precision-recall curves for the implemented models.

5.2 Held-out evaluation

As an additional baseline, we re-implement a sentence-level model based on convolutional neural networks (CNNs) described in Lin et al. (2016). This is a state-of-the-art model for fine-grained relation extraction that was previously tested on the single-relation dataset from Riedel et al. (2010). In addition to CNNs, their architecture uses a different position encoding scheme: position markers encode a relative position of each word with respect to the target entities.[5] We use the same GloVe word embeddings for this model and perform a hyper-parameter optimization on the validation set.

Our dataset lets us compare the baseline models and the models that use context relations on the same data. Following the previous work on rela-

[4] We test for the RNN layer size the values $\{64, 128, 256, 512\}$, for entity marker embeddings the values $\{1, 3, 5, 7\}$ and for the Dropout rate the values in the range 0.0–0.75.

[5] We have briefly experimented with such position markers for our models, but found no improvements.

Relation type	LSTM-Baseline *P*	*R*	ContextAtt *P*	*R*
COUNTRY	0.8899	0.9344	0.9130	0.9382
LOCATED IN	0.8329	0.8832	0.8655	0.8994
SHARES BORDER	0.7579	0.7078	0.7962	0.8075
INSTANCE OF	0.7864	0.8568	0.8478	0.8401
SPORT	0.9753	0.9828	0.9822	0.9823
CITIZENSHIP	0.9001	0.9448	0.9041	0.9417
PART OF	0.5623	0.4854	0.6269	0.5113
SUBCLASS OF	0.5230	0.4390	0.5272	0.5908

Table 2: Precision (*P*) and recall (*R*) for the top relations.

tion extraction, we report the aggregated precision-recall curves for each model on the held-out data (Figure 3).[6] To compute the curves, we rank the predictions of each model by their confidence and traverse this list top to bottom measuring the precision and recall at each step.

The models that take the context into account perform similar to the baselines at the smallest recall numbers, but start to positively deviate from them at higher recall rates. In particular, the ContextAtt model performs better than any other system in our study over the entire recall range. Compared to the competitive LSTM-baseline that uses the same relation encoder, the ContextAtt model achieves a 24% reduction of the average error: from 0.2096 ± 0.002 to 0.1590 ± 0.002. The difference between the models is statistically significant ($p = 0.009$).[7]

We also compute macro precision-recall curves that give equal weights to all relations in the dataset. Figure 4 shows that the ContextAtt model performs best over all relation types. One can also see that the ContextSum doesn't universally outperforms the LSTM-baseline. It demonstrates again that using attention is crucial to extract relevant information from the context relations.

On the relation-specific results (Table 2) we observe that the context-enabled model demonstrates the most improvement on precision and seems to be especially useful for taxonomy relations (see SUBCLASS OF, PART OF).

[6] We do not compare against the approach of Surdeanu et al. (2012) that also performs sentence-level relation extraction, since the provided implementation does not feature the complete pipeline and is only applicable on a particular Freebase dataset.

[7] The average error and the standard deviation are estimated on 5 training iterations for each model. The statistical significance is computed using the Wilcoxon rank-sum test on the error rates.

6 Conclusions

We have introduced a neural network architecture for relation extraction on the sentence level that takes into account other relations from the same context. We have shown by comparison with competitive baselines that these context relations are beneficial for relation extraction with a large set of relation types.

Our approach can be easily applied to other types of relation extraction models as well. For instance, Lin et al. (2016) extract sentence-level features and then combine features from multiple sentences with a selective attention mechanism. It would be possible to replace their sentence-level feature extractor with our model.

Acknowledgments

This work has been supported by the German Research Foundation as part of the Research Training Group Adaptive Preparation of Information from Heterogeneous Sources (AIPHES) under grant No. GRK 1994/1, and via the QA-EduInf project (grant GU 798/18-1 and grant RI 803/12-1). We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Tesla K40 GPU used for this research.

References

Michael Färber, Basil Ell, Carsten Menne, and Achim Rettinger. 2015. A Comparative Survey of DBpedia, Freebase, OpenCyc, Wikidata, and YAGO. *Semantic Web Journal*, 1:1–5.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1–32.

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S. Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT)*, pages 541–550, Portland, Oregon, USA.

Diederik Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *arXiv preprint*.

Qi Li, Heng Ji, and Liang Huang. 2013. Joint Event Extraction via Structured Prediction with Global Features. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 73–82, Sofia, Bulgaria.

Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural Relation Extraction with Selective Attention over Instances. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 2124–2133, Berlin, Germany.

Christopher D. Manning, John Bauer, Jenny Finkel, Steven J Bethard, Mihai Surdeanu, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics (ACL): System Demonstrations*, pages 55–60, Baltimore, Maryland, USA.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011, Singapore, Singapore.

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 (ACL)*, pages 1105–1116, Berlin, Germany.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling Relations and Their Mentions without Labeled Text. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 148–163, Barcelona, Spain.

Dan Roth and Wen-Tau Yih. 2004. A linear programming formulation for global inference in natural language tasks. In *Proceedings of the 8th Conference on Computational Natural Language Learning (CoNLL)*, pages 1–8, Boston, Massachusetts, USA.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

Jannik Strötgen and Michael Gertz. 2013. Multilingual and cross-domain temporal tagging. *Language Resources and Evaluation*, 47(2):269–298.

Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D. Manning. 2012. Multi-instance Multi-label Learning for Relation Extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 455–465, Jeju, Korea.

Andreas Vlachos and Sebastian Riedel. 2014. Fact Checking: Task definition and dataset construction. In *Proceedings of the ACL Workshop on Language Technologies and Computational Social Science*, pages 18–22, Baltimore, Maryland, USA.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: A Free Collaborative Knowledgebase. *Communications of the ACM*, 57(10):78–85.

Kun Xu, Siva Reddy, Yansong Feng, Songfang Huang, and Dongyan Zhao. 2016. Question Answering on Freebase via Relation Extraction and Textual Evidence. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 2326–2336, Berlin, Germany.

Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant Supervision for Relation Extraction via Piecewise Convolutional Neural Networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1753–1762, Lisbon, Portugal.

Han Zhao, Zhengdong Lu, and Pascal Poupart. 2015. Self-adaptive hierarchical sentence model. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, pages 4069–4076, Buenos Aires, Argentina.