

Relation Classification via Relation Validation

Jose G. Moreno

University of Toulouse
IRIT, UMR 5505 CNRS
F-31000, Toulouse, France
jose.moreno@irit.fr

Antoine Doucet

University of La Rochelle
L3i
F-17000, La Rochelle, France
antoine.doucet@univ-lr.fr

Brigitte Grau

LIMSI, UPR 3251 CNRS
F - 91405 Orsay , France
bg@limsi.fr

Abstract

Recognising if a relation holds between two entities in a text plays a vital role in information extraction. To address this problem, multiple models have been proposed based on fixed or contextualised word representations. In this paper, we propose a meta relation classification model that can integrate the most recent models by the use of a related task, namely relation validation. To do so, we encode the text that may contain the relation and a relation triplet candidate into a sentence-triplet representation. We grounded our strategy in recent neural architectures that allow single sentence classification as well as pair comparisons. Finally, our model is trained to determine the most relevant sentence-triplet pair from a set of candidates. Experiments on two public data sets for relation extraction show that the use of the sentence-triplet representation outperforms strong baselines and achieves comparable results when compared to larger models.

1 Introduction

Recognising and classifying relations between two entities in a text plays a vital role in knowledge base population (KBP), a major sub-task of information extraction (IE). Some examples of typical relations in knowledge bases (KB) are spouse, CEO, place of birth, profession, etc. Nowadays, there exist large KB that store millions of facts such as DBpedia (Bizer et al., 2009) or YAGO (Hoffart et al., 2013). However, more than 70% of people entities have not associated information for relations such as place of birth or nationality (Dong et al., 2014).

Most approaches model the relation classification (RC) (dos Santos et al., 2015; Nguyen and Grishman, 2015) task as a learning problem where it is required to predict if a passage contains a type of relation (multi-class classification). This setup requires annotated examples of each class, i.e. each

type of relation, which can be difficult to obtain. To overcome this problem, distant supervision has been proposed (Mintz et al., 2009) for automatically annotating texts given relation triplets existing in a KB by projecting triplets into texts to increase the input data. Its main counterpart is that distant supervision models must deal with wrongly annotated examples. The difficulty of the task is shown by the results of the TAC KBP slot filling task¹. For instance, in 2014, the maximum F1-score of the task was 0.3672 (Surdeanu and Ji, 2014). Another trend is trying to collect information directly from the web in an unsupervised setting, i.e. the open IE paradigm (Banko et al., 2007). In these two last settings, one crucial point is to be able to assess the validity of the extracted relations. This point motivated an extra track in TAC KBP 2015 following a divide-and-conquer setup. It consists in validating the relations extracted by relation extraction (RE) systems in order to improve their final scores.

The purpose of relation validation (RV) aims at taking advantage of several hypotheses, provided by one or several systems, for improving the recognition of relations in texts and discarding false ones. Given a candidate relation triplet $(e1, R, e2)$ and a passage, this task can be defined as learning to decide if the passage supports the relation in a binary classification setup. Trigger words and relation patterns are usually modelled in relation validation as features for representing the relation type. In Wang and Neumann (2008), the relation validation setup is modified and presented as an entailment problem, where systems learn whether the text entails the relation based on linguistic features.

In this paper, we propose not only to learn the representation of the relation type, but also to learn the representation of the validation knowledge by using a neural architecture for modelling relation

¹<https://catalog.ldc.upenn.edu/LDC2018T22>

validation, inspired by neural entailment models. We aim to decide whether the text supports the relation by encoding the text and the triplet² in a transformer architecture as in (Baldini Soares et al., 2019; Zhao et al., 2019). Once a model for relation validation is learned, we use it to validate the output of a relation classification model. Our experiments show that our proposal outperforms robust neural models for relation classification but fails to improve most recent works.

The remainder of this paper is structured as follows: Section 2 presents some relevant models for relation classification and validation. Section 3 details our strategy to classify relations based on relation validation. Then, the experimental setup and results are presented in Sections 4. Finally, conclusions are drawn in Section 5.

2 Related Work

Different ensemble models (Viswanathan et al., 2015) have been defined for the relation validation KBP task based on the prediction made by the RE systems. However, Yu et al. (2014) show that relation validation requires considering linguistic features for recognising if a relation is expressed in a text by exploiting rich linguistic knowledge from multiple lexical, syntactic, and semantic levels. In Wang and Neumann (2008), the relation to validate is transformed by simple patterns in a sentence and an alignment between the two texts is performed by a kernel-based approach.

Traditional methods for relation extraction are based on feature engineering and rely on lexical and syntactic information. Dependency trees provide clues for deciding the presence of a relation in unsupervised relation extraction (Culotta and Sorensen, 2004; Bunescu and Mooney, 2005; Fundel et al., 2007). Gamallo et al. (2012) defined patterns of relation by parsing the dependencies in open information extraction. Words around the entity mentions in sentences give clues to characterise the semantics of a relation (Niu et al., 2012; Hoffmann et al., 2011; Yao et al., 2011; Riedel et al., 2010; Mintz et al., 2009). In addition to linguistic information, collective information about the entities and their relations were exploited for RV (Rahman et al., 2018) by adding features based on a graph of entities and for RE by Augenstein

²We are aware that our model mainly based its improvements on input modification. However, we strongly believe that this is unfairly underestimated in the field.

(2016) that integrated global information about the object of a relation. The latter model shows the importance of adding information about the entities in the triplet. The above approaches rely on Natural Language Processing (NLP) tools for syntactic analysis and on lexical knowledge for identifying triggers. Thus, it remains difficult to overcome the lexical gap between texts and relation names when learning relation patterns for different types of relations in an open domain.

Recently, end-to-end neural network (NN) based approaches have been emerged and getting lots of attention for the relation classification task (dos Santos et al., 2015; Nguyen and Grishman, 2015; Vu et al., 2016; Dligach et al., 2017; Zheng et al., 2016; Zhang et al., 2018). However, they do not leverage any triplet representation of a relation for better understanding the relatedness between the text and the triplet. A lot of NN models for evaluating the similarity of two sentences have been proposed. They encode each entry by a CNN or an RNN (e.g., LSTM or BiLSTM), and compute a similarity between the sentence representations (Severyn and Moschitti, 2015) or compute interactions between the texts by an attention layer (Yin et al., 2016).

Most recent models encode one or two sentences by using the pre-trained neural models. Their use in RC has been successfully tested by Baldini Soares et al. (2019) where entities are marked and the sentence representation is used. Then a simple but effective sequence classification is performed using the sentence representation token which encodes the full sentence including the marked tokens. Their performances are boosted by using more documents in an unsupervised fashion. Despite more information being used, Baldini Soares et al. (2019) do not use an explicit relation representation. In an effort to cope with this problem, we explore the use of pre-trained neural models into the RV problem by explicitly using a triplet-sentence representation.

3 Relation classification via relation validation

Our proposal first learns how to validate relations ground on a sentence-triplet representation in order to predict if a relation stands or not in a sentence. To do so, our model is based on a pre-trained BERT model for sequence classification (Devlin et al., 2018). Using pre-trained models to address RC is

a promising strategy as shown by Baldini Soares et al. (2019). In both cases, i.e. RV or RC, a major consideration is the input definition to correctly identify the target entities, mainly because pre-trained models do not include this option by default. In this section, we present the details of the architecture together with the input transformations to correctly feed a sequence classification model such as BERT.

3.1 BERT-based Architecture

We opted for a simplified version³ of the architecture proposed in Baldini Soares et al. (2019) for relation classification, namely $BERT_{EM}$. It is based on fine-tuning of a pre-trained transformer called BERT (Devlin et al., 2018) where an extra layer is added to make the classification of the sentence representation, e.g. a classification task is performed using as input the [CLS] token. As reported by Baldini Soares et al. (2019), an important component is the use of mark symbols to identify the entities to classify.

3.2 Relation Classification

3.2.1 Problem definition

Given a tokenised sentence $S = "t_1 t_2 \dots t_n"$, an origin offset $o_o \in 1, n$, a target offset $o_t \in 1, n$, and a set of k relations $R = \{r_1, r_2, \dots, r_k\}$. The relation extraction problem consists in determining which relation $r_p \in R$ stands in the sentence between the tokens in positions o_o and o_t , respectively.⁴

3.2.2 Input considerations

We follow the input considerations for RC proposed by (Baldini Soares et al., 2019). Thus, to introduce those markers, the original input of RC models

$$\text{input}(S) = [\text{CLS}] \quad t_1 \quad t_2 \quad \dots \quad t_n \quad [\text{SEP}] \quad (1)$$

is modified to include the entities markers

$$\text{input}'(S) = [\text{CLS}] \quad \dots \$ \quad t_{o_o} \quad \$ \quad \dots \# \quad t_{o_t} \quad \# \dots \quad [\text{SEP}] \quad (2)$$

Note that $\text{length}(\text{input}'(S)) = \text{length}(\text{input}(S)) + 4$, because we added the tokens \$ and # twice.

³We used the EntityMarkers[CLS] version. Other configurations were not explored and are left for future work.

⁴Note that a *non-relation* or *other relation* may be part of the set R .

3.3 Relation Validation

3.3.1 Problem definition

Given a tokenised sentence $S = "t_1 t_2 \dots t_n"$, an origin offset $o_o \in 1, n$, a target offset $o_t \in 1, n$, and a triplet $t = \langle t_{o_o}, r, t_{o_t} \rangle$. The relation validation problem consists in determining whether the relation r between t_{o_o} and t_{o_t} is supported by the sentence S or not.

3.3.2 Input considerations

We transform triplets $t = \langle t_{o_o}, r, t_{o_t} \rangle$ into a sequence of its label words. Then we use the sentence S on one side and the triplet t on the other side as input of the model to match the relation validation problem into a text entailment setup as suggested by Wang and Neumann (2008). So, in this case, the input is modified to

$$\begin{aligned} \text{input}''(S) &= [\text{CLS}] \dots \$ \quad t_{o_o} \quad \$ \\ &\quad \dots \# \quad t_{o_t} \quad \# \dots [\text{SEP}] \quad (3) \\ t_{o_o} \quad t_{o_t} \quad r_{w_1} \quad r_{w_2} \dots \quad r_{w_m} [\text{SEP}] \end{aligned}$$

Note that $\text{length}(\text{input}''(S)) = \text{length}(\text{input}(S)) + 4 + (m + 2)$, because of the tokens \$ and #, and the triplet t is represented by $m + 2$ tokens (m words for the relation r and the two entities tokens). This architecture is possible because of the single or double input capabilities of transformer architectures such as BERT. Our proposed architecture is depicted in Figure 1. As for RC, we add the mark symbols in the sentence but not for the triplet. The final prediction is based on the sentence representation or the [CLS] token.

As our work focuses on relation extraction, a prior stage is needed to transform any relation classification data set into a relation validation one (i.e. as many examples as relations/classes). This transformation consists in generating $|R|$ relation validation examples for each relation extraction one, by considering the correct relation as positive and others as negatives. In this case, if \mathcal{S} is the set of examples for RC, then the set of examples for RV (\mathcal{S}_{RV}) is $|R|$ times larger than \mathcal{S} . However, to prevent imbalance, negative sampling is commonly used. In this case, $|\mathcal{S}_{RV}| = (ns + 1) \times |\mathcal{S}|$ where ns is the number of negative examples used to build \mathcal{S}_{RV} .

3.4 Validation of a classification prediction

Our main contribution is the definition of a new model for RC using RV, namely $BERT+RC+RV$.

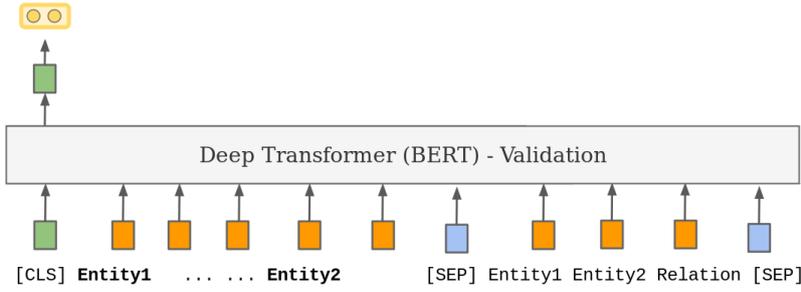


Figure 1: Our relation validation model. Tokens in bold are marked using "\$" for the Entity1 and "#" for the Entity2.

During training time our RV model behaves as described in Algorithm 1. The set $\mathcal{S}_{\mathcal{R}\mathcal{V}}$ used as input is built as described in Section 3.3.2. createInput generates an input such as in Equation 2. The output is a relation validation model (M_{RV}) capable of detecting if the input is valid or not.

Algorithm 1: BERT+RC+RV train

Input: Set of examples $\mathcal{S}_{\mathcal{R}\mathcal{V}}$ {Sentence (S), triplet (t), label (l_{RV})}

$epoch = 1$

while $epoch < max_{epochs}$ **do**

for $S, t, l_{RV} \in \mathcal{S}_{\mathcal{R}\mathcal{V}}$ **do**

$input''(S) = createInput(S, t)$

 update with Loss($input''(S), l_{RV}$)

Output: Validation model (M_{RV})

On the other hand, at inference time not all cases are evaluated. Our model can use as input the outputs of multiples RC models⁵ (\mathcal{S}_v) as described in Algorithm 2. Each example in \mathcal{S}_v is composed of a sentence and n_{RC} labels predicted by n_{RC} RC models, i.e. each example has a list (L) of n_{RC} predictions. Thus, our RV model defines the most suitable label based on the sentence and the triplet together instead of a classic RC model that only uses the sentence. getTriplet is a function based on a simple dictionary that returns the relation words (r_{w_1}, \dots, r_{w_m}) related to a label l_{rc} and the entities (t_{o_o} and t_{o_t}) in S . This way, our model is not only capable of learning from the same data but also capable of aggregating multiple RC predictions.

⁵In our experiments, we used the outputs of our implementation of a state-of-the-art RC model, $BERT_{EM}$, described in Section 4.2.

Algorithm 2: BERT+RC+RV prediction

Input: Set of examples to validate \mathcal{S}_v {Sentence (S), labels (L)}, a Validation model (M_{RV})

$l_V = []$

for $S, L \in \mathcal{S}_v$ **do**

$l_{i-valid} = []$

for $l_i \in unique(L)$ **do**

$t = getTriplet(l_i, S)$

$input''(S) = createInput(S, t)$

$confid = predict(M_{RV}, input''(S))$

$l_{i-valid}.append(l_i, confid)$

$l_V.append(labelMaxConfidence(l_{i-valid}))$

Output: List of predictions (l_V)

4 Experiments and Results

4.1 Data Sets

In this study, we experimented on two publicly available data set: *SemEval10*⁶ and *TACRED*⁷. Statistics of these standard relation classification data sets are presented in Table 1. We created a relation validation version from both data sets as described in Section 3.3.2. The input of our RV model needs a set of relation words which, originally, are not present in the data sets. Thus, to obtain these words, we used a rather simple strategy that consists of tokenising the relations names and using them as relation words. If needed it considers the relation direction by reversing the position of the tokenised words. Table 2 shows some examples of the selected words.

In both cases, we used the respective official F_1 metric⁸ for evaluation.

⁶Task 8 (Hendrickx et al., 2010) from <http://semeval2.fbk.eu/semeval2.php?location=tasks>

⁷<https://nlp.stanford.edu/projects/tacred/>

⁸Macro-F1-measures are calculated using each script. Both scripts exclude the *other* class during evaluation.

Data set	Train	Dev	Test	# Relations
SemEval10	8000	-	2717	19
TACRED	68124	22631	15509	42

Table 1: Summary of SemEval10 and TACRED data sets for relation classification.

4.2 Implementation details

We implemented $BERT_{EM}$ (*EntityMarkers[CLS]* version) of Baldini Soares et al. (2019) for RC and adapted it to perform RV⁹. For SemEval10, we used 10% of training data as validation data which allows fair comparison against previous works. A maximum number of epochs was fixed to 5 and the best epoch in validation used for prediction¹⁰. Negative sampling was fixed to 10 where the input sentence remains and the entities remain the same but the words used for the relation representation ($r_{w_1}, r_{w_2}, \dots, r_{w_m}$) are sampled from other classes. *Binary Cross Entropy* was used as loss function, Adam as optimiser, *bert-base-uncased*¹¹ as pre-trained model, and other parameters were assigned following the library recommendations (Wolf et al., 2019).¹² The final layer is composed of as many neurons as classes in each data set for RC and equal to two for RV (negative or positive).

Data set	Relation	Words
SemEval10	Cause-Effect(e1,e2)	Cause, Effect
	Cause-Effect(e2,e1)	Effect, Cause
	Content-Container(e1,e2)	Content, Container
TACRED	org:founded.by	org, founded, by
	per:city_of_death	per, city, of, death
	per:age	per, age

Table 2: Examples of words used per relation.

4.3 Results

Average and best result of 5 runs of our implementation of (Baldini Soares et al., 2019) using the SemEval10 data set are presented in Table 3 ($BERT_{EM}^*$). The reported results are within the values reported in the original paper for this configuration, but we used *bert-base-uncased* instead

⁹Our code is publicly available at <https://github.com/jgmorenof/rcviarv2020>.

¹⁰Our models got the best validation performances at epoch 5, no further epochs were explored.

¹¹<https://github.com/google-research/bert>

¹²We did not perform parameters search.

	SemEval10	TACRED
$BERT_{EM}^*$ - average	87.03	65.50
$BERT_{EM}^*$ - best	87.70	66.02
$BERT+RC+RV$ - average (ours)	88.36	66.20
$BERT+RC+RV$ - best (ours)	88.44	67.48
$BERT_{EM}^*$ - voting	89.02	68.67
$BERT+RC+RV$ - voting (ours)	89.41	69.13
<i>TRE</i> (Alt et al., 2019)	87.1	67.4
<i>BERT-LSTM-base</i> (Shi and Lin, 2019)	-	67.8
<i>C-GCN+PALSTM</i> (Zhang et al., 2018)	-	68.2
<i>C-AGGCN</i> (Guo et al., 2019)	-	68.2
<i>Att-Pooling-CNN</i> (Wang et al., 2016)	88.0	-
<i>Entity-Aware BERT</i> (Wang et al., 2019)	89.0	-
<i>KnowBert-W+W</i> (Peters et al., 2019)	89.1	<u>71.5</u>
<i>R-BERT</i> (Wu and He, 2019)	89.25	-
$BERT_{EM}$ (Baldini Soares et al., 2019)	89.2	<u>70.1</u>
<i>Span-BERT</i> (Joshi et al., 2019)	-	<u>70.8</u>
$BERT_{EM}+MTB$ (Baldini Soares et al., 2019)	<u>89.5</u>	<u>71.5</u>
<i>EPGNN</i> (Zhao et al., 2019)	<u>90.2</u>	-

Table 3: Results of official F_1 metric for the SemEval10 and TACRED data sets. Best result of our tested models is marked in **bold**. Results that outperform our method are underlined. '*' indicates that the result was obtained by our implementation of (Baldini Soares et al., 2019). Other values were taken from referenced papers.

	Number of candidates					
	2		3		4	
	Corr.	Incorr.	Corr.	Incorr.	Corr.	Incorr.
$BERT+RC+RV$	338	154	37	52	2	4
	68.69%	31.30%	41.57%	58.42%	33.33%	66.66%

Table 4: Percentage of correct (Corr.) and incorrect (Incorr.) predictions from RV model for the SemEval10 data set grouped by the number of candidates provided by RC.

	Epoch				
	1	2	3	4	5
$BERT+RC+RV$	0.8790	0.8807	0.8793	0.8802	0.8831
$BERT_{EM}^*$ - run1	-	-	-	-	0.8760
$BERT_{EM}^*$ - run2	-	-	-	-	0.8683
$BERT_{EM}^*$ - run3	-	-	-	-	0.8688
$BERT_{EM}^*$ - run4	-	-	-	-	0.8770
$BERT_{EM}^*$ - run5	-	-	-	-	0.8614

Table 5: Performances for one run of our method vs $BERT_{EM}$ runs in terms of F_1 using the SemEval10 data set. We calculated our results by epoch after training.

of *bert-large-uncased* due to computational constraints. In both cases, for average and best, our results using the relation validation model outperform their counterparts by a non-negligible margin. In order to understand the cases in which *BERT+RC+RV* makes the right prediction, we have reported the percentage of correct and incorrect predictions grouped by the number of candidates in Table 4. Note that at this stage *BERT+RC+RV* does not consider the number of predictions made for a candidate (as is made by voting) but analyse each candidate independently of its popularity. Although we used 5 runs, none of the examples obtained five candidates as for every test example at least two models predicted the same class. The number of correct predictions made by our validation model is 68.69% when there are only 2 candidates but decreases as the number of candidates increase (down to 33.33% for 4 candidates). However, in most of the cases, the predictions of the relation classification model only get 2 candidates (83.81%). Clearly, this result shows that there is still room for improvement by proposing better RV models.

Following this direction, we apply majority voting¹³ over the predictions of *BERT_{EM}* and *BERT+RC+RV*. Results are included in Table 3. Note that voting benefits our baseline but also our method by a similar margin. The lower part of Table 3 allows comparing our results to those of the most recent RC models. The best result, giving an F_1 score of 0.8941 is obtained based on majority voting of the prediction from the RV model. When compared against results reported in SemEval10, our method achieves the third position slightly behind *BERT_{EM}+MTB*, but quite far from *EPGNN* (Zhao et al., 2019). However, *BERT+RC+RV* remains an easy-to-implement model as no special modification is needed when compared with *BERT_{EM}+MTB* which uses extra auto-supervised training plus a larger model¹⁴ and *EPGNN* which needs graph embeddings. Moreover, we believe that *BERT_{EM}+MTB* can be improved if more robust models are validated.

We also studied the performance of our method by epoch, as reported in Table 5. Results of *BERT_{EM}** are presented for epoch 5 as this epoch got the best validation result. Note that our method

¹³The class that receives the highest number of votes will be chosen.

¹⁴*bert-large-uncased* uses three times more parameters (340 millions) than *bert-base-uncased* (110 millions).

outperforms all individual RC predictions from the first epoch and no underperformance is observed across epochs. This result suggests that our method is an effective way to mixture RC predictions.

Finally, we experimented with our model using the TACRED data set. Results are reported in Table 3. The results follow the same pattern as with the SemEval10 data set, except for one important difference: The performance obtained with *BERT_{EM}** ($F_1 = 65.50$) is much lower than the value reported by the authors ($F_1 = 69.13$). This can be explained from the fact that the number of relations in TACRED is twice as high as in SemEval10. Subsequently, more parameters allowed a richer representation and a better starting point (+4.5 absolute points w.r.t. F_1).

5 Conclusion

In this paper, we presented a new strategy to improve the neural models for relation classification by using relation validation knowledge, i.e. the sentence-triplet representation. Experiments with two public data sets experimentally support our hypothesis. The proposed strategy enables new ways to improve existing methods as it can be easily plugged into more recent (or future) and powerful models. Future work will be focused on the use of this strategy across tasks from different (and far) domains as our relation validation architecture can validate triplets with unseen relations. This opened an interesting research direction for relation classification by focusing more on triplet-sentence representations rather than exclusively on the sentence.

Acknowledgements

This work has been partly supported by the European Union’s Horizon 2020 research and innovation programme under grant 825153 (EMBEDDIA).

We also thank the anonymous reviewers for their careful reading of this paper and their many insightful comments and suggestions.

References

- Christoph Alt, Marc Hübner, and Leonhard Henning. 2019. Improving Relation Extraction by Pre-trained Language Representations. In *Proceedings of AKBC*.
- Isabelle Augenstein. 2016. *Web Relation Extraction with Distant Supervision*. Ph.D. Dissertation. University of Sheffield.

- Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the Blanks: Distributional Similarity for Relation Learning. In *Proceedings of the 57th Annual Meeting of the ACL*. 2895–2905.
- Michele Banko, Michael J Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. 2007. Open Information Extraction from the Web. In *IJ-CAI*, Vol. 7. 2670–2676.
- Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. 2009. DBpedia-A crystallization point for the Web of Data. *Journal of web semantics* 7, 3 (2009), 154–165.
- Razvan C Bunescu and Raymond J Mooney. 2005. A shortest path dependency kernel for relation extraction. In *Proceedings of the conference on HLT and EMNLP*. Association for Computational Linguistics, 724–731.
- Aron Culotta and Jeffrey Sorensen. 2004. Dependency Tree Kernels for Relation Extraction. In *Proceedings of the 42nd Annual Meeting on ACL*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805* (2018).
- Dmitriy Dligach, Timothy Miller, Chen Lin, Steven Bethard, and Guergana Savova. 2017. Neural Temporal Relation Extraction. *EACL 2017* (2017), 746.
- Xin Dong, Evgeniy Gabilovich, Jeremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmman, Shaohua Sun, and Wei Zhang. 2014. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 601–610.
- Cicero dos Santos, Bing Xiang, and Bowen Zhou. 2015. Classifying Relations by Ranking with Convolutional Neural Networks. In *Proceedings of the 53rd Annual Meeting of ACL and the 7th International JCNLP*. 626–634.
- Katrin Fundel, Robert Küffner, and Ralf Zimmer. 2007. RelEx-Relation extraction using dependency parse trees. *Bioinformatics* 23, 3 (2007), 365–371.
- Pablo Gamallo, Marcos Garcia, and Santiago Fernández-Lanza. 2012. Dependency-based open information extraction. In *Proceedings of the joint workshop on unsupervised and semi-supervised learning in NLP*. 10–18.
- Zhijiang Guo, Yan Zhang, and Wei Lu. 2019. Attention Guided Graph Convolutional Networks for Relation Extraction. In *Proceedings of the 57th Annual Meeting of ACL*. 241–251.
- Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2010. SemEval-2010 Task 8: Multi-Way Classification of Semantic Relations between Pairs of Nominals. In *Proceedings of the 5th SemEval*. 33–38.
- Johannes Hoffart, Fabian M Suchanek, Klaus Berberich, and Gerhard Weikum. 2013. YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia. *Artificial Intelligence* 194 (2013), 28–61.
- 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 ACL*. 541–550.
- Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2019. Spanbert: Improving pre-training by representing and predicting spans. *arXiv preprint arXiv:1907.10529* (2019).
- Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the 47th Annual Meeting of the ACL and the 4th International JCNLP of the AFNLP*. 1003–1011.
- Thien Huu Nguyen and Ralph Grishman. 2015. Relation extraction: Perspective from convolutional neural networks. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*. 39–48.
- Feng Niu, Ce Zhang, Christopher Ré, and Jude W Shavlik. 2012. DeepDive: Web-scale Knowledge-base Construction using Statistical Learning and Inference. *VLDS 12* (2012), 25–28.
- Matthew E. Peters, Mark Neumann, Robert L Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. Knowledge Enhanced Contextual Word Representations. In *EMNLP*.
- Rashedur Rahman, Brigitte Grau, and Sophie Rosset. 2018. Impact of Entity Graphs on Extracting Semantic Relations. In *Information Management and Big Data*. 31–47.
- Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Machine Learning and Knowledge Discovery in Databases*. Springer, 148–163.
- Aliaksei Severyn and Alessandro Moschitti. 2015. Learning to Rank Short Text Pairs with Convolutional Deep Neural Networks. In *Proceedings of the 38th International ACM SIGIR*. 373–382.
- Peng Shi and Jimmy Lin. 2019. Simple BERT Models for Relation Extraction and Semantic Role Labeling. *arXiv preprint arXiv:1904.05255* (2019).

- Mihai Surdeanu and Heng Ji. 2014. Overview of the english slot filling track at the tac2014 knowledge base population evaluation. In *Proc. TAC*.
- Vidhoon Viswanathan, Nazneen Fatema Rajani, Yinon Bentor, and Raymond Mooney. 2015. Stacked Ensembles of Information Extractors for Knowledge-Base Population. In *Proceedings of the 53rd Annual Meeting of ACL and the 7th International JCNLP*. 177–187.
- Ngoc Thang Vu, Heike Adel, Pankaj Gupta, and Hinrich Schütze. 2016. Combining Recurrent and Convolutional Neural Networks for Relation Classification. In *Proceedings of the 2016 Conference of the NAACL-HTL*. 534–539.
- Haoyu Wang, Ming Tan, Mo Yu, Shiyu Chang, Dakuo Wang, Kun Xu, Xiaoxiao Guo, and Saloni Potdar. 2019. Extracting Multiple-Relations in One-Pass with Pre-Trained Transformers. In *Proc. of the 57th Annual Meeting of ACL*. 1371–1377.
- Linlin Wang, Zhu Cao, Gerard de Melo, and Zhiyuan Liu. 2016. Relation Classification via Multi-Level Attention CNNs. In *Proceedings of the 54th Annual Meeting of the ACL*. 1298–1307.
- Rui Wang and Günter Neumann. 2008. Relation validation via textual entailment. *Ontology-based information extraction systems (obies 2008)* (2008).
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. HuggingFace’s Transformers: State-of-the-art Natural Language Processing. *arXiv:1910.03771* (2019).
- Shanchan Wu and Yifan He. 2019. Enriching Pre-trained Language Model with Entity Information for Relation Classification. In *CIKM*.
- Limin Yao, Aria Haghighi, Sebastian Riedel, and Andrew McCallum. 2011. Structured relation discovery using generative models. In *Proceedings of the Conference on EMNLP*. 1456–1466.
- Wenpeng Yin, Hinrich Schütze, Bing Xiang, and Bowen Zhou. 2016. ABCNN: Attention-Based Convolutional Neural Network for Modeling Sentence Pairs. *Transactions of the Association for Computational Linguistics* 4 (2016), 259–272.
- Dian Yu, Hongzhao Huang, Taylor Cassidy, Heng Ji, Chi Wang, and et al. 2014. The Wisdom of Minority: Unsupervised Slot Filling Validation based on Multi-dimensional Truth-Finding, In *Proceedings of 2014 International CICLING*.
- Yuhao Zhang, Peng Qi, and Christopher D. Manning. 2018. Graph Convolution over Pruned Dependency Trees Improves Relation Extraction. In *Proceedings of the 2018 Conference on EMNLP*. 2205–2215.
- Yi Zhao, Huaiyu Wan, Jianwei Gao, and Youfang Lin. 2019. Improving Relation Classification by Entity Pair Graph. In *ACML*. 1156–1171.
- Suncong Zheng, Jiaming Xu, Peng Zhou, Hongyun Bao, Zhenyu Qi, and Bo Xu. 2016. A neural network framework for relation extraction: Learning entity semantic and relation pattern. *Knowledge-Based Systems* 114 (2016), 12–23.