A Case Study on Learning a Unified Encoder of Relations

Lisheng Fu Bonan Min[†] Thien Huu Nguyen[*] Ralph Grishman
New York University, New York, NY, USA
{lisheng, grishman}@cs.nyu.edu
[†] Raytheon BBN Technologies, Cambridge, MA, USA
bonan.min@raytheon.com
[*] University of Oregon, Eugene, OR, USA
thien@cs.uoregon.edu

Abstract

Typical relation extraction models are trained on a single corpus annotated with a pre-defined relation schema. An individual corpus is often small, and the models may often be biased or overfitted to the corpus. We hypothesize that we can learn a better representation by combining multiple relation datasets. We attempt to use a shared encoder to learn the unified feature representation and to augment it with regularization by adversarial training. The additional corpora feeding the encoder can help to learn a better feature representation layer even though the relation schemas are different. We use ACE05 and ERE datasets as our case study for experiments. The multi-task model obtains significant improvement on both datasets.

1 Introduction

Relations represent specific semantic relationships between two entities. For example, there is Physical.Located relationship between *Smith* and *Brazil* in the sentence: *Smith* went to a conference in *Brazil*. Relation extraction is a crucial task for many applications such as knowledge base population. Several relation schemas and annotated corpora have been developed such as the Automatic Content Extraction (ACE), and the Entities, Relations and Events (ERE) annotation (Song et al., 2015). These schemas share some similarity, but differ in details. A relation type may exist in one schema but not in another. An example might be annotated as different types in different datasets. For example, Part-whole.Geographical relations in ACE05 are annotated as Physcial.Located relations in ERE. Most of these corpora are relatively small. Models trained on a single corpus may be biased or overfitted towards the corpus.

Despite the difference in relation schemas, we hypothesize that we can learn a more general representation with a unified encoder. Such a representation could have better out-of-domain or low-resource performance. We develop a multi-task model to learn a representation of relations in a shared relation encoder. We use separate decoders to allow different relation schemas. The shared encoder accesses more data, learning less overfitted representation. We then regularize the representation with adversarial training in order to further enforce the sharing between different datasets. In our experiments, we take ACE05 [1] and ERE [2] datasets as a case study. Experimental results show that the model achieves higher performance on both datasets.

2 Related Work

Relation extraction is typically reduced to a classification problem. A supervised machine learning model is designed and trained on a single dataset to predict the relation type of pairs of entities. Traditional methods rely on linguistic or semantic features (Zhou et al., 2005; Jing and Zhai, 2007), or kernels based on syntax or sequences (Bunescu and Mooney, 2005a,b; Plank and Moschitti, 2013) to represent sentences of relations. More recently, deep neural nets start to show promising results. Most rely on convolutional neural nets (Zeng et al., 2014, 2015; Nguyen and Grishman, 2015, 2016; Fu et al., 2017) or recurrent neural nets (Zhang et al., 2015; Zhou et al., 2016; Miwa and Bansal, 2016) to learn the representation of relations. Our supervised base model will be similar to (Zhou et al., 2016). Our initial experiments did not use syntactic features (Nguyen and Grishman, 2016; Fu et al., 2017) that require additional parsers.

[1] https://catalog.ldc.upenn.edu/LDC2006T06

[2] We use 6 LDC releases combined: LDC2015E29, LDC2015E68, LDC2015E78, LDC2015R26, LDC2016E31, LDC2016E73

In order to further improve the representation learning for relation extraction, Min et al. (2017) tried to transfer knowledge through bilingual representation. They used their multi-task model to train on the bilingual ACE05 datasets and obtained improvement when there is less training available (10%-50%). Our experiments will show our multi-task model can make significant improvement on the full training set.

In terms of the regularization to the representation, Duong et al. (2015) used l2 regularization between the parameters of the same part of two models in multi-task learning. Their method is a kind of soft-parameter sharing, which does not involve sharing any part of the model directly. Fu et al. (2017) applied domain adversarial networks (Ganin and Lempitsky, 2015) to relation extraction and obtained improvement on out-of-domain evaluation. Inspired by the adversarial training, we attempt to use it as a regularization tool in our multi-task model and find some improvement.

3 Supervised Neural Relation Extraction Model

The supervised neural model on a single dataset was introduced by Zeng et al. (2014) and followed by many others (Nguyen and Grishman, 2015; Zhou et al., 2016; Miwa and Bansal, 2016; Nguyen and Grishman, 2016; Fu et al., 2017). We use a similar model as our base model. It takes word tokens, position of arguments and their entity types as input. Some work (Nguyen and Grishman, 2016; Fu et al., 2017) used extra syntax features as input. However, the parsers that produce syntax features could have errors and vary depending on the domain of text. The syntax features learned could also be too specific for a single dataset. Thus, we focus on learning representation from scratch, but also compare the models with extra features later in the experiments. The encoder is a bidirectional RNN with attention and the decoder is one hidden fully connected layer followed by a softmax output layer.

In the input layer, we convert word tokens into word embeddings with pretrained word2vec (Mikolov et al., 2013). For each token, we convert the distance to the two arguments of the example to two position embeddings. We also convert the entity types of the arguments to entity embeddings. The setup of word embedding and position embedding was introduced by Zeng et al. (2014). The entity embedding (Nguyen and Grishman, 2016; Fu et al., 2017) is included for arguments that are entities rather than common nouns. At the end, each token is converted to an embedding w_i as the concatenation of these three types of embeddings, where $i \in [0, T)$, T is the length of the sentence.

A wide range of encoders have been proposed for relation extraction. Most of them fall into categories of CNN (Zeng et al., 2014), RNN (Zhou et al., 2016) and TreeRNN (Miwa and Bansal, 2016). In this work, we follow Zhou et al. (2016) to use Bidirectional RNN with attention (BiRNN), which works well on both of the datasets we are going to evaluate on. BiRNN reads embeddings of the words from both directions in the sentence. It summarizes the contextual information at each state. The attention mechanism aggregates all the states of the sentence by paying more attention to informative words. Given input w_i from the input layer, the encoder is defined as the following:

$$\overrightarrow{h_i} = \overrightarrow{GRU}(w_i, h_{i-1}), \quad (1)$$

$$\overleftarrow{h_i} = \overleftarrow{GRU}(w_i, h_{i-1}), \quad (2)$$

$$h_i = concatenate(\overrightarrow{h_i}, \overleftarrow{h_i}) \quad (3)$$

$$v_i = tanh(W_v h_i + b_v), \quad (4)$$

$$\alpha_i = \frac{exp(v_i^\top v_w)}{\sum_t exp(v_t^\top v_w))}, \quad (5)$$

$$\phi(x) = \sum_i \alpha_i h_i. \quad (6)$$

We use GRU (Cho et al., 2014) as the RNN cell. W_v and b_v are the weights for the projection v_i. v_w is the word context vector, which works as a query of selecting important words. The importance of the word is computed as the similarity between v_i and v_w. The importance weight is then normalized through a softmax function. Then we obtain the high level summarization $\phi(x)$ for the relation example.

The decoder uses this high level representation as features for relation classification. It usually contains one hidden layer (Zeng et al., 2014; Nguyen and Grishman, 2016; Fu et al., 2017) and a softmax output layer. We use the same structure which can be formalized as the following:

$$h = ReLU(W_h \phi(x) + b_h), \quad (7)$$

$$p = softmax(W_o h + b_o), \quad (8)$$

where W_h and b_h are the weights for the hidden

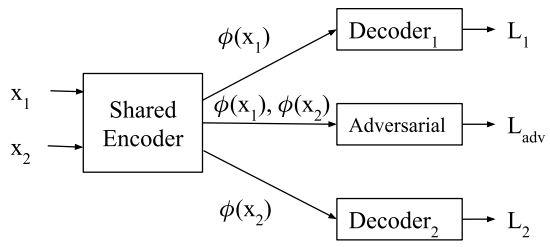

Figure 1: Multi-task model with regularization

layer, W_o and b_o are the weights for the output layer. We use cross-entropy as the training loss.

4 Learning Unified Representation

While the data for one relation task may be small, noisy and biased, we can learn a better representation combining multiple relation tasks. We attempt to use multi-task learning to learn a unified representation across different relation tasks. The method is simple and straightforward. We use the same encoder to learn the unified feature representation for both relation tasks, and then we train classifiers for each task on top of this representation. We then apply regularization on this representation by adversarial training.

4.1 Multi-task Learning

Given example x_1 from relation schema 1 and x_2 from relation schema 2, we use the same encoder to obtain representation $\phi(x_1)$ and $\phi(x_2)$ respectively. Then we build separate decoders for them using the same structure (7) (8). To train them at the same time, we put examples from both tasks in the same batch. The ratio of the examples are controlled so that the the model reads both datasets once every epoch. We use linear interpolation to combine the loss from them.

$$L = (1 - \lambda)L_1 + \lambda L_2, \tag{9}$$

where λ is used to control the attention to each task. The model may learn the two tasks at different speed. During optimization, one task can be seen as the main task, while the other can be seen as the auxiliary task. The benefit of joint learning to the main task may vary depending on how much attention the model pays to the auxiliary task.

4.2 Regularization by Adversarial Training

Being optimized simultaneously by different decoders, the model could still learn very different representation for similar examples coming from different tasks. We want to prevent this and to further push the model to learn similar representation for similar examples even if they come from different tasks. We attempt to regularize the representation using adversarial training between the two tasks.

Given the representation $\phi(x_1)$ and $\phi(x_2)$ learned from the two tasks, we build a classifier to predict which task the examples come from (11). We add a gradient reversal layer (Ganin and Lempitsky, 2015) at the input of this classifier (10) to implement the adversarial training.

$$\phi(x) = GRL(\phi(x)), \tag{10}$$

$$p = softmax(W\phi(x) + b). \tag{11}$$

While the classifier learns to distinguish the sources of the input representation, the input representation is learned in the opposite direction to confuse the classifier thanks to GRL. Thus, the input representation ($\phi(x_1)$ and $\phi(x_2)$) will be pushed to be close to each other. The gradient reversal layer (GRL) is defined as the identity function for forward propagation (12) and reversed gradient for back propagation (13).

$$GRL(x) = x, \tag{12}$$

$$\frac{dGRL(x)}{dx} = -I. \tag{13}$$

We also use the cross-entropy loss for this adversarial training, and combine the loss L_{adv} with the two relation tasks.

$$L = (1 - \lambda)L_1 + \lambda L_2 + \beta L_{adv}, \tag{14}$$

where we can use β to control how close the representations are between the two relation tasks.

5 Experiments

5.1 Datasets

To apply the multi-task learning, we need at least two datasets. We pick ACE05 and ERE for our case study. The ACE05 dataset provides a cross-domain evaluation setting . It contains 6 domains: broadcast conversation (bc), broadcast news (bn), telephone conversation (cts), newswire (nw), usenet (un) and weblogs (wl). Previous work (Gormley et al., 2015; Nguyen and Grishman, 2016; Fu et al., 2017) used newswire as training set (bn & nw), half of bc as the development

Training Data	100%					50%				
	ACE05				ERE	ACE05				ERE
Method	**bc**	**wl**	**cts**	**avg**	**test**	**bc**	**wl**	**cts**	**avg**	**test**
Supervised	61.44	52.40	52.38	55.40	55.78	56.03	47.81	48.65	50.83	53.60
Pretraining	60.21	53.34	56.10	56.55	56.39	55.39	49.17	52.91	52.49	54.66
Multi-task	61.67	55.03	**56.47**	57.72	57.29	57.39	51.44	54.28	54.37	55.72
+ Regularization	**62.24**	**55.30**	56.27	**57.94**	**57.75**	**57.73**	**52.30**	**54.63**	**54.89**	**55.91**

Table 1: Multi-task Learning and Regularization.

set, and the other half of bc, cts and wl as the test sets. We followed their split of documents and their split of the relation types for asymmetric relations. The ERE dataset has a similar relation schema to ACE05, but is different in some annotation guidelines (Aguilar et al., 2014). It also has more data than ACE05, which we expect to be helpful in the multi-task learning. It contains documents from newswire and discussion forums. We did not find an existing split of this dataset, so we randomly split the documents into train (80%), dev (10%) and test (10%).

5.2 Model Configurations

We use word embedding pre-trained on newswire with 300 dimensions from word2vec (Mikolov et al., 2013). We fix the word embeddings during the training. We follow Nguyen and Grishman (2016) to set the position and entity type embedding size to be 50. We use 150 dimensions for the GRU state, 100 dimensions for the word context vector and use 300 dimensions for the hidden layer in the decoders. We train the model using Adam (Kingma and Ba, 2014) optimizer with learning rate 0.001. We tune λ linearly from 0 to 1, and β logarithmically from $5 \cdot 10^{-1}$ to 10^{-4} For all scores, we run experiments 10 times and take the average.

5.3 Augmentation between ACE05 and ERE

Training separately on the two corpora (row "Supervised" in Table 1), we obtain results on ACE05 comparable to previous work (Gormley et al., 2015) with substantially fewer features. With syntactic features as (Nguyen and Grishman, 2016; Fu et al., 2017) did, it could be further improved. In this paper, however, we want to focus on representation learning from scratch first. Our experiments focus on whether we can improve the representation with more sources of data.

A common way to do so is pre-training. As a baseline, we pre-train the encoder of the supervised model on ERE and then fine-tune on ACE05, and vice versa (row "Pretraining" in Table 1). We observe improvement on both fine-tuned datasets. This shows the similarity between the encoders of the two datasets. However, if we fix the encoder trained from one dataset, and only train the decoder on the other dataset, we will actually obtain a much worse model. This indicates that neither dataset contains enough data to learn a universal feature representation layer for classification. This leaves the possibility to further improve the representation by learning a better encoder.

We then attempt to learn a multi-task model using a shared encoder. We use 14K words as the vocabulary from ACE05 and 20K from ERE. There are about 8K words shared by the two datasets (same for both pretrained and multi-task models). By multi-task learning, we expect the model to conceive the embeddings for words better and construct more general representation. Experiments determined that the multi-task learning works best at $\lambda = 0.8$ for both ACE05 and ERE datasets (Table 1). It obtains improvement on both the out-of-domain evaluation on ACE and in-domain evaluation on ERE. It works especially well on weblogs (wl) and telephone conversation (cts) domains on ACE, which possibly benefits from the discussion forum data from ERE.

On the other hand, we use the adversarial training between the two datasets to further enforce the representation to be close to each other. There is strong dependency between the schemas of these two datasets. Two examples from different datasets could have the same semantics in terms of relation type. We try to force the representation of these examples to be similar. With appropriate amount of this regularization ($\beta = 0.001$), the model can be further improved (Table 1). The amount of improvement is modest compared to sharing the encoder. This may show that the

Training Data	100%				50%			
Method	**bc**	**wl**	**cts**	**avg**	**bc**	**wl**	**cts**	**avg**
(Nguyen and Grishman, 2016)	63.07	**56.47**	53.65	57.73	-	-	-	-
Supervised	61.82	55.68	55.15	57.55	56.81	50.49	50.10	52.47
Multi-task	**63.59**	56.11	**56.78**	**58.83**	**58.24**	**52.90**	**53.09**	**54.37**

Table 2: Multi-task Learning with extra features on ACE05.

multi-task model can already balance representation with enough labels on both sides. We also artificially remove half of the training data of each dataset to see the performance in a relatively low-resource setting (row "Training Data" Table 1). We observe larger improvement with both multi-task learning and regularization. Because of the decrease of the training data, the best λ is 0.9 for ACE05 and 0.7 for ERE. We also use slightly stronger regularization ($\beta = 0.01$).

5.4 More Features on ACE05

Since ACE05 has been studied for a long time, numerous features have been found to be effective on this dataset. (Nguyen and Grishman, 2016) incorporated some of those features into the neural net and beat the state-of-art on the dataset. Although representation learning from scratch could be more general across multiple datasets, we compare the effect of multi-task learning with extra features on this specific dataset.

We add chunk embedding and on_dep_path embedding (Nguyen and Grishman, 2016). Similar to entity type embedding, chunk embedding is created according to each token's chunk type, we set the embedding size to 50. On_dep_path embedding is a vector indicating whether the token is on the dependency path between the two entities. In the multi-task model, the shared encoder is a bidirectional RNN (BiRNN) without attention (Equation (1-3)). These two embeddings will be concatenated to the output of the BiRNN to obtain the new h_i and then passed to Equation (4).

As the results (Table 2), our supervised baseline is slightly worse than the previous state-of-the-art neural model with extra features, but the multi-task learning can consistently help. The improvement is more obvious with 50% training data. It is also worth to note that with 50% training data, the extra features improve the supervised base model, but not the multi-task learning model. It shows the effectiveness of the multi-task model when there is less training data.

6 Conclusion and Future Work

We attempt to learn unified representation for relations by multi-task learning between ACE05 and ERE datasets. We use a shared encoder to learn the unified feature representation and then apply regularization by adversarial training. The improvement on both datasets shows the promising future of learning representation for relations in this unified way. This will require less training data for new relation schemas. It will be interesting future work to further explore the multi-task learning between different datasets, especially to capture the dependency between different schemas in the decoder.

Acknowledgments

This work was supported by DARPA/I2O Contract No. W911NF-18-C-0003 under the World Modelers program. The views, opinions, and/or findings contained in this article are those of the author and should not be interpreted as representing the official views or policies, either expressed or implied, of the Department of Defense or the U.S. Government. This document does not contain technology or technical data controlled under either the U.S. International Traffic in Arms Regulations or the U.S. Export Administration Regulations.

References

Jacqueline Aguilar, Charley Beller, Paul McNamee, Benjamin Van Durme, Stephanie Strassel, Zhiyi Song, and Joe Ellis. 2014. A comparison of the events and relations across ace, ere, tac-kbp, and framenet annotation standards. In *Proceedings of the Second Workshop on EVENTS: Definition, Detection, Coreference, and Representation (pp. 45-53).*

Razvan C. Bunescu and Raymond J. Mooney. 2005a. A shortest path dependency kernel for relation extraction. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, pp. 724-731. Association for Computational Linguistics.*

Razvan C. Bunescu and Raymond J. Mooney. 2005b. Subsequence kernels for relation extraction. In *Advances in Neural Information Processing Systems, pp. 171-178*.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoderdecoder for statistical machine translation. In *Proceedings of EMNLP*.

Long Duong, Trevor Cohn, Steven Bird, and Paul Cook. 2015. Low resource dependency parsing: Cross-lingual parameter sharing in a neural network parser. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), vol. 2, pp. 845-850*.

Lisheng Fu, Thien Huu Nguyen, Bonan Min, and Ralph Grishman. 2017. Domain adaptation for relation extraction with domain adversarial neural network. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers), vol. 2, pp. 425-429*.

Yaroslav Ganin and Victor Lempitsky. 2015. Unsupervised domain adaptation by backpropagation. In *Proceedings of ICML*.

Matthew R. Gormley, Mo Yu, and Mark Dredze. 2015. Improved relation extraction with feature-rich compositional embedding models. In *Proceedings of EMNLP*.

Jiang Jing and ChengXiang Zhai. 2007. Instance weighting for domain adaptation in nlp. In *Proceedings of ACL*.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *ICLR*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Proceedings of ICLR*.

Bonan Min, Zhuolin Jiang, Marjorie Freedman, and Ralph Weischedel. 2017. Learning transferable representation for bilingual relation extraction via convolutional neural networks. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers), vol. 1, pp. 674-684*.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using lstms on sequences and tree structures. In *Proceedings of ACL*.

Thien Huu Nguyen and Ralph Grishman. 2015. Relation extraction: Perspective from convolutional neural networks. In *Proceedings of the 1st NAACL Workshop on Vector Space Modeling for NLP (VSM)*.

Thien Huu Nguyen and Ralph Grishman. 2016. Combining neural networks and log-linear models to improve relation extraction. In *Proceedings of IJCAI Workshop on Deep Learning for Artificial Intelligence (DLAI)*.

Barbara Plank and Alessandro Moschitti. 2013. Embedding semantic similarity in tree kernels for domain adaptation of relation extraction. In *Proceedings of ACL*.

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.

Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of EMNLP*.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *Proceedings of COLING*.

Shu Zhang, Dequan Zheng, Xinchen Hu, and Ming Yang. 2015. Bidirectional long short-term memory networks for relation classification. In *Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation*.

GuoDong Zhou, Jian Su, Jie Zhang, and Min Zhang. 2005. Exploring various knowledge in relation extraction. In *Proceedings ofACL*.

Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of ACL*.