Supplementary Material for ”Attention-Based Capsule Network with Dynamic Routing for Relation Extraction”

0.1 Appendices

0.1.1 Single entity pair and multiple entity pairs relation extraction

As the Figure 1 depicts, for single entity pair relation extraction the object is a tuple of two named entities. Each mention of this tuple in text generates a different instance.

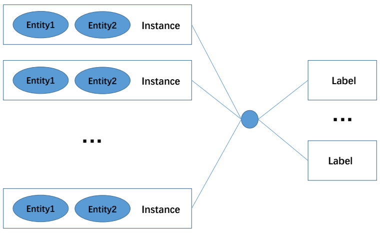


Figure 1: Multi-instance multi-label learning for single entity pair in one instance.

As the Figure 2 depicts, for multiple entity pairs relation extraction the object is an aggregation of tuples of two named entities. In order to simplify the calculation, we limit the maximum number of tuples to two, which means there are at most four entities in one instance(three entities in one instance is possible for the case when two tuples have a common entity). Each mention both containing these two tuples in text generates a different instance. Each relation has a relation mention which is not involved in training.

0.1.2 Data preprocessing

We excluded sentences longer than $L = 120$ and randomly split data for entity pairs with more than 500 mentions. For the NYT dataset, we filtered out 53 relations with more mentions. For the UW dataset, as the test set of the UW dataset contained only 200 sentences, we adopted a subset of the test set from the NYT dataset: all entity pairs with the corresponding four relations in UW and another 1500 randomly selected NA pairs. For the Wikidata dataset, we reconstructed the data from (Sorokin and Gurevych, 2017) and filtered out the sentences with multiple entities(up to four) labeled with mutiple relations(up to two). We have submitted the three datasets in the archive.

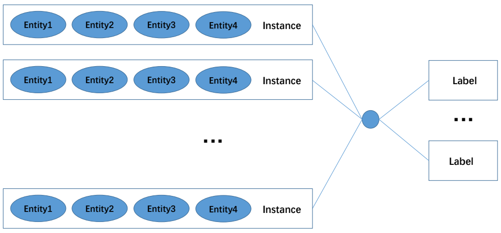


Figure 2: Multi-instance multi-label learning for multiple entity pairs in one instance.

Table 1: Train Dataset statistics (#Rel includes NA).

Dataset	#Rel	#Ent-Pair	#Mention	Sen-len
NYT	53	290429	577434	120
UW	5	132419	546731	120
Wikidata	339	284556	215895	120

0.1.3 Model parameters

We tuned the hyperparameters of our model using grid search where we set the learning rate amongst $\{0.1,\ 0.01,\ 0.001\}$ and the batch size among $\{64{,}128{,}256\}$. We also tried different types and dimensions of pretrained word embeddings, and we found that different datasets performed better when using different word embeddings .

Table 2 shows the parameter settings. We set some parameters empirically, such as the batch size, the word dimension, the number of epochs. We select 300 LSTM’s units based

Table 2: Parameter settings.

Number of epochs	3
Learning rate	0.001
Dropout probability	0.5
Batch size	128
Word dimension(NYT)	50
Word dimension(UW)	200
Word dimension(Wikidata)	200
Routing iteration	3
Primary capsule dimension	8
C	32
Maximum length of sentence	120
LSTMs' unit size	300

on our empirically parameter study from the set $\{250, 300, 350, 400, 450\}$. We utilized the word embeddings released by (Lin et al., 2016) for NYT dataset experiment. We trained word embeddings of $d_w = 200$ using Glove (Pennington et al., 2014) on the New York Times Corpus for UW dataset experiment. We trained word embeddings of $d_w = 200$ using Glove (Pennington et al., 2014) on the Wikipedia Corpus for Wikidata dataset experiment.

0.1.4 Feature templates

For single entity pair relation extraction, the input sample of one sentence is similar to: $[word-id_1, relative1-distance_1, relative2-distance_1], [word-id_2 relative1-distance_2, relative2-distance_2], ..., [word-id_L, relative1-distance_L, relative2-distance_L]$

For example,

[13,-4,-6],[145,-3,-5],...,[132,115,113]

For multiple entity pair relation extraction, the input sample of one sentence is similar to:

$[word-id_1, relative1-distance_1, relative2-distance_1, relative3-distance_1, relative4-distance_1], [word-id_2, relative1-distance_2, relative2-distance_2, relative3-distance_2, relative4-distance_2], ..., [word-id_L, relative1-distance_L, relative2-distance_L, relative3-distance_L, relative4-distance_L]$

For example,

[123,-4,-6,-2,2147483529],[115,-3,-5,2147483529],...,[1652,115,113,2147483529]

The relative distances are converted to positive integers in the experiment for better embedding through $rel-distance_i = rel-distance_i + 119$. The missing relative distances are set to a very large integer 147483529 (just a very big relative distance to to identify that the entity is not in this sentence).

0.1.5 Details of Primary Capsule Layer

The dimension change in the primary capsule layer (Tensorflow, for example) is given as follows:

Bi-LSTMs with Attention Layer Output:

$(batch, 120, 600, 1)$

Primary Capsule Layer Output:

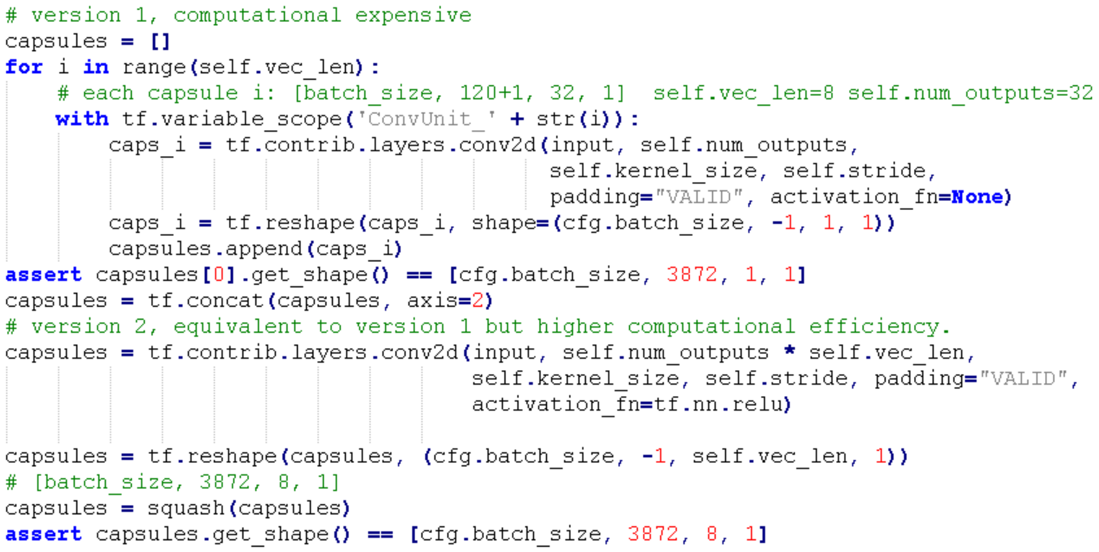
$(batch, (120+1)*32 = 3872, 8, 1)$ The Figure 3 below shows two versions of the tensorflow code for the primary capsule layer.

0.1.6 Additional Experimental Results

UW dataset: The total number of relations in our experiments on the UW dataset is $E = 5$ (including NA). We train word embeddings of $d_w = 200$ using Glove (Pennington et al., 2014) on the New York Times Corpus. The precision-recall curves for different models on the test set are shown in Figure 4. Our model BiLSTM+Capsule achieves comparable results compared with all baselines, where PCNN+ATT refers to (Lin et al., 2016) , DMN refers to (Feng et al., 2017) and Rank+ExATT refers to (Ye et al., 2017). We also show the precision numbers for some particular recalls as well as the AUC in Table 3, where our model achieves comparable results with PCNN+ATT.

Table 3: Precisions on the UW dataset.

Recall	0.1	0.2	0.3	0.4	AUC
PCNN+ATT	0.800	0.698	0.669	0.616	0.599
Memory	0.825	0.747	0.684	0.637	0.698
Rank+ExATT	0.826	0.768	0.718	0.677	0.704
Our Model	0.817	0.764	0.675	0.584	0.618



```
# version 1, computational expensive
capsules = []
for i in range(self.vec_len):
    # each capsule i: [batch_size, 120+1, 32, 1]  self.vec_len=8 self.num_outputs=32
    with tf.variable_scope('ConvUnit_' + str(i)):
        caps_i = tf.contrib.layers.conv2d(input, self.num_outputs,
                                          self.kernel_size, self.stride,
                                          padding="VALID", activation_fn=None)
        caps_i = tf.reshape(caps_i, shape=(cfg.batch_size, -1, 1, 1))
        capsules.append(caps_i)
assert capsules[0].get_shape() == [cfg.batch_size, 3872, 1, 1]
capsules = tf.concat(capsules, axis=2)
# version 2, equivalent to version 1 but higher computational efficiency.
capsules = tf.contrib.layers.conv2d(input, self.num_outputs * self.vec_len,
                                    self.kernel_size, self.stride, padding="VALID",
                                    activation_fn=tf.nn.relu)

capsules = tf.reshape(capsules, (cfg.batch_size, -1, self.vec_len, 1))
# [batch_size, 3872, 8, 1]
capsules = squash(capsules)
assert capsules.get_shape() == [cfg.batch_size, 3872, 8, 1]
```

Figure 3: Two versions of tensorflow code for primary capsule layer

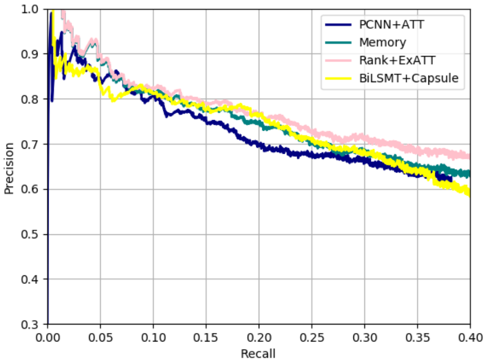


Figure 4: PR curves for the UW dataset

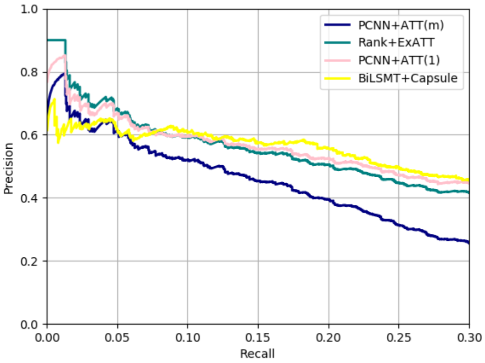


Figure 5: PR curves for the Wikidata dataset

Wikidata dataset: The precision-recall curves for different models on the test set are shown in Figures 5. Since the precision drops significantly with large recalls, we emphasize a part of the curve with recall number smaller than 0.3. Our model BiLSTM+Capsule achieves comparable results compared with all baselines, where PCNN+ATT(1) refers to train sentences with two entities and one relation label, PCNN+ATT(m) refers to train sentences with four entities[1] and two relation labels, Rank+ExATT refers to (Ye et al., 2017).

References

Xiaocheng Feng, Jiang Guo, Bing Qin, Ting Liu, and Yongjie Liu. 2017. Effective deep memory networks for distant supervised relation extraction. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI*, pages 19–25.

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 (Volume 1: Long Papers)*, volume 1, pages 2124–2133.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 confer-*

---

[1] Two additional position embeddings.

ence on empirical methods in natural language processing (EMNLP), pages 1532–1543.

Daniil Sorokin and Iryna Gurevych. 2017. Context-aware representations for knowledge base relation extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1784–1789.

Hai Ye, Wenhan Chao, Zhunchen Luo, and Zhoujun Li. 2017. Jointly extracting relations with class ties via effective deep ranking. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1810–1820.