we experiment on the four top-level classes in this corpus as in previous work (Rutherford and Xue, 2015). We extract all the implicit relations of PDTB, and follow the setup of (Rutherford and Xue, 2015). We split the data into a training set (Sections 2-20), development set (Sections 0-1), and test set (Section 21-22). Table 1 summarizes the statistics of the four PDTB discourse relations, i.e., Comparison, Contingency, Expansion and Temporal.

Relation	Train	Dev	Test
Comparison	1855	189	145
Contingency	3235	281	273
Expansion	6673	638	538
Temporal	582	48	55
Total	12345	1156	1011

Table 1: Statistics of Implicit Discourse Relations in PDTB.

We first convert the tokens in PDTB to lowercase. The word embeddings used for initializing the word representations are provided by GloVe (Pennington et al., 2014), and the dimension of the embeddings D_e is 50. The hyper-parameters, including the momentum δ, the two learning rates λ and λ_e, the dropout rate q, the dimension of LSTM output vector d, the dimension of memory vector d_m are all set according to the performance on the development set Due to space limitation, we do not present the details of tuning the hyper-parameters and only give their final settings as shown in Table 2.

δ	λ	λ_e	q	d	d_m
0.9	0.01	0.002	0.1	50	200

Table 2: Hyper-parameters for Neural Network with Multi-Level Attention.

To evaluate our model, we adopt two kinds of experiment settings. The first one is the four-way classification task, and the second one is the binary classification task, where we build a one-vs-other classifier for each class. For the second setting, to solve the problem of unbalanced classes in the training data, we follow the reweighting method of (Rutherford and Xue, 2015) to reweigh the training instances according to the size of each relation class. We also use visualization methods to analyze how multi-level attention helps our model.

3.2 Results

First, we design experiments to evaluate the effectiveness of attention levels and how many attention levels are appropriate. To this end, we implement a baseline model (LSTM with no attention) which directly applies the mean pooling operation over LSTM output vectors of two arguments without any attention mechanism. Then we consider different attention levels including one-level, two-level and three-level. The detailed results are shown in Table 3. For four-way classification, macro-averaged $F1$ and Accuracy are used as evaluation metrics. For binary classification, $F1$ is adopted to evaluate the performance on each class.

System	Four-way		Binary			
	$F1$	Acc.	Comp.	Cont.	Expa.	Temp.
LSTM	39.40	54.50	33.72	44.79	68.74	33.14
NNMA (one-level)	43.48	55.59	34.72	49.47	68.52	36.70
NNMA (two-level)	**46.29**	57.17	36.70	**54.48**	**70.43**	**38.84**
NNMA (three-level)	44.95	**57.57**	**39.86**	53.69	69.71	37.61

Table 3: Performances of NNMA with Different Attention Levels.

From Table 3, we can see that the basic LSTM model performs the worst. With attention levels added, our NNMA model performs much better. This confirms the observation above that one-pass reading is not enough for identifying the discourse relations. With respect to the four-way F_1 measure, using NNMA with one-level attention produces a 4% improvement over the baseline system with no attention. Adding the second attention level gives another 2.8% improvement. We perform significance test for these two improvements, and they are both significant under one-tailed t-test ($p < 0.05$). However, when adding the third attention level, the performance does not promote much and almost reaches its plateau. We can see that three-level NNMA experiences a decease in F_1 and a slight increase in Accuracy compared to two-level NNMA. The results imply that with more attention levels considered, our model may perform slightly better, but it may incur the over-fitting problem due to adding more parameters. With respect to the binary classification F_1 measures, we can see