Coherence Modeling of Asynchronous Conversations: A Neural Entity Grid Approach

Tasnim Mohiuddin[*] and **Shafiq Joty**[*]
Nanyang Technological University
{mohi0004,srjoty}@ntu.edu.sg

Dat Tien Nguyen[*]
University of Amsterdam
t.d.nguyen@uva.nl

A Supplemental Material

A.1 Optimal Hyper-parameter Settings for Discrimination Task in Monologue

Hyperparameters	Lex. Neural Grid
Minibatch size	10
Max length	25000
Embedding Size	300
#Filters	150
Conv. filter length	10
Conv. stride	1
Conv. Padding	VALID
Pooling filter length	10
Pooling Stride	10

Table 1: Optimal hyper-parameter settings for Lexicalized Neural Grid model on WSJ dataset.

A.2 Optimal Hyper-parameter Settings for Discrimination Task in Conversation

Hyperparameters	Temporal	Path-level	Tree-level
Minibatch size	10	10	10
Max length	22,000	2,500	2,500
Max branch	NA	NA	20
Embedding Size	300	300	300
#Filters	150	150	150
Conv. filter length	15	15	15
Conv. filter width	NA	NA	1
Conv. stride	1	1	1
Conv. Padding	VALID	VALID	VALID
Pooling filter length	15	15	15
Pooling filter width	NA	NA	1
Pooling Stride	15	15	15

Table 2: Optimal hyper-parameter settings for Temporal, Path-level, and Tree-level models on CNET dataset.

[*]All authors contributed equally.

A.3 Optimal Hyper-parameter Settings for Thread Reconstruction

Hyperparameters	Thread Reconstruction Model
Minibatch size	16
Max length	2500
Embedding Size	300 (google)
#Filters	150
Conv. filter length	11
Conv. filter width	2
Conv. stride	1
Conv. Padding	VALID
Pooling filter length	11
Pooling filter width	2
Pooling Stride	10

Table 3: Optimal hyper-parameter settings for Thread Reconstruction Model on CNET dataset.

A.4 Complete Results on WSJ

Table 4 shows the complete results in terms of accuracy and F_1 scores on the standard discrimination task and the inverse ordering task.

A.5 Complete Results on CNET

Table 5 shows the complete results in terms of accuracy and F_1 scores on the standard discrimination task and the inverse ordering task.

A.6 Example: Why Aggregating Path-level Decisions as Opposed to Path-level Scores

Consider a conversation with three paths to which a model assigns 20, 10, and 5 (total 35), and the corresponding incoherent paths get 5, 15, and 10 (total 30). Aggregating scores would favor the model, although it makes wrong decisions for two out of three.

A.7 Using Glove as Pretrined Embedding

Alongside with google's word2vec, we also tried Glove as pretrained embedding which gave 86%

	Model	Emb.	Standard Acc	Standard F_1	Inverse Acc	Inverse F_1
I	Grid (E&C)	-	81.58	81.60	75.78	75.78
	Ext. Grid (E&C)	-	84.95	84.95	80.34	80.34
II	Neural Grid (N&J)	Random	84.36	84.36	83.94	83.94
	Ext. Neural Grid (N&J)	Random	86.93	86.93	83.00	83.00
III	Lex. Neural Grid	Random	87.03†	87.03†	86.88†	86.88†
	Lex. Neural Grid	Google	**88.56**†	**88.56**†	**88.23**†	**88.23**†

Table 4: Discrimination results on the **WSJ** dataset. Superscript † indicates a lexicalized model is significantly superior to the unlexicalized Neural Grid (N&J) model with p-value < 0.01.

Conv. Rep	Model	Emb.	Standard Acc	Standard F_1	Inverse Acc	Inverse F_1
Temporal	Neural Grid (N&J)	random	82.28	82.28	70.53	70.53
	Lex. Neural Grid	random	86.63	86.63	80.40	80.40
	Lex. Neural Grid	Google	87.17	87.17	80.76	80.76
Path-level	Neural Grid (N&J)	random	81.47	82.39	71.60†	75.68†
	Lex. Neural Grid	random	86.13	88.13	85.73†	88.38†
	Lex. Neural Grid	Google	86.67	88.44	87.20†	89.31†
Tree-level	Neural Grid (N&J)	random	83.98†	83.98†	77.33†	77.33†
	Lex. Neural Grid	random	89.87†	89.87†	89.23†	89.23†
	Lex. Neural Grid	Google	**91.29**†	**91.29**†	**90.40**†	**90.40**†

Table 5: Discrimination results on **CNET**. Superscript † indicates a model is significantly superior to its temporal counterpart with p-value < 0.01.

accuracy on WSJ dataset. In the same model with same hyperparameter settings, by using word2vec as pretrained embedding, we achieved 88.56% accuracy.

A.8 Multifilter in Convolutional Layer

We tried with different filter shapes in the convolutional layer. We also tried by incorporating multiple concurrent filters with different shapes in this layer. But in our cases, the addition of multiple concurrent filters did not give any extra benefit.

A.9 Average Pooling in Addition to Max Pooling

In addition to max pooling, we added average pooling to our model to capture more subtle features. But this addition of average pooling did not improve the accuracy.

A.10 Dynamic Margin in Loss Calculation

In computing *pairwise ranking* loss, Nguyen and Joty (2017) used a constant value (C=1) as margin. We tried with dynamic margin $C_{i,j}$ instead of a constant value. For each pair, we computed the difference between the positive and negative examples by counting the number of mismatched orders. This difference worked as margin in the dynamic case. We also tried with several constant values as margin. For CNET dataset, constant value 6 as margin produced the best result.

A.11 Tree-level Model: Another Approach

In this approach, we wanted to represent the tree in such a way that the entities in different branches but in same depth level remain close to each other. The problem with this representation is that - individual path of the tree cannot be captured. As a result, the performance of this approach was not up to the mark.