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

Abstract

We propose a novel coherence model for written asynchronous conversations (e.g., forums, emails), and show its applications in coherence assessment and thread reconstruction tasks. We conduct our research in two steps. First, we propose improvements to the recently proposed neural entity grid model by lexicalizing its entity transitions. Then, we extend the model to asynchronous conversations by incorporating the underlying conversational structure in the entity grid representation and feature computation. Our model achieves state of the art results on standard coherence assessment tasks in monologue and conversations outperforming existing models. We also demonstrate its effectiveness in reconstructing thread structures.

1 Introduction

Sentences in a text or a conversation do not occur independently, rather they are connected to form a coherent discourse that is easy to comprehend. **Coherence models** are computational models that can distinguish a coherent discourse from incoherent ones. It has ranges of applications in text generation, summarization, and coherence scoring.

s_0: **LDI** Corp., Cleveland, said it will offer $50 million in commercial **paper** backed by leaserental receivables.

s_1: The program matches funds raised from the sale of the commercial **paper** with small to medium-sized leases.

s_2: **LDI** termed the **paper** “non-recourse financing”, meaning that investors would be repaid from the lease receivables, rather than directly by LDI Corp.

s_3: **LDI** leases and sells data-processing, telecommunications and other high-tech equipment.

	INVESTORS	MILLION	FUNDS	EQUIPMENT	CORP.	**PAPER**	SALE	TELECOMM.	LEASE	PROGRAM	CLEVELAND	RECEIVABLES	LEASES	DATA-PROCESS.	**LDI**	NON-RECOURSE
s_0	–	O	–	–	S	X	–	–	–	–	X	X	–	–	X	–
s_1	–	–	O	–	–	X	X	–	–	S	–	–	X	–	–	–
s_2	S	–	–	–	X	S	–	–	X	–	–	X	–	–	S	X
s_3	–	–	–	O	–	–	–	X	–	–	–	–	–	X	S	–

Table 1: Entity grid representation (bottom) for a document (top) from the WSJ corpus.

Inspired by formal theories of discourse, a number of coherence models have been proposed (Barzilay and Lapata, 2008; Lin et al., 2011; Li and Jurafsky, 2017). The **entity grid** model (Barzilay and Lapata, 2008) is one of the most popular coherence models that has received much attention over the years. As exemplified in Table 1, the model represents a text by a grid that captures how grammatical roles of different discourse entities (*e.g.,* nouns) change from one sentence to another in the text. The grid is then converted into a feature vector containing probabilities of local entity transitions, enabling machine learning models to measure the degree of coherence. Earlier extensions of this basic model incorporate entity-specific features (Elsner and Charniak, 2011b), multiple ranks (Feng and Hirst, 2012), and coherence relations (Feng et al., 2014).

Recently, Nguyen and Joty (2017) proposed a neural version of the grid models. Their model first transforms the grammatical roles in a grid into their distributed representations, and employs a convolution operation over it to model entity transitions in the distributed space. The spatially max-pooled features from the convoluted features are used for coherence scoring. This model achieves state-of-the-art results in standard evaluation tasks on the Wall Street Journal (WSJ) corpus.

Although the neural grid model effectively captures long entity transitions, it is still limited in that it does not consider any lexical information regarding the entities, thereby, fails to distinguish

[*] All authors contributed equally.

between entity types. Although the extended neural grid considers entity features like named entity and proper mention, it requires an explicit feature extraction step, which can prevent us to transfer the model to a resource-poor language or domain.

Apart from these limitations, previous research on coherence models has mainly focused on monologic discourse (*e.g.,* news article). The only exception is the work of Elsner and Charniak (2011a), who applied coherence models to the task of conversation disentanglement in **synchronous** conversations like phone and chat conversations.

With the emergence of Internet technologies, **asynchronous** communication media like emails, blogs, and forums have become a commonplace for discussing events and issues, seeking answers, and sharing personal experiences. Participants in these media interact with each other asynchronously, by writing at different times. We believe coherence models for asynchronous conversations can help many downstream applications in these domains. For example, we will demonstrate later that coherence models can be used to predict the underlying thread structure of a conversation, which provides crucial information for building effective conversation summarization systems (Carenini et al., 2008) and community question answering systems (Barron-Cedeno et al., 2015).

To the best of our knowledge, none has studied the problem of coherence modeling in asynchronous conversation before. Because of its asynchronous nature, information flow in these conversations is often not sequential as in monologue or synchronous conversation. This poses a novel set of challenges for discourse analysis models (Joty et al., 2013; Louis and Cohen, 2015). For example, consider the forum conversation in Figure 2(a). It is not obvious how a coherence model like the entity grid can represent the conversation, and use it in downstream tasks effectively.

In this paper we aim to remedy the above limitations of existing models in two steps. First, we propose improvements to the existing neural grid model by *lexicalizing* its entity transitions. We propose methods based on word embeddings to achieve better generalization with the lexicalized model. Second, we adapt the model to asynchronous conversations by incorporating the underlying *conversational structure* in the grid representation and subsequently in feature computation. For this, we propose a novel grid representation for asynchronous conversations, and adapt the convolution layer of the neural model accordingly.

We evaluate our approach on two discrimination tasks. The first task is the standard one, where we assess the models based on their performance in discriminating an original document from its random permutation. In our second task, we ask the models to distinguish an original document from its inverse order of the sentences. For our adapted model to asynchronous conversation, we also evaluate it on *thread reconstruction*, a task specific to asynchronous conversation. We performed a series of experiments, and our main findings are:

(a) Our experiments on the WSJ corpus validate the utility of our proposed extension to the existing neural grid model, yielding absolute F_1 improvements of up to 4.2% in the standard task and up to 5.2% in the inverse-order discrimination task, setting a new state-of-the-art.

(b) Our experiments on a forum dataset show that our adapted model that considers the conversational structure outperforms the temporal baseline by more than 4% F_1 in the standard task and by about 10% F_1 in the inverse order discrimination task.

(c) When applied to the thread reconstruction task, our model achieves promising results outperforming several strong baselines.

We have released our source code and datasets at https://ntunlpsg.github.io/project/coherence/n-coh-acl18/

2 Background

In this section we give an overview of existing coherence models. In the interest of coherence, we defer description of the neural grid model (Nguyen and Joty, 2017) until next section, where we present our extension to this model.

2.1 Traditional Entity Grid Models

Introduced by Barzilay and Lapata (2008), the **entity grid** model represents a text by a two-dimensional matrix. As shown in Table 1, the rows correspond to sentences, and the columns correspond to entities (noun phrases). Each entry $E_{i,j}$ represents the syntactic role that entity e_j plays in sentence s_i, which can be one of: subject (S), object (O), other (X), or absent (–). In cases where an entity appears more than once with different grammatical roles in the same sentence, the role with the highest rank (S $\succ$ O $\succ$ X) is considered.

Motivated by the Centering Theory (Grosz et al., 1995), the model considers **local entity transitions** as the deciding patterns for assessing coherence. A local entity transition of length k is a sequence of $\{$S,O,X,–$\}^k$, representing grammatical roles played by an entity in k consecutive sentences. Each grid is represented by a vector of 4^k transition probabilities computed from the grid. To distinguish between transitions of important entities from unimportant ones, the model considers the *salience* of the entities, which is measured by their occurrence frequency in the document. With the feature vector representation, coherence assessment task is formulated as a ranking problem in a SVM preference ranking framework (Joachims, 2002). Barzilay and Lapata (2008) showed significant improvements in two out of three evaluation tasks when a coreference resolver is used to identify coreferent entities in a text.

Elsner and Charniak (2011b) show improvements to the grid model by including non-head nouns as entities. Instead of employing a coreference resolver, they match the nouns to detect coreferent entities. They demonstrate further improvements by extending the grid to distinguish between entities of different types. They do so by incorporating entity-specific features like named entity, noun class and modifiers. Lin et al. (2011) model transitions of discourse roles for entities as opposed to their grammatical roles. They instantiate discourse roles by discourse relations in Penn Discourse Treebank (Prasad et al., 2008). In a follow up work, Feng et al. (2014) trained the same model but using relations derived from deep discourse structures annotated with Rhetorical Structure Theory (Mann and Thompson, 1988).

2.2 Other Existing Models

Guinaudeau and Strube (2013) proposed a **graph-based** unsupervised method. They convert an entity grid into a bipartite graph consisting of two sets of nodes, representing sentences and entities, respectively. The edges are assigned weights based on the grammatical role of the entities in the respective sentences. They perform one-mode projections to transform the bipartite graph to a directed graph containing only sentence nodes. The coherence score of the document is then computed as the average *out-degree* of sentence nodes.

Louis and Nenkova (2012) introduced a coherence model based on **syntactic patterns** by assuming that sentences in a coherent text exhibit certain syntactic regularities. They propose a local coherence model that captures the co-occurrence of structural features in adjacent sentences, and a global model based on a hidden Markov model, which learns the global syntactic patterns from clusters of sentences with similar syntax.

Li and Hovy (2014) proposed a **neural** framework to compute the coherence score of a document by estimating coherence probability for every window of three sentences. They encode each sentence in the window using either a recurrent or a recursive neural network. To get a document-level coherence score, they sum up the window-level log probabilities. Li and Jurafsky (2017) proposed two encoder-decoder models augmented with latent variables for both coherence evaluation and discourse generation. Their first model incorporates global discourse information (topics) by feeding the output of a sentence-level HMM-LDA model (Gruber et al., 2007) into the encoder-decoder model. Their second model is trained end-to-end with variational inference.

In our work, we take an entity-based approach, and extend the neural grid model proposed recently by Nguyen and Joty (2017).

3 Extending Neural Entity Grid

In this section we first briefly describe the neural entity grid model proposed by Nguyen and Joty (2017). Then, we propose our extension to this model that leads to improved performance. We present our coherence model for asynchronous conversation in the next section.

3.1 Neural Entity Grid

Figure 1 depicts the neural grid model of Nguyen and Joty (2017). Given an entity grid E, they first transform each entry $E_{i,j}$ (a grammatical role) into a distributed representation of d dimensions by looking up a shared embedding matrix $M \in \mathbb{R}^{|G| \times d}$, where G is the vocabulary of possible grammatical roles, *i.e.,* $G = \{S, O, X, -\}$. Formally, the look-up operation can be expressed as:

$$L = \Big[M(E_{1,1}) \cdots M(E_{i,j}) \cdots M(E_{I,J}) \Big] \quad (1)$$

where $M(E_{i,j})$ refers to the row in M that corresponds to grammatical role $E_{i,j}$, and I and J are

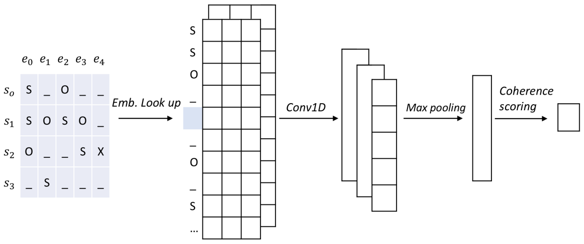


Figure 1: Neural entity grid model proposed by Nguyen and Joty (2017). The model is trained using a pairwise ranking approach with shared parameters for positive and negative documents.

the number of rows (sentences) and columns (entities) in the entity grid, respectively. The result of the look-up operation is a tensor $L \in \mathbb{R}^{I \times J \times d}$, which is fed to a convolution layer to model local entity transitions in the distributed space.

The convolution layer of the neural network composes patches of entity transitions into high-level abstract features by treating entities independently (*i.e.,* 1D convolution). Formally, it applies a *filter* $\mathbf{w} \in \mathbb{R}^{m.d}$ to each local entity transition of length m to generate a new abstract feature z_i:

$$z_i = h(\mathbf{w}^T L_{i:i+m,j} + b_i) \tag{2}$$

where $L_{i:i+m,j}$ denotes concatenation of m vectors in L for entity e_j, b_i is a bias term, and h is a nonlinear activation function. Repeated application of this filter to each possible m-length transitions of different entities in the grid generates a *feature map*, $\mathbf{z}^i = [z_1, \cdots, z_{I.J+m-1}]$. This process is repeated N times with N different filters to get N different feature maps, $[\mathbf{z}^1, \cdots, \mathbf{z}^N]$. A *max-pooling* operation is then applied to extract the most salient features from each feature map:

$$\mathbf{p} = [\mu_l(\mathbf{z}^1), \cdots, \mu_l(\mathbf{z}^N)] \tag{3}$$

where $\mu_l(\mathbf{z}^i)$ refers to the max operation applied to each non-overlapping window of l features in the feature map $\mathbf{z}^i$. Finally, the pooled features are used in a linear layer to produce a *coherence score*:

$$y = \mathbf{u}^T \mathbf{p} + b \tag{4}$$

where $\mathbf{u}$ is the weight vector and b is a bias term. The model is trained with a *pairwise ranking* loss based on ordered training pairs (E_i, E_j):

$$\mathcal{L}(\theta) = \max\{0, 1 - \phi(E_i|\theta) + \phi(E_j|\theta)\} \tag{5}$$

where entity grid E_i exhibits a higher degree of coherence than grid E_j, and $y = \phi(E_k|\theta)$ denotes the transformation of input grid E_k to a coherence score y done by the model with parameters θ. We will see later that such ordering of documents (grids) can be obtained automatically by permuting the original document. Notice that the network shares its parameters (θ) between the positive (E_i) and the negative (E_j) instances in a pair.

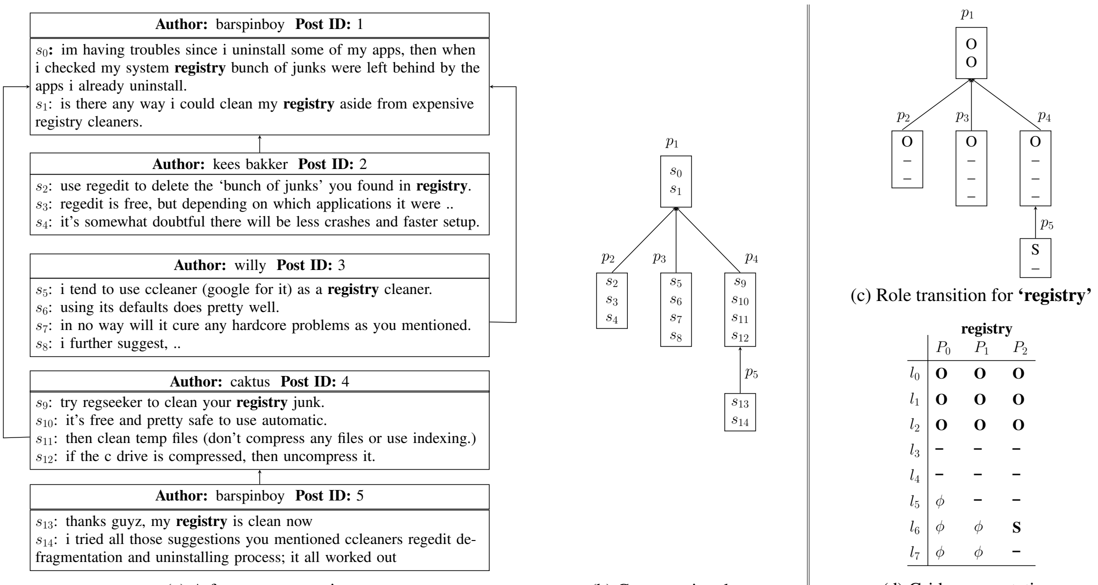
Since entity transitions in the convolution step are modeled in a continuous space, it can effectively capture longer transitions compared to traditional grid models. Unlike traditional grid models that compute transition probabilities from a *single* grid, convolution filters and role embeddings in the neural model are learned from all training instances, which helps the model to generalize well.

Since the abstract features in the feature maps are generated by convolving over role transitions of different entities in a document, the model implicitly considers relations between entities in a document, whereas transition probabilities in traditional entity grid models are computed without considering any such relation between entities. Convolution over the entire grid also incorporates *global* information (*e.g.,* topic) of a discourse.

3.2 Lexicalized Neural Entity Grid

Despite its effectiveness, the neural grid model presented above has a limitation. It does not consider any lexical information regarding the entities, thus, cannot distinguish between transitions of different entities. Although the extended neural grid model proposed in (Nguyen and Joty, 2017) does incorporate entity features like named entity type and proper mention, it requires an explicit feature extraction step using tools like named entity recognizer. This can prevent us in transferring the model to resource-poor languages or domains.

To address this limitation, we propose to lexicalize entity transitions. This can be achieved by attaching the entity with the grammatical roles. For example, if an entity e_j appears as a subject (S) in sentence s_i, the grid entry $E_{i,j}$ will be encoded as e_j-S. This way, an entity OBAMA as subject (OBAMA-S) and as object (OBAMA-O) will have separate entries in the embedding matrix M. We can initialize the word-role embeddings randomly, or with pre-trained embeddings for the word (OBAMA). In another variation, we kept word and role embeddings separate and con-



Author: barspinboy **Post ID:** 1

s_0**:** im having troubles since i uninstall some of my apps, then when i checked my system **registry** bunch of junks were left behind by the apps i already uninstall.
s_1: is there any way i could clean my **registry** aside from expensive registry cleaners.

Author: kees bakker **Post ID:** 2

s_2: use regedit to delete the 'bunch of junks' you found in **registry**.
s_3: regedit is free, but depending on which applications it were ..
s_4: it's somewhat doubtful there will be less crashes and faster setup.

Author: willy **Post ID:** 3

s_5: i tend to use ccleaner (google for it) as a **registry** cleaner.
s_6: using its defaults does pretty well.
s_7: in no way will it cure any hardcore problems as you mentioned.
s_8: i further suggest, ..

Author: caktus **Post ID:** 4

s_9: try regseeker to clean your **registry** junk.
s_{10}: it's free and pretty safe to use automatic.
s_{11}: then clean temp files (don't compress any files or use indexing.)
s_{12}: if the c drive is compressed, then uncompress it.

Author: barspinboy **Post ID:** 5

s_{13}: thanks guyz, my **registry** is clean now
s_{14}: i tried all those suggestions you mentioned ccleaners regedit defragmentation and uninstalling process; it all worked out

(a) A forum conversation

(b) Conversational tree

(c) Role transition for **'registry'**

registry

	P_0	P_1	P_2
l_0	**O**	**O**	**O**
l_1	**O**	**O**	**O**
l_2	**O**	**O**	**O**
l_3	–	–	–
l_4	–	–	–
l_5	ϕ	–	–
l_6	ϕ	ϕ	**S**
l_7	ϕ	ϕ	–

(d) Grid representations

Figure 2: (a) A forum conversation, (b) Thread structure of the conversation, (c) Entity role transition over a conversation tree, and (d) 2D role transition matrix for an entity; ϕ denotes zero-padding.

catenated them after the look-up, thus enforcing OBAMA-S and OBAMA-O to share a part of their representations. However, in our experiments, we found the former approach to be more effective.

4 Coherence Models for Asynchronous Conversations

The main difference between monologue and asynchronous conversation is that information flow in asynchronous conversation is not sequential as in monologue, rather it is often interleaved. For example, consider the forum conversation in Figure 2(a). There are three possible subconversations, each corresponding to a path from the root node to a leaf node in the conversation graph in Figure 2(b). In response to seeking suggestions about how to clean *system registry*, the first path ($p_1 \leftarrow p_2$) suggests to use *regedit*, the second path ($p_1 \leftarrow p_3$) suggests *ccleaner*, and the third one ($p_1 \leftarrow p_4$) suggests using *regseeker*. These discussions are interleaved in the chronological order of the posts ($p_1 \leftarrow p_2 \leftarrow p_3 \leftarrow p_4 \leftarrow p_5$). Therefore, monologue-based coherence models may not be effective if applied directly to the conversation.

We hypothesize that coherence models for asynchronous conversation should incorporate the conversational structure like the tree structure in Figure 2(b), where the nodes represent posts and the edges represent 'reply-to' links between them. Since the grid models operate at the sentence level, we construct conversational structure at the sentence level. We do this by linking the boundary sentences across posts and by linking sentences in the same post chronologically. Specifically, we connect the first sentence of post p_j to the last sentence of post p_i if p_j replies to p_i, and sentence s_{t+1} is linked to s_t if both s_t and s_{t+1} are in the same post.[1] Now the question is, how can we represent a conversation tree with an entity grid, and then model entity transitions in the tree? In the following, we describe our approach to this problem.

4.1 Conversational Entity Grid

The conversation tree captures how topics flow in an asynchronous conversation. Our key hypothesis is that in a coherent conversation entities exhibit certain local patterns in the conversation tree in terms of their distribution and syntactic realization. Figure 2(c) shows how the grammatical roles of entity *'registry'* in our example conversation change over the tree. For coherence assessment, we wish to model entity transitions along each of the conversation paths (top-to-bottom), and also their spatial relations across the paths (left-to-right). The existing grid representation is insufficient to model the *two-dimensional (2D) spatial* entity transitions in a conversation tree.

We propose a three-dimensional (3D) grid for representing entity transitions in an asynchronous conversation. The first dimension in our grid rep-

[1] The links between sentences are not explicitly shown in Figure 2(b) to avoid visual clutter.

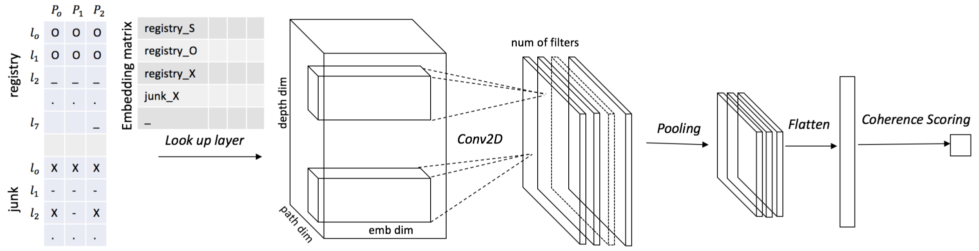


Figure 3: **Conversational Neural Grid** model for assessing coherence in asynchronous conversations.

resents *entities*, while the second and third dimensions represent *depth* and *path* of the tree, respectively. Figure 2(d) shows an example representation for an entity '*registry*'. Each column in the matrix represents transitions of the entity along a path, whereas each row represents transitions of the entity at a level of the conversation tree.

Although illustrated with a tree structure, our method is applicable to general graph-structured conversations, where a post can reply to multiple previous posts. Our model relies on paths from the root to the leaf nodes, which can be extracted for any graph as long as we avoid loops.

4.2 Modeling Entity Transitions

As shown in Figure 3, given a 3D entity grid as input, the look-up layer (Eq. 1) of our neural grid model produces a 4D tensor $L \in \mathbb{R}^{I \times J \times P \times d}$, where I is the total number of entities in the conversation, J is the depth of the tree, P is the number of paths in the tree, and d is the embedding dimension. The convolution layer then uses a 2D filter $\mathbf{w} \in \mathbb{R}^{m.n.d}$ to convolve local patches of entity transitions

$$z_i = h(\mathbf{w}^T L_{i,j:j+m,p:p+n} + b_i) \tag{6}$$

where m and n are the height and width of the filter, and $L_{i,j:j+m,p:p+n} \in \mathbb{R}^{m.n.d}$ denotes a concatenated vector containing $(m \times n)$ embeddings representing a 2D window of entity transitions. As we repeatedly apply the filter to each possible window with stride size 1, we get a 2D feature map Z^i of dimensions $(I.J+m-1) \times (I.P+n-1)$. Employing N different filters, we get N such 2D feature maps, $[Z^1, \cdots, Z^N]$, based on which the max pooling layer extracts the most salient features:

$$\mathbf{p} = [\mu_{l \times w}(Z^1), \cdots, \mu_{l \times w}(Z^N)] \tag{7}$$

where $\mu_{l \times w}$ refers to the $\max$ operation applied to each non-overlapping 2D window of $l \times w$ features in a feature map. The pooled features are then linearized and used for coherence scoring in the final layer of the network as described by Equation 4.

5 Experiments on Monologue

To validate our proposed extension to the neural grid model, we first evaluate our lexicalized neural grid model in the standard evaluation setting.

Evaluation Tasks and Dataset: We evaluate our models on the standard **discrimination** task (Barzilay and Lapata, 2008), where a coherence model is asked to distinguish an original document from its incoherent renderings generated by random permutations of its sentences. The model is considered correct if it ranks the original document higher than the permuted one.

We use the same train-test split of the WSJ dataset as used in (Nguyen and Joty, 2017) and other studies (Elsner and Charniak, 2011b; Feng et al., 2014). Following previous studies, we use 20 random permutations of each article for both training and testing, and exclude permutations that match the original article. Table 2 gives some statistics about the dataset along with the number of pairs used for training and testing. Nguyen and Joty (2017) randomly selected 10% of the training pairs for development purposes, which we also use for tuning hyperparameters in our models.

In addition to the standard setting, we also evaluate our models on an *inverse-order* setting, where we ask the models to distinguish an original document from the inverse order of its sentences (*i.e.,* from last to first). The transitions of roles in a negative grid are in the reverse order of the original grid. We do not train our models explicitly on this task, rather use the trained model from the standard setting. The number of test pairs in this setting is same as the number of test documents.

Model Settings and Training: We train the neural models with the pairwise ranking loss in Equation 5. For a fair comparison, we use

	Sections	# Doc.	Avg. # Sen.	# Pairs
Train	00-13	1,378	21.5	26,422
Test	14-24	1,053	22.3	20,411

Table 2: Statistics on the WSJ dataset.

similar model settings as in (Nguyen and Joty, 2017)[2] – ReLU as activation functions (h), RMSProp (Tieleman and Hinton, 2012) as the learning algorithm, Glorot-uniform (Glorot and Bengio, 2010) for initializing weight matrices, and uniform $\mathcal{U}(-0.01, 0.01)$ for initializing embeddings randomly. We applied batch normalization (Ioffe and Szegedy, 2015), which gave better results than using dropout. Minibatch size, embedding size and filter number were fixed to 32, 300 and 150, respectively. We tuned for optimal filter and pooling lengths in $\{2, \cdots, 12\}$. We train up to 25 epochs, and select the model that performs best on the development set; see **supplementary** documents for best hyperparameter settings for different models. We run each experiment five times, each time with a different random seed, and we report the average of the runs to avoid any randomness in results. Statistical significance tests are done using an *approximate randomization* test with SIGF V.2 (Padó, 2006).

Results and Discussions: We present our results on the standard discrimination task and the inverse-order task in Table 3; see Std (F_1) and Inv (F_1) columns, respectively. For space limitations, we only show F_1 scores here, and report both accuracy and F_1 in the supplementary document. We compare our lexicalized models (group III) with the unlexicalized models (group II) of Nguyen and Joty (2017).[3] We also report the results of non-neural entity grid models (Elsner and Charniak, 2011b) in group I. The extended versions use entity-specific features.

We experimented with both *random* and *pretrained* initialization for word embeddings in our lexicalized models. As can be noticed in Table 3, both versions give significant improvements over the unlexicalized models on both the standard and the inverse-order discrimination tasks (2.7 - 4.3% absolute). Our best model with Google pretrained embeddings (Mikolov et al., 2013) yields state-of-the-art results. We also experimented with Glove (Pennington et al., 2014), which has more vocabulary coverage than word2vec – Glove covers 89.77% of our vocabulary items, whereas word2vec covers 85.66%. However, Glove did not perform well giving F_1 score of 86% in the standard discrimination task. Schnabel et al. (2015) also report similar results where word2vec was found to be superior to Glove in most evaluation tasks. Our model also outperforms the extended neural grid model that relies on an additional feature extraction step for entity features. These results demonstrate the efficacy of lexicalization in capturing fine-grained entity information without loosing generalizability, thanks to distributed representation and pre-trained embeddings.

	Model	Emb.	**Std** (F_1)	**Inv** (F_1)
I	Grid (E&C)	-	81.60	75.78
	Ext. Grid (E&C)	-	84.95	80.34
II	Neural Grid (N&J)	Random	84.36	83.94
	Ext. Neural Grid (N&J)	Random	85.93	83.00
III	Lex. Neural Grid	Random	$87.03^\dagger$	$86.88^\dagger$
	Lex. Neural Grid	Google	$\mathbf{88.56}^\dagger$	$\mathbf{88.23}^\dagger$

Table 3: 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.

6 Experiments on Conversation

We evaluate our coherence models for asynchronous conversations on two tasks: discrimination and thread reconstruction.

6.1 Evaluation on Discrimination

The discrimination tasks are applicable to conversations also. We first present the dataset we use, then we describe how we create coherent and incoherent examples to train and test our models.

Dataset: Our conversational corpus contains discussion threads regarding *computer troubleshooting* from the technology related news site CNET.[4] This corpus was originally collected by Louis and Cohen (2015), and it contains 13,352 threads. For our experiments, we selected 3,825 threads assuring that each contains at least 3 and at most 15 posts. We use 2,400 threads for training, 750 for testing and 675 for development purposes. Table 4 shows some basic statistics about the resulting dataset. The threads roughly contain 29 sentences and 6 comments on average.

[2] https://ntunlpsg.github.io/project/coherence/n-coh-acl17

[3] Our reproduced results for the neural grid model are slightly lower than their reported results ($\sim 1\%$). We suspect this is due to the randomness in the experimental setup.

[4] https://www.cnet.com/

	#Thread	Avg Com	Avg Sen	#Pairs (tree)	#Pairs (path)
Train	2,400	6.01	28.76	47,948	106,122
Test	750	5.75	27.79	14,986	33,852
Dev	675	6.27	30.70	13,485	28,897
Total	3,825	5.98	28.77	76,419	168,871

Table 4: Statistics on the **CNET** dataset.

Model Settings and Training: To validate the efficacy of our conversational grid model, we compare it with the following baseline settings:

- **Temporal:** In the temporal setting, we construct an entity grid from the chronological order of the sentences in a conversation, and use it with our monologue-based coherence models. Models in this setting thus disregard the structure of the conversation and treat it as a monologue.
- **Path-level:** This is a special case of our model, where we consider each path (a column in our conversational grid) in the conversation tree separately. We construct an entity grid for a path and provide as input to our monologue-based models.

To train the models with pairwise ranking, we create 20 incoherent conversations for each original conversation by shuffling the sentences in their temporal order. For models involving conversation trees (path-level and our model), the tree structure remains unchanged for original and permuted conversations, only the position of the sentences vary based on the permutation. Since the shuffling is done globally at the conversation level, this scheme allows us to compare the three representations (temporal, path-level and tree-level) fairly with the same set of permutations.

An incoherent conversation may have paths in the tree that match the original paths. We remove those matched paths when training the path-level model. See Table 4 for number of pairs used for training and testing our models. We evaluate path-level models by aggregating correct/wrong decisions for the paths – if the model makes more correct decisions for the original conversation than the incoherent one, it is counted as a correct decision overall. Aggregating path-level *coherence scores* (*e.g.,* by averaging or summing) would allow a coherence model to get awarded for assigning higher score to an original path (hence, correct) while making wrong decisions for the rest; see supplementary document for an example. Similar to the setting in Monologue, we did not train explicitly on the inverse-order task, rather use the trained model from the standard setting.

Conv. Rep	Model	Emb.	**Std** (F_1)	**Inv** (F_1)
Temporal	Neural Grid (N&J)	random	82.28	70.53
	Lex. Neural Grid	random	86.63	80.40
	Lex. Neural Grid	Google	87.17	80.76
Path-level	Neural Grid (N&J)	random	82.39	75.68†
	Lex. Neural Grid	random	88.13	88.38†
	Lex. Neural Grid	Google	88.44	89.31†
Tree-level	Neural Grid (N&J)	random	83.98†	77.33†
	Lex. Neural Grid	random	89.87†	89.23†
	Lex. Neural Grid	Google	**91.29**†	**90.40**†

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

Results and Discussions: Table 5 compares the results of our models on the two discrimination tasks. We observe more gains in conversation than in monologue for the lexicalized models – 4.9% to 7.3% on the standard task, and 10% to 13.6% on the inverse-order task. Notice especially the huge gains on the inverse-order task. This indicates lexicalization helps to better adapt to new domains.

A comparison of the results on the standard task across the representations shows that path-level models perform on par with the temporal models, whereas the tree-level models outperform others by a significant margin. The improvements are 2.7% for randomly initialized word vectors and 4% for Google embeddings. Although, the path-level model considers some conversational structures, it observes only a portion of the conversation in its input. The common topics (expressed by entities) of a conversation get distributed across multiple conversational paths. This limits the path-level model to learn complex relationships between entities in a conversation. By encoding an entire conversation into a single grid and by modeling the spatial relations between the entities, our conversational grid model captures both local and global information (topic) of a conversation.

Interestingly, the improvements are higher on the inverse-order task for both path- and tree-level models. The inverse order yields more dissimilarity at the paths with respect to the original order, thus making them easier to distinguish.

If we notice the hyperparameter settings for the best models on this task (see supplementary document), we see they use a filter width of 1. This indicates that to find the right order of the sentences in conversations, it is sufficient to consider entity transitions along the conversational paths in a tree.

6.2 Evaluation on Thread Reconstruction

One crucial advantage of our tree-level model over other models is that we can use it to build predictive models to uncover the thread structure of a conversation from its posts. Consider again the thread in Figure 2. Our goal is to train a coherence model that can recover the tree structure in Figure 2(b) from the sequence of posts $(p_1, p_2, \ldots, p_5)$.

This task has been addressed previously (Wang et al., 2008, 2011). Most methods learn an edge-level classifier to decide for a possible link between two posts using features like distance in position/time, cosine similarity, etc. To our knowledge, we are the first to use coherence models for this problem. However, our goal in this paper is not to build a state-of-the-art system for thread reconstruction, rather to evaluate coherence models by showing its effectiveness in scoring candidate tree hypotheses. In contrast to previous methods, our approach therefore considers the whole thread structure at once, and computes coherence scores for all possible candidate trees of a conversation. The tree that receives the highest score is predicted as the thread structure of the conversation.

Training: We train our coherence model for thread reconstruction using pairwise ranking loss as before. For a given sequence of comments in a thread, we construct a set of valid candidate trees; a valid tree is one that respects the chronological order of the comments, *i.e.,* a comment can only reply to a comment that precedes it. The training set contains ordered pairs (T_i, T_j), where T_i is a true (gold) tree and T_j is a valid but false tree.

Experiments: The number of valid trees grows exponentially with the number of posts in a thread, which makes the inference difficult. As a proof of concept that coherence models are useful for finding the right tree, we built a simpler dataset by selecting forum threads from the CNET corpus ensuring that a thread contains at most 5 posts. The final dataset contains 1200 threads with an average of 3.8 posts and 27.64 sentences per thread.

We assess the performance of the models at two levels: (*i*) **thread-level**, where we evaluate if the model could identify the entire conversation thread correctly, and (*ii*) **edge-level**, where we evaluate if the model could identify individual replies correctly. For comparison, we use a number of simple but well performing baselines:

- **All-previous** creates thread structure by linking a comment to its previous (in time) comment.
- **All-first** creates thread structure by linking all the comments to the initial comment.
- **COS-sim** creates thread structure by linking a comment to one of the previous comments with which it has the highest cosine similarity. We use TF.IDF representation for the comments.

	Thread-level	**Edge-level**	
	Acc	F_1	Acc
All-previous	27.00	52.00	61.83
All-first	25.67	48.23	58.19
COS-sim	27.66	50.56	60.30
Conv. Entity Grid	**30.33**†	**53.59**†	**62.81**†

Table 6: Thread reconstruction results; † indicates significant difference from COS-sim (p< .01).

Table 6 compares our best conversational grid model (tree-level with Google vectors) with the baselines. The low thread-level accuracy across all the systems prove that reconstructing an entire tree is a difficult task. Models are reasonably accurate at the edge level. Our coherence model shows promising results, yielding substantial improvements over the baselines. It delivers 2.7% improvements in thread-level and 2.5% in edge-level accuracy over the best baseline (COS-sim).

Interestingly, our best model for this task uses a filter width of 2 (maximum can be 4 for 5 posts). This indicates that spatial (left-to-right) relations between entity transitions are important to find the right thread structure of a conversation.

7 Conclusion

We presented a coherence model for asynchronous conversations. We first extended the existing neural grid model by lexicalizing its entity transitions. We then adapt the model to conversational discourse by incorporating the thread structure in its grid representation and feature computation. We designed a 3D grid representation for capturing spatio-temporal entity transitions in a conversation tree, and employed a 2D convolution to compose high-level features from this representation.

Our lexicalized grid model yields state of the art results on standard coherence assessment tasks in monologue and conversations. We also show a novel application of our model in forum thread reconstruction. Our future goal is to use the coherence model to generate new conversations.

References

Alberto Barron-Cedeno, Simone Filice, Giovanni Da San Martino, Shafiq Joty, Lluís Màrquez, Preslav Nakov, and Alessandro Moschitti. 2015. Thread-level information for comment classification in community question answering. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, ACL'15, pages 687–693, Beijing, China. Association for Computational Linguistics.

Regina Barzilay and Mirella Lapata. 2008. Modeling local coherence: An entity-based approach. *Computational Linguistics*, 34(1):1–34.

Giuseppe Carenini, Raymond T. Ng, and Xiaodong Zhou. 2008. Summarizing emails with conversational cohesion and subjectivity. In *Proceedings of the 46nd Annual Meeting on Association for Computational Linguistics, ACL'08*, pages 353–361, OH. ACL.

Micha Elsner and Eugene Charniak. 2011a. Disentangling chat with local coherence models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 1179–1189, Stroudsburg, PA, USA. Association for Computational Linguistics.

Micha Elsner and Eugene Charniak. 2011b. Extending the entity grid with entity-specific features. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, HLT '11, pages 125–129, Portland, Oregon. Association for Computational Linguistics.

Vanessa Wei Feng and Graeme Hirst. 2012. Extending the entity-based coherence model with multiple ranks. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '12, pages 315–324, Avignon, France. Association for Computational Linguistics.

Vanessa Wei Feng, Ziheng Lin, and Graeme Hirst. 2014. The impact of deep hierarchical discourse structures in the evaluation of text coherence. In *COLING*.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *JMLR W&CP: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS 2010)*, volume 9, pages 249–256, Sardinia, Italy.

Barbara J. Grosz, Scott Weinstein, and Aravind K. Joshi. 1995. Centering: A framework for modeling the local coherence of discourse. *Comput. Linguist.*, 21(2):203–225.

Amit Gruber, Yair Weiss, and Michal Rosen-Zvi. 2007. Hidden topic markov models. In *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, volume 2 of *Proceedings of Machine Learning Research*, pages 163–170, San Juan, Puerto Rico. PMLR.

Camille Guinaudeau and Michael Strube. 2013. Graph-based local coherence modeling. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 1: Long Papers*, pages 93–103.

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 448–456. JMLR.org.

Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, pages 133–142, Edmonton, Alberta, Canada. ACM.

Shafiq Joty, Giuseppe Carenini, and Raymond T. Ng. 2013. Topic segmentation and labeling in asynchronous conversations. *J. Artif. Int. Res.*, 47(1):521–573.

Jiwei Li and Eduard Hovy. 2014. A model of coherence based on distributed sentence representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2039–2048, Doha, Qatar. Association for Computational Linguistics.

Jiwei Li and Dan Jurafsky. 2017. Neural net models of open-domain discourse coherence. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 198–209, Copenhagen, Denmark. Association for Computational Linguistics.

Ziheng Lin, Hwee Tou Ng, and Min-Yen Kan. 2011. Automatically evaluating text coherence using discourse relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 997–1006, Portland, Oregon. Association for Computational Linguistics.

Annie Louis and Shay B. Cohen. 2015. Conversation trees: A grammar model for topic structure in forums. In *EMNLP*.

Annie Louis and Ani Nenkova. 2012. A coherence model based on syntactic patterns. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 1157–1168, Stroudsburg, PA, USA. Association for Computational Linguistics.

William C Mann and Sandra A Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8(3):243–281.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Dat Nguyen and Shafiq Joty. 2017. A neural local coherence model. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1320–1330. Association for Computational Linguistics.

Sebastian Padó. 2006. *User's guide to `sigf`: Significance testing by approximate randomisation*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. 2008. The penn discourse treebank 2.0. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Tobias Schnabel, Igor Labutov, David M Mimno, and Thorsten Joachims. 2015. Evaluation methods for unsupervised word embeddings. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 298–307.

T. Tieleman and G Hinton. 2012. *RMSprop*. COURSERA: Neural Networks

Li Wang, Marco Lui, Su Nam Kim, Joakim Nivre, and Timothy Baldwin. 2011. Predicting thread discourse structure over technical web forums. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 13–25, Stroudsburg, PA, USA. Association for Computational Linguistics.

Yi-Chia Wang, Mahesh Joshi, William Cohen, and Carolyn Ros. 2008. Recovering implicit thread structure in newsgroup style conversations. In *Proceedings of the Eleventh International Conference on Web and Social Media, ICWSM 2008.*