Thread Reconstruction in Conversational Data using Neural Coherence Models

Nguyen, D.T.; Joty, S.; Boussaha, B.E.A.; de Rijke, M.

[Link to publication](#)

Thread Reconstruction in Conversational Data using Neural Coherence Models

Dat Tien Nguyen
University of Amsterdam
Amsterdam, Netherlands
t.d.nguyen@uva.nl

Shafiq Joty
Nanyang Technological University
Singapore, Singapore
srjoty@ntu.edu.sg

Basma El Amel Boussaha
University of Nantes
Nantes, France
basma.boussaha@univ-nantes.fr

Maarten de Rijke
University of Amsterdam
Amsterdam, Netherlands
derijke@uva.nl

ABSTRACT

Discussion forums are an important source of information. They are often used to answer specific questions a user might have and to discover more about a topic of interest. Discussions in these forums may evolve in intricate ways, making it difficult for users to follow the flow of ideas. We propose a novel approach for automatically identifying the underlying thread structure of a forum discussion. Our approach is based on a neural model that computes coherence scores of possible reconstructions and then selects the highest scoring, i.e., the most coherent one. Preliminary experiments demonstrate promising results outperforming a number of strong baseline methods.

KEYWORDS

Thread reconstruction; Coherence model; Convolutional neural network

1 INTRODUCTION

Discussion forums are an important source of information, both to help a user answer specific information needs they might have and to help users who want to explore a topic without having a specific question in mind. Discussions in these forums are typically composed of multiple inter-woven threads, regardless of whether that threaded structure is made explicit in the representation and presentation of the conversational data [19]. Recovering the underlying thread structure is helpful for users of discussion forums as it makes possible to disentangle discussions related to subtopics and/or to particular conversational goals [20].

More precisely, the thread reconstruction task is defined as follows. Given a discussion stream in which messages are sorted by posting time, the thread reconstruction task is to construct the thread tree. The edges in the tread tree identify which of the previously contributed posts the current post replies to. For example,

consider the example thread from CNET forum site[1] in Figure 1, where we have five (p)osts. The thread has a tree structure with three branches: $p_1 \leftarrow p_2$, $p_1 \leftarrow p_3$ and $p_1 \leftarrow p_4 \leftarrow p_5$. Given the posts $\{p_1, p_2, \ldots, p_5\}$, our goal in thread reconstruction is to recover the underlying tree structure of the thread.

Several methods have been proposed for the thread reconstruction task [19–21]. These methods learn an edge-level classifier to decide for a possible connection using features like distance in position/time, cosine similarity between comments, etc. However, these models suffer from the limitation that they consider one edge at a time rather than the global tree structure of the thread. Modeling edges locally disregards interactions between all possible edges and can lead to suboptimal solutions. In contrast, in this paper we propose to model an entire thread for the reconstruction task. We propose to use a neural coherence model [15] from natural language processing (NLP) for scoring candidate tree hypotheses.

Coherence models [2, 7] were originally proposed for coherence assessment of monologues (e.g., news articles, books). However, forum conversations are different from monologues in the sense that information flow in these conversations are not sequential; topics in these conversations are often interleaved in the temporal order of the comments [10, 13]. For example, in the thread in Figure 1, there are three possible subconversations each corresponding to a branch. The branch $p_1 \leftarrow p_2$ suggests *using regedit*, the branch $p_1 \leftarrow p_3$ suggests *ccleaner*, and the third branch suggests *regseeker*. Beacause of these differences, when applied directly to these conversations, the coherence models may not perform as we expect. Furthermore, these models are not specifically trained for the reconstruction task.

In this paper, we make the following contributions. First, we hypothesize that coherence models should consider the thread structure of a conversation and we extend the original grid representation proposed by Barzilay and Lapata [2] to encode the thread structure of a forum conversation. Then we train a convolutional neural network (CNN) model with pairwise ranking using the grid representation for the thread reconstruction task. Our method considers the whole thread structure at once, and computes coherence scores for all possible candidate trees. The highest scoring tree corresponds to the predicted tree structure for the given thread.

We evaluated our approach on discussion threads from CNET. The results show that our method is quite promising outperforming several strong baselines on this dataset.
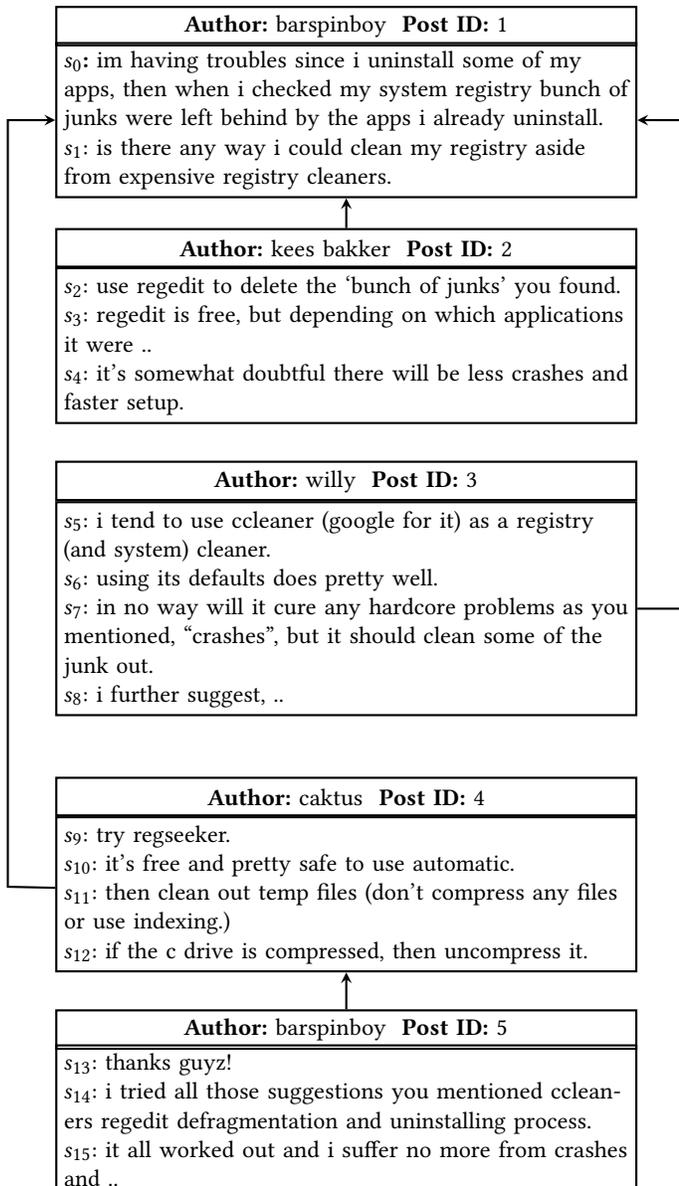
---

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

Figure 1: A truncated forum thread from CNET with five comments by temporal order. *Reply-to* **links between posts are denoted by arrowed edges.**

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.
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 (and system) cleaner.
s_6: using its defaults does pretty well.
s_7: in no way will it cure any hardcore problems as you mentioned, "crashes", but it should clean some of the junk out.
s_8: i further suggest, ..

Author: caktus **Post ID:** 4

s_9: try regseeker.
s_{10}: it's free and pretty safe to use automatic.
s_{11}: then clean out 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!
s_{14}: i tried all those suggestions you mentioned ccleaners regedit defragmentation and uninstalling process.
s_{15}: it all worked out and i suffer no more from crashes and ..

2 COHERENCE MODELS

In this section, we give a brief overview of the coherence models that were originally proposed for monologues (e.g., news articles) and that are related to our work. In the next section, we propose extensions to these models for forum-like conversations that we use for thread reconstruction.

2.1 Entity Grid and Its Extensions

Barzilay and Lapata [2] proposed an entity-based model for representing and assessing text coherence. Their model represents a text by a matrix called *entity grid* that captures transitions of entities (i.e., noun phrases) across sentences. As shown in Table 1, the rows of the grid correspond to sentences, and the columns correspond to entities appearing in the text. Each entry $G_{i,j}$ in the entity grid represents the syntactic role that entity e_j plays in sentence s_i, which can be one of: subject (**S**), object (**O**), or other (**X**). Entities not appearing in a sentence are marked by a special symbol (**-**).

To represent the grid using a feature vector, Barzilay and Lapata [2] compute probability for each local entity transition of length k (i.e., $\{S, O, X, -\}^k$), and represent each grid by a vector of 4^k transitions probabilities. Coherence assessment is then formulated as a ranking task in an SVM preference ranking framework [9].

A number of extensions of the basic entity grid model have been proposed. Elsner and Charniak [7] extended the basic grid to distinguish between entities of different types by incorporating entity-specific features like named entity, noun class, modifiers, etc. Feng and Hirst [8] used the basic grid representation, but improved its learning to rank scheme. Their model learns not only from original document and its permutations but also from ranking preferences among the permutations themselves.

2.2 Neural Entity Grid Model

Although the entity grid and its extensions have been successfully applied to many downstream applications including coherence rating [2], readability assessment [2, 16], essay scoring [4], and story generation [14], they have some limitations. First, they use discrete representation for grammatical roles and features, which leads to the so-called **curse of dimensionality** problem [3]. In particular, to model transitions of length k with $\mathcal{C}$ different grammatical roles, the basic entity grid model needs to compute $\mathcal{C}^k$ transition probabilities from a *single* grid. The estimated distribution becomes sparse as k increases, which prevents the model from considering longer transitions – existing models typically use $k \leq 3$. Second, these models compute feature representations from entity grids in a task-agnostic way. Decoupling feature extraction from the target task can limit the model's capacity to learn task-specific features.

To deal with the above issues of entity grid models, we [15] recently proposed a neural extension to the grid models. As shown in Figure 2, the neural model takes an extracted entity grid as input, and transforms each grammatical role in the grid into a distributed representation by looking up a shared embedding matrix $E \in \mathbb{R}^{|\mathcal{V}| \times d}$, where $\mathcal{V} = \{S, O, X, -\}$ is a set of grammatical roles, and d is the embedding dimensions. The embedding vectors produced by the lookup layer are combined by subsequent layers of the network to generate a coherence score for the document. The network uses a *convolutional* layer, which applies N *filters* to get N different *feature maps*. The abstract features in each feature map are then pooled using a *max-pooling* operation. The pooled features are then used for coherence scoring at the final layer of the model.

Convolution learns to compose local transitions of a grid into higher-level representations, while max-pooling captures the most salient local features from each feature map. Since the convolution-pooling operates over the distributed representation of grid entries, compared to traditional grid models, the transition length k can be sufficiently large to capture long-range dependencies without overfitting on the training data. Also, the embedding vectors and the convolutional filters are learned from all training documents
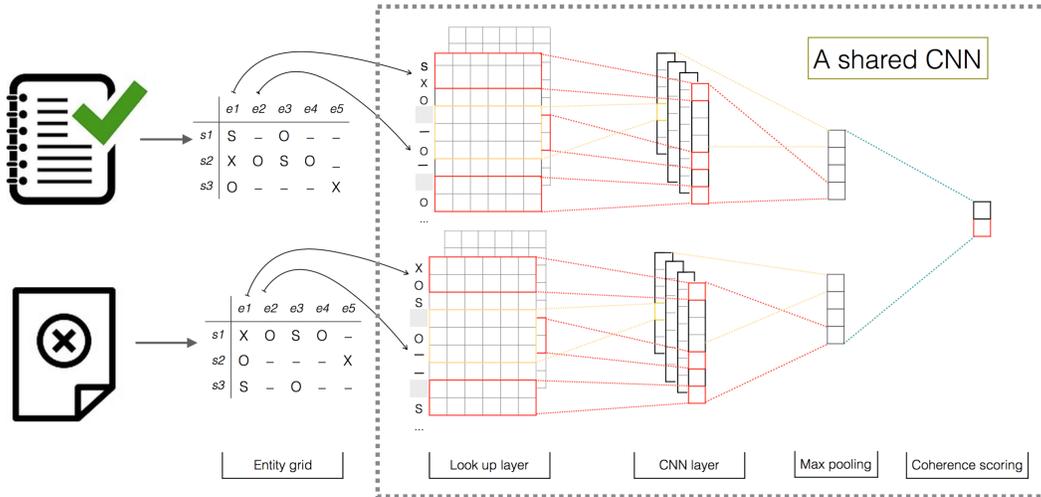
Figure 2: Neural model for coherence scoring and the pairwise training method [taken from our previous work [15]].

as opposed to a single document in traditional grid models, which helps the neural model to obtain better generalization and robustness. The evaluation on three different coherence assessment tasks demonstrates the superiority of the neural model yielding state of the art results. In this work, we therefore extend the neural model for forum-like conversations and use it for thread reconstruction.

3 NEURAL COHERENCE MODEL FOR FORUM THREADS

The main difference between forum conversations and monologues is that the information flow in forum conversations is often not sequential as in monologue. As a result, the coherence models that are originally developed for monologues may not perform as expected when they are directly applied to threaded conversations [10]. We hypothesize that the coherence models should consider the conversational structure in the form of "reply-to" relations between comments as shown by a tree structure in Figure 1. In the following subsections, we describe how we extend the neural entity grid model to incorporate the tree structure of a thread.

3.1 Entity Grid for Forum Threads

The thread structure in Figure 1 has a tree structure, where nodes represent comments and edges represent "reply-to" links between comments. Since entity grid models operate at the sentence level, we construct the conversational thread at the sentence level. We do this by linking the boundary sentences across comments and by linking sentences in the same comment chronologically; i.e., we connect the first sentence of comment c_j to the last sentence of comment c_i if c_j is a reply to c_i, and sentence s_{t+1} is linked to s_t if both s_t and s_{t+1} are in the same comment.

To encode a sentence-level conversation tree into an entity grid, we propose couple of modifications to the original entity grid representation. In the modified representation as shown in Table 1, rows represent depth levels of the conversation tree as opposed to sentences in the original grid. An entry $G_{i,j}$ in our conversational entity grid represents the sequence of grammatical roles (left to

right) that the entity e_j plays in the sentences occurring at the j-th level of the conversation tree. For instance, our example tree has three sentences s_3, s_6 and s_{10} at depth level 3. The entity *REGEDIT* has the role of a *Subject*, a *not present* and a *not present*, respectively, in these three sentences, thus encoded as 'S--' in the entity grid.

3.2 Thread Reconstruction

The conversational entity grid captures transition of entities in terms of their grammatical roles in a conversation tree. We believe this representation can be quite useful for thread reconstruction – i.e., discovering the latent structure of a forum thread.

We train a convolutional neural network (we refer to our model as **Grid-CNN** for the rest of this paper) using the conversational entity grid representation for the thread reconstruction task. The CNN model has the same structure as described in Section 2.2.

We use a *pairwise ranking* approach [5] to train the Grid-CNN model. For a given number of comments in a gold tree, we first construct a set of valid candidate trees. A valid tree is one that respects the chronological order of the comments in a thread – for example, a comment can only reply to a comment that comes before in the temporal order. The training set comprises *ordered* pairs (T_i, T_j), where thread T_i is a *true (gold)* tree and T_j is a valid but *false* tree. We seek to find model parameters that assign a higher score to T_i than to T_j. We minimize the following ranking loss:

$$\mathcal{J}(\theta) = \max\{0, 1 - \phi(G_i|\theta) + \phi(G_j|\theta)\} \qquad (1)$$

where G_i and G_j are the conversational entity grids corresponding to threads T_i and T_j, respectively, and θ defines the model parameters including the embedding matrix and the weight vectors.

During testing, our Grid-CNN model predicts coherence scores for all the possible candidate trees given the posts in a thread, and the tree with the highest score is considered to be the underlying structure of the thread.

Table 1: Transition of some entities across tree structure of the thread example. Legend: S stands for subject, O for object, X for a role other than subject or object, and – means that an entity does not appear in the sentence.

Tree structure	depth	CLEANER	REGEDIT	TROUBLES	SYSTEM	JUNKS	APPS	REGISTRY	BUNCH
s_0	0	–	–	–	O	X	X	*O*	*O*
s_1	1	O	–	–	–	–	–	*O*	–
s_2 s_5 s_9	2	–O–	O––	– – –	– – –	X––	– – –	–*O*–	O––
s_3 s_6 s_{10}	3	– – –	S––	– – –	– – –	– – –	– – –	– – –	– – –
s_4 s_7 s_{11}	4	– – –	– – –	– – –	– – –	–*X*–	– – –	– – –	– – –
s_8 s_{12}	5	––	––	––	––	––	––	––	––

4 EXPERIMENT

Our main research question is to determine whether text coherence can benefit thread reconstruction; we want to assess at two levels, the level of trees ("did we reconstruct the discourse correctly") and the level of edges ("did we identify individual replies correctly").

4.1 Experimental setup

As a preliminary experiment, we collect 2,200 threads from the CNET dataset made available in previous research [11] that have less than 6 posts. We remove from the raw data some meta information and keep the temporal order (post ID) of the dataset. We leave 1,500 threads for training, 200 threads as our development set and the rest for testing. See Table 2 for further details about the corpus.

Table 2: Corpus statistics. *Non-trivial* replies are posts that reply to other posts except the first post.

# Threads	Avg. # Posts	Avg. # Sent	Non-trivial replies
2,200	3.6	27.64	57%

We train the Grid-CNN model by optimizing the pairwise ranking loss in Equation 1 using the gradient-based online learning algorithm RMSprop[18]. [2] We use up to 25 epochs. We use dropout [17] of hidden units to avoid overfitting, and do *early stopping* by observing accuracy on the DEV set – if the performance (accuracy score) does not increase for 10 consecutive epochs, we stop training and pick the best model recorded so far. We search for optimal mini**batch** size in $\{16, 32, 64, 128\}$, **emb**edding size in $\{50, 80, 100, 200\}$, **dropout** rate in $\{0.2, 0.3, 0.5\}$, **filter** number in $\{100, 150, 200, 300\}$, **win**dow size in $\{3, 4, 5, 6, 7, 8\}$, and **pool**ing length in $\{4, 5, 6, 7\}$. The best model (see Table 3) on the development set is then used for the final evaluation on the test set.

For comparison, we use a number of simple but well performing baselines:

All-previous Create thread structure by linking a comment to its previous comment.

All-first Create thread structure by linking all the comments to the first comment.

COS-sim Create thread structure by linking a comment to one of the previous comments with which it has the highest cosine similarity.

	Batch	Emb.	Dropout	Filter	Win.	Pool
Grid-CNN	64	100	0.5	150	6	6

Table 3: Optimal hyper-parameter setting for our neural models based on development set accuracy.

We consider two variations of the thread reconstruction task: tree-level and edge-level. For the tree-level version of the thread reconstruction task, the model predicted tree should match entirely the original structure of the thread. For the edge-level version, we measure the post-level (the link between two posts) classification performance. If a link appears in the predicted tree and the original one, we count as a true prediction; otherwise it is a false case.

4.2 Results and discussion

Table 4 presents the results for the baselines and our method, Grid-CNN. The **COS-sim** gets the lowest score on both reconstruction tasks. This means reply structure between two posts does not rely only on their term matching patterns. The other two baselines **All-previous** and **All-first** perform quite well on our dataset. This is not surprising since it is very common in this kind of forums that most participants reply either to the previous post or to the first post that asks the question.

Our neural method gets promising results, yielding substantial improvements over the baselines in all cases. The **Grid-CNN** model delivers relative improvements from 32% to 57% in accuracy for the tree-level reconstruction task. It also outperforms the baselines in the edge-level prediction task with improvements from 4% to 13% in F_1-score and from 1% to 12% in accuracy.

We further manually inspected the false prediction cases for our method. We observed that most of the false trees fall into the trivial structures (All-previous or All-first). This could be due to the dominance of these cases in our training data – 40.07% of the posts reply to the first post and 76.29% reply to the previous post.

5 RELATED WORK

Several previous studies treat thread reconstruction as an edge-level classification problem. Wang et al. [21] use cosine similarity between posts and exploit temporal order information (e.g., time distance, post distance) to recover the thread structure. Aumayr et al. [1] consider thread reconstruction as a classification problem. They

---

[2]Other adaptive algorithms, e.g., ADAM [12], ADADELTA [22] gave similar results.

Table 4: Performance on the thread reconstruction task.

	Tree-level	Edge-level	
	Acc	F_1	Acc
All-previous	20.00	58.45	65.62
All-first	17.60	54.90	60.27
COS-sim	16.80	53.58	58.75
Grid-CNN	**26.40**	**60.55**	**66.12**

train a decision tree classifier based on some basic features such as reply distance in number of posts, time distance, cosine similarity and thread lengths, etc. Their model takes a pair of posts as input and predicts the link between them. A joint model using dependency parsing and conditional random fields was proposed to predict links between two posts and their dialogue acts [19]. Dehghani et al. [6] works on reconstructing tree structure of conversation threads in email data.

In contrast to previous approaches, we treat thread reconstruction as a ranking problem and use a neural coherence model to rank all possible candidate trees. We show that modeling coherence of threaded conversations is an effective approach to thread reconstruction.

6 CONCLUSIONS

This paper introduces a novel approach to solve thread reconstruction problem in discussion forums. Our method uses a neural coherence model based on an entity grid representation and a convolutional neural network (CNN). First, we extend the original grid representation to encode the thread structure of a forum conversation. Then we train a CNN model with pairwise ranking using the grid representation for the thread reconstruction task. Our method considers the whole thread structure at once, and computes coherence scores for all possible candidate trees. The highest scoring tree is returned as the predicted tree structure.

We evaluated our approach on discussion threads from CNET forum site. The result shows that our method is very promising. It significantly improves performance over trivial baselines, particularly for the tree-level accuracy. In the future, we would like to experiment with larger datasets containing threads with many posts. We also plan to integrate other discourse structures like dialogue acts into our model to get further improvements.

REFERENCES

[1] Erik Aumayr, Chan Jeffrey, and Conor Hayes. 2011. Reconstruction of Threaded Conversations in Online Discussion Forums. In *Proceedings of the Eleventh International Conference on Web and Social Media, ICWSM 2011.*

[2] Regina Barzilay and Mirella Lapata. 2008. Modeling Local Coherence: An Entity-Based Approach. *Computational Linguistics* 34, 1 (2008), 1–34. http://www.aclweb.org/anthology/J08-1001

[3] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A Neural Probabilistic Language Model. *J. Mach. Learn. Res.* 3 (March 2003). http://dl.acm.org/citation.cfm?id=944919.944966

[4] Jill Burstein, Joel Tetreault, and Slava Andreyev. 2010. Using Entity-based Features to Model Coherence in Student Essays. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT '10)*. Association for Computational Linguistics, Los Angeles, California, 681–684.

[5] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research* 12 (2011), 2493–2537.

[6] Mostafa Dehghani, Azadeh Shakery, Masoud Asadpour, and Arash Koushkestani. 2013. A Learning Approach for Email Conversation Thread Reconstruction. *J. Inf. Sci.* 39, 6 (Dec. 2013), 846–863. https://doi.org/10.1177/0165551513494638

[7] Micha Elsner and Eugene Charniak. 2011. 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)*. Association for Computational Linguistics, Portland, Oregon, 125–129.

[8] 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)*. Association for Computational Linguistics, Avignon, France, 315–324.

[9] 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)*. ACM, Edmonton, Alberta, Canada, 133–142.

[10] Shafiq Joty, Giuseppe Carenini, and Raymond T. Ng. 2013. Topic Segmentation and Labeling in Asynchronous Conversations. *J. Artif. Int. Res.* 47, 1 (May 2013), 521–573. http://dl.acm.org/citation.cfm?id=2566972.2566986

[11] Su Nam Kim, Li Wang, and Timothy Baldwin. 2010. Tagging and linking web forum posts. In *CoNLL-2010*. 192–202.

[12] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *CoRR* abs/1412.6980 (2014). http://arxiv.org/abs/1412.6980

[13] Annie P Louis and Shay B Cohen. 2015. Conversation trees: A grammar model for topic structure in forums. Association for Computational Linguistics.

[14] Neil McIntyre and Mirella Lapata. 2010. Plot Induction and Evolutionary Search for Story Generation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL '10)*. Association for Computational Linguistics, Uppsala, Sweden, 1562–1572.

[15] Dat Tien 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) (ACL '17)*. Association for Computational Linguistics, Vancouver, Canada, (to appear).

[16] Emily Pitler, Annie Louis, and Ani Nenkova. 2010. Automatic Evaluation of Linguistic Quality in Multi-document Summarization. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL '10)*. Association for Computational Linguistics, Uppsala, Sweden, 544–554.

[17] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research* 15 (2014), 1929–1958.

[18] T. Tieleman and G Hinton. 2012. RMSprop. (2012).

[19] 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)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 13–25. http://dl.acm.org/citation.cfm?id=2145432.2145435

[20] Yi-Chia Wang, Mahesh Joshi, William Cohen, and Carolyn Rosé. 2008. Recovering implicit thread structure in newsgroup style conversations. In *AAAI*.

[21] Yi-Chia Wang, Mahesh Joshi, William Cohen, and Carolyn RosÃI. 2008. Recovering Implicit Thread Structure in Newsgroup Style Conversations. In *Proceedings of the Eleventh International Conference on Web and Social Media, ICWSM 2008.*

[22] Matthew D. Zeiler. 2012. ADADELTA: An Adaptive Learning Rate Method. *CoRR* abs/1212.5701 (2012). http://arxiv.org/abs/1212.5701