Evaluating Discourse in Structured Text Representations

Elisa Ferracane[1], **Greg Durrett**[2], **Junyi Jessy Li**[1] and **Katrin Erk**[1]

[1]Department of Linguistics
[2]Department of Computer Science
The University of Texas at Austin
elisa@ferracane.com, gdurrett@cs.utexas.edu
jessy@austin.utexas.edu, katrin.erk@mail.utexas.edu

Abstract

Discourse structure is integral to understanding a text and is helpful in many NLP tasks. Learning *latent* representations of discourse is an attractive alternative to acquiring expensive labeled discourse data. Liu and Lapata (2018) propose a structured attention mechanism for text classification that derives a tree over a text, akin to an RST discourse tree. We examine this model in detail, and evaluate on additional discourse-relevant tasks and datasets, in order to assess whether the structured attention improves performance on the end task and whether it captures a text's discourse structure. We find the learned latent trees have little to no structure and instead focus on lexical cues; even after obtaining more structured trees with proposed model modifications, the trees are still far from capturing discourse structure when compared to discourse dependency trees from an existing discourse parser. Finally, ablation studies show the structured attention provides little benefit, sometimes even hurting performance.[1]

1 Introduction

Discourse describes how a document is organized, and how discourse units are rhetorically connected to each other. Taking into account this structure has shown to help many NLP end tasks, including summarization (Hirao et al., 2013; Durrett et al., 2016), machine translation (Joty et al., 2017), and sentiment analysis (Ji and Smith, 2017). However, annotating discourse requires considerable effort by trained experts and may not always yield a structure appropriate for the end task. As a result, having a model induce the discourse structure of a text is an attractive option. Our goal in this paper is to evaluate such an induced structure.

Inducing structure has been a recent popular approach in syntax (Yogatama et al., 2017; Choi et al., 2018; Bisk and Tran, 2018). Evaluations of these latent trees have shown they are inconsistent, shallower than their explicitly parsed counterparts (Penn Treebank parses) and do not resemble any linguistic syntax theory (Williams et al., 2018).

For discourse, Liu and Lapata (2018) (L&L) induce a document-level structure while performing text classification with a structured attention that is constrained to resolve to a non-projective dependency tree. We evaluate the document-level structure induced by this model. In order to compare the induced structure to existing linguistically-motivated structures, we choose Rhetorical Structure Theory (RST) (Mann and Thompson, 1988), a widely-used framework for discourse structure, because it also produces tree-shaped structures.[2] We evaluate on some of the same tasks as L&L, but add two more tasks we theorize to be more discourse-sensitive: text classification of writing quality, and sentence order discrimination (as proposed by Barzilay and Lapata (2008)).

Our research uncovers multiple negative results. We find that, contrary to L&L, the structured attention does not help performance in most cases; further, the model is not learning discourse. Instead, the model learns trees with little to no structure heavily influenced by lexical cues to the task. In an effort to induce better trees, we propose several principled modifications to the model, some of which yield more structured trees. However, even the more structured trees bear little resemblance to ground truth RST trees.

We conclude the model holds promise, but re-

[1]Code and data available at https://github.com/elisaF/structured

[2]The Penn Discourse Treebank (PDTB; Prasad et al., 2008) captures lexically-grounded discourse for individual connectives and adjacent sentences, and does not span an entire document; Segmented Discourse Representation Theory (Lascarides and Asher, 2008) is a graph.

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 646–653
Florence, Italy, July 28 - August 2, 2019.

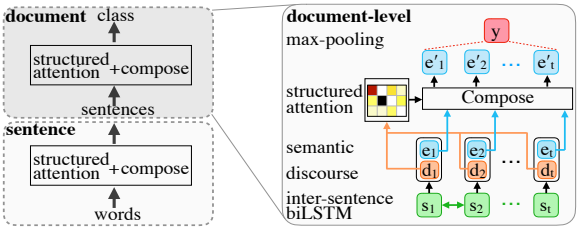


Figure 1: Model of Liu and Lapata (2018) with the document-level portion (right) that composes sentences into a document representation.

quires moving beyond text classification, and injecting supervision (as in Strubell et al. (2018)).

Our contributions are (1) comprehensive performance results on existing and additional tasks and datasets showing document-level structured attention is largely unhelpful, (2) in-depth analyses of induced trees showing they do not represent discourse, and (3) several principled model changes to produce better structures but that still do not resemble the structure of discourse.

2 Rhetorical Structure Theory (RST)

In RST, coherent texts consist of minimal units, which are linked to each other, recursively, through rhetorical relations (Mann and Thompson, 1988). Thus, the goal of RST is to describe the rhetorical organization of a text by using a hierarchical tree structure that captures the communicative intent of the writer. An RST discourse tree can further be represented as a *discourse dependency tree*. We follow the algorithm of Hirao et al. (2013) to create an unlabelled dependency tree based on the nuclearity of the tree.

3 Models

We present two models: one for text classification, and one for sentence ordering. Both are based on the L&L model, with a design change to cause stronger percolation of information up the tree (we also experiment without this change).

Text classification The left-hand side of Figure 1 presents an overview of the model: the model operates first at the sentence-level to create sentence representations, and then at the document-level to create a document representation from the previously created sentence representations. In more detail, the model composes GloVe embeddings (Pennington et al., 2014) into a sentence representation using structured attention (from which a tree can be derived), then sentence representations into a single document representation for class prediction. At both sentence and document level, each object (word or sentence, respectively) attends to other objects that could be its parent in the tree. Since the sentence and document-level parts of the model are identical, we focus on the document level (Figure 1, right), which is of interest to us for evaluating discourse effects.

Sentence representations $s_1, \ldots, s_t$ are fed to a bidirectional LSTM, and the hidden representations $[h_1, \ldots, h_t]$ consist of a semantic part (e_t) and a structure part (d_t): $[\mathbf{e}_t, \mathbf{d}_t] = \mathbf{h}_t$. Unnormalized scores $\mathbf{f}_{ij}$ representing potentials between parent i and child j are calculated using a bilinear function over the structure vector:

$$\mathbf{t}_p = \tanh(\mathbf{W}_p \mathbf{d}_i); \qquad \mathbf{t}_c = \tanh(\mathbf{W}_c \mathbf{d}_j) \quad (1)$$

$$\mathbf{f}_{ij} = \mathbf{t}_p^T \mathbf{W}_a \mathbf{t}_c \quad (2)$$

The matrix-tree theorem allows us to compute marginal probabilities a_{ij} of dependency arcs under the distribution over non-projective dependency trees induced by $\mathbf{f}_{ij}$ (details in Koo et al. (2007)). This computation is fully differentiable, allowing it to be treated as another neural network layer in the model. We importantly note the model only uses the *marginals*. We can post-hoc use the Chu-Liu-Edmonds algorithm to retrieve the highest-scoring tree under $\mathbf{f}$, which we call $\mathbf{f_{best}}$ (Chu and Liu, 1965; Edmonds, 1967).

The semantic vectors of sentences $\mathbf{e}$ are then updated using this attention. Here we diverge from the L&L model: in their implementation,[3] each node is updated based on a weighted sum over its *parents* in the tree (their paper states both parents and children).[4] We instead inform each node by a sum over its *children*, more in line with past work where information more intuitively percolates from children to parents and not the other way (Ji and Smith, 2017) (we also run experiments without this design change). We calculate the context for all possible children of that sentence as:

$$c_i = \sum_{k=1}^{n} a_{ik} e_k \quad (3)$$

where a_{ik} is the probability that k is the child of i, and e_k is the semantic vector of the child.

The children vectors are then passed through a non-linear function, resulting in the *updated* semantic vector e'_i for parent node i.

$$e'_i = \tanh(W_r[e_i, c_i]) \quad (4)$$

[3] https://github.com/nlpyang/structured

[4] We found similar results for using both parents and children as well as using parents only.

	Yelp	Debates	WQ	WQTC	WSJSO
L&L(orig)	68.51 | **68.27** (0.19)	81.82 | 79.48 (2.90)	84.14 | **82.69** (1.36)	80.73 | 79.63 (1.03)	96.17 | **95.29** (0.84)
L&L(ours)	68.51 | **68.23** (0.23)	78.88 | 77.81 (1.80)	84.14 | **82.70** (1.36)	82.49 | **81.11** (0.95)	95.57 | **94.76** (1.11)
−doc attn	68.34 | **68.13** (0.17)	82.89 | **81.42** (1.08)	83.75 | **82.80** (0.94)	80.60 | 79.25 (0.94)	95.57 | **95.11** (0.42)
−both attn	68.19 | 68.05 (0.13)	79.95 | 77.34 (1.79)	84.27 | **83.16** (1.25)	77.58 | 76.16 (1.25)	95.23 | **94.68** (0.37)
L&L(reported)	68.6	76.5	-	-	-

Table 1: Max | mean (standard deviation) accuracy on the test set averaged across four training runs with different initialization weights. Bolded numbers are within 1 standard deviation of the best performing model. L&L(orig) uses the original L&L code; L&L(ours) includes the design change and bug fix. L&L(reported) lists results reported by L&L on a single training run.

Finally, a max pooling layer over e'_i followed by a linear layer produces the predicted document class y. The model is trained with cross entropy loss.

Additionally, the released L&L implementation has a bug where attention scores and marginals are not masked correctly in the matrix-tree computation, which we correct.

Sentence order discrimination This model is identical, except for task-specific changes. The goal of this synthetic task, proposed by Barzilay and Lapata (2008), is to capture discourse coherence. A negative class is created by generating random permutations of a text's original sentence ordering (the positive class). A coherence score is produced for each positive and negative example, with the intuition that the originally ordered text will be more coherent than the jumbled version. Because we compare two examples at a time (original and permuted order), we modify the model to handle paired inputs and replace cross-entropy loss with a max-margin ranking loss.

4 Experiments

We evaluate the model on four text classification tasks and one sentence order discrimination task.

4.1 Datasets

Details and statistics are included in Appendix A.[5]

Yelp (in L&L, 5-way classification) comprises customer reviews from the Yelp Dataset Challenge (collected by Tang et al. (2015)). Each review is labeled with a 1 to 5 rating (least to most positive).

Debates (in L&L, binary classification) are transcribed debates on Congressional bills from the U.S. House of Representatives (compiled by Thomas et al. (2006), preprocessed by Yogatama and Smith (2014)). Each speech is labeled with 1 or 0 indicating whether the speaker voted in favor of or against the bill.

Writing quality (WQ) (not in L&L, binary classification) contains science articles from the New York Times (extracted from Louis and Nenkova (2013)). Each article is labeled as either 'very good' or 'typical' to describe its writing quality. While both classes contain well-written text, Louis and Nenkova (2013) find features associated with discourse including sentiment, readability, along with PDTB-style discourse relations are helpful in distinguishing between the two classes.

Writing quality with topic control (WQTC) (not in L&L, binary classification) is similar to WQ, but controlled for topic using a topic similarity list included with the WQ source corpus.[6]

Wall Street Journal Sentence Order (WSJSO) (not in L&L, sentence order discrimination) is the WSJ portion of PTB (Marcus et al., 1993).

4.2 Settings

For each experiment, we train the model four times varying only the random seed for weight initializations. The model is trained for a fixed amount of time, and the model from the timestep with highest development performance is chosen. We report accuracies on the test set, and tree analyses on the development set. Our implementation is built on the L&L released implementation, with changes as noted in Section 3. Preprocessing and training details are in Appendix A.

4.3 Results

We report accuracy (as in prior work) in Table 1, and perform two ablations: removing the structured attention at the document level, and removing it at both document and sentence levels. Additionally, we run experiments on the original code

[5] Of the document-level datasets used in L&L (SNLI was sentence-level), we omit IMDB and Czech Movies because on IMDB their model did not outperform prior work, and Czech (a language with freer word order than English) highlighted the non-projectivity of their sentence-level trees, which is not the focus of our work.

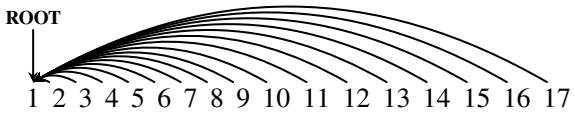
[6] An analysis in section 4.3 shows the WQ-trained model focuses on lexical items strongly related to the article topic.

	Yelp	Debates	WQ	WQTC	WSJSO
tree height	2.049	2.751	2.909	4.035	2.288
prop. of leaf nodes	0.825	0.849	0.958	0.931	0.892
norm. arc length	0.433	0.397	0.420	0.396	0.426
% vacuous trees	73%	38%	42%	14%	100%

Table 2: Statistics for learned trees averaged across four runs (similar results without the design change or bug fix are in the Appendix Table 6). See Table 4 for gold statistics on WQTC.

without the design change or bug fix to confirm our findings are similar (see L&L(orig) in Table 1).

Document-level structured attention does not help. Structured attention at the sentence level helps performance for all except WQ, where no form of attention helps. However, structured attention at the document level yields mostly negative results, in contrast to the improvements reported in L&L. In Yelp, WSJSO, and WQ, there is no difference. In Debates, the attention hurts performance. Only in WQTC does the structured attention provide a benefit. While a single training run could produce the improvements seen in L&L, the results across four runs depict a more accurate picture. When inducing structures, it is particularly important to repeat experiments as the structures can be highly inconsistent due to the noise caused by random initialization (Williams et al., 2018).



(1)**madam speaker, i rise in opposition to h.r. 3283 on both process and policy grounds.**. . . (17)look beyond the majority's smoke and mirrors, and vote against this ill-timed and ill-conceived legislation.

Figure 2: Learned dependency tree from Debates.

Trees do not learn discourse. Although document level structured attention provides little benefit in performance, we probe whether the model could still be learning some discourse. We visually inspect the learned $\mathbf{f_{best}}$ trees and in Table 2 we report statistics on them (see Appendix Table 6 for similar results with the original code).

The visual inspection (Figure 2) reveals shallow trees (also reported in L&L), but furthermore the trees have little to no structure.[7] We observe an interesting pattern where the model picks one of the first two or last two sentences as the root, and all other sentences are children of that node. We label these trees as 'vacuous' and the strength of this pattern is reflected in the tree statistics (Table 2). The height of trees is small, showing the trees are shallow. The proportion of leaf nodes is high, that is, most nodes have no children. Finally, the normalized arc length is high, where nodes that are halfway into the document still connect to the root.

[7]While shallow trees are expected in PDTB-style discourse, even these trees would exhibit meaningful structure between adjacent sentences, which is entirely absent here.

Yelp	uuu, sterne, star, rating, deduct, 0, edit, underwhelmed, update, allgemein
Debates	oppose, republican, majority, thank, gentleman, leadership, california, measure, president, vote
WQ	valley, mp3, firm, capital, universal, venture, silicon, analyst, capitalist, street

Table 3: Top 10 words most associated with the root sentence (measured with PPMI).

We further probe the root sentence, as the model places so much attention on it. We hypothesize the root sentence has strong lexical cues for the task, suggesting the model is instead attending to particular words. In Yelp, reviewers often start or end with a sentiment-laden sentence summarizing their rating. In Debates, speakers begin or end their speech by stating their stance on the bill. In WQ and WQTC, the interpretation of the root is less clear. In WSJSO, we find the root is always the first sentence of the correctly ordered document, which is reasonable and commonly attested in a discourse tree, but the remainder of the vacuous tree is entirely implausible.

To confirm our suspicion that the root sentence is lexically marked, we measure the association between words appearing in the root sentence and those elsewhere by calculating their positive pointwise mutual information scores (Table 3).

In Yelp, we find root words often express sentiment and explicitly mention the number of stars given ('sterne' in German, or 'uuu' as coined by a particularly prolific Yelper), which are clear indicators of the rating label. For Debates, words express speaker opinion, politeness and stance which are strong markers for the binary voting label. The list for WQ revolves around tech, suggesting the model is learning topics instead of writing quality. Thus, in WQTC we control for topics.

5 Learning better structure

We next probe whether the structure in L&L can be improved to be more linguistically appropriate, while still performing well on the end task. Given that structured attention helps only on WQTC and

	Acc	height	leaf	arc	vacuous
Full	**81.11**	4.035	0.931	0.396	14%
-biLSTM	77.80	11.51	0.769	0.353	4%
-biLSTM, +w	75.57	7.364	0.856	0.359	3%
-biLSTM, +p	77.11	10.430	0.790	0.349	3%
-biLSTM, +4p	**81.71**	9.588	0.811	0.353	3%
parsed RST	-	25.084	0.567	0.063	0%

Table 4: Mean test accuracy and tree statistics on the WQTC dev set (averaged across four runs). -biLSTM removes the document-level biLSTM, +w uses the weighted sum, +p performs 1 extra percolation, and +4p does 4 levels of percolation. The last row are ('gold') parsed RST discourse dependency trees.

learns vacuous trees less frequently, we focus on this task. We experiment with three modifications. First, we remove the document-level biLSTM since it performs a level of composition that might prevent the attention from learning the true structure. Second, we note equation 3 captures possible children only at one level of the tree, but not possible subtrees. We thus perform an additional level of percolation over the marginals to incorporate the children's children of the tree. That is, after equation 4, we calculate:

$$c'_i = \sum_{k=1}^{n} a_{ik} e'_i; \qquad e''_i = \tanh(W_r[e'_i, c'_i]) \quad (5)$$

Third, the max-pooling layer gives the model a way of aggregating over sentences while ignoring the learned structure. Instead, we propose a sum that is weighted by the probability of a given sentence being the root, i.e., using the learned root attention score a^r_i: $y_i = \sum_{i=1}^{n} a^r_i e''_i$.

We include ablations of these modifications and additionally derive RST discourse dependency trees,[8] collapsing intrasentential nodes, as an approximation to the ground truth.

The results (Table 4) show that simply removing the biLSTM produces trees with more structure (deeper trees, fewer leaf nodes, shorter arc lengths, and less vacuous trees), confirming our intuition that it was doing the work for the structured attention. However, it also results in lower performance. Changing the pooling layer from max to weighted sum both hurts performance and results in shallower trees (though still deeper than Full), which we attribute to this layer still being a pooling function. Introducing an extra level of tree percolation yields better trees but also a drop in performance. Finally, using 4 levels of percolation both reaches the accuracy of Full and retains the more structured trees.[9] We hypothesize accuracy doesn't surpass Full because this change also introduces extra parameters for the model to learn.

While our results are a step in the right direction, the structures are decidedly not discourse when compared to the parsed RST dependency trees, which are far deeper with far fewer leaf nodes, shorter arcs and no vacuous trees. Importantly, the tree statistics show the structures do not follow the typical right-branching structure in news: the trees are shallow, nodes often connect to the root instead of a more immediate parent, and the vast majority of nodes have no children. In work concurrent to ours, Liu et al. (2019) proposes a new iterative algorithm for the structured attention (in the same spirit as our extra percolations) and applies it to a transformer-based summarization model. However, even these induced trees are not comparable to RST discourse trees. The induced trees are multi-rooted by design (each root is a summary sentence) which is unusual for RST;[10] their reported tree height and edge agreement with RST trees are low.

6 Conclusion

In this paper, we evaluate structured attention in document representations as a proxy for discourse structure. We first find structured attention at the document level is largely unhelpful, and second it instead captures lexical cues resulting in vacuous trees with little structure. We propose several principled changes to induce better structures with comparable performance. Nevertheless, calculating statistics on these trees and comparing them to parsed RST trees shows they still contain no meaningful discourse structure. We theorize some amount of supervision, such as using ground-truth discourse trees, is needed for guiding and constraining the tree induction.

Acknowledgments

We thank the reviewers for insightful feedback. We acknowledge the Texas Advanced Computing Center for grid resources. The first author was supported by the NSF Graduate Research Fellowship Program under Grant No. 2017247409.

[8] We use the RST parser in Feng and Hirst (2014) and follow Hirao et al. (2013) to derive discourse dependency trees.

[9] More than 4 levels caused training to become unstable.

[10] Less than 25% of trees in the RST Discourse Treebank (Carlson et al., 2001) have more than 1 root; less than 8% have more than 2 roots.

References

Regina Barzilay and Mirella Lapata. 2008. Modeling Local Coherence - An Entity-Based Approach. *Computational Linguistics*, 34(1):1–34.

Yonatan Bisk and Ke Tran. 2018. Inducing grammars with and for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 25–35. Association for Computational Linguistics.

Lynn Carlson, Daniel Marcu, and Mary Ellen Okurovsky. 2001. Building a discourse-tagged corpus in the framework of rhetorical structure theory. In *Proceedings of the Second SIGdial Workshop on Discourse and Dialogue*, pages 1–10. Association for Computational Linguistics.

Jihun Choi, Kang Min Yoo, and Sang-goo Lee. 2018. Learning to compose task-specific tree structures. In *Proceedings of the 2018 Association for the Advancement of Artificial Intelligence (AAAI)*.

Y.J. Chu and T.H. Liu. 1965. On the shortest arborescence of a directed graph. *Science Sinica*, 14:1396–1400.

Greg Durrett, Taylor Berg-Kirkpatrick, and Dan Klein. 2016. Learning-based single-document summarization with compression and anaphoricity constraints. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1998–2008. Association for Computational Linguistics.

Jack Edmonds. 1967. Optimum branchings. *Journal of Research of the National Bureau of Standards*, 71:233–240.

Vanessa Wei Feng and Graeme Hirst. 2014. A linear-time bottom-up discourse parser with constraints and post-editing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 511–521. Association for Computational Linguistics.

Tsutomu Hirao, Yasuhisa Yoshida, Masaaki Nishino, Norihito Yasuda, and Masaaki Nagata. 2013. Single-document summarization as a tree knapsack problem. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1515–1520.

Yangfeng Ji and Noah A. Smith. 2017. Neural Discourse Structure for Text Categorization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 996–1005. Association for Computational Linguistics.

Shafiq Joty, Francisco Guzmán, Lluís Màrquez, and Preslav Nakov. 2017. Discourse structure in machine translation evaluation. *Computational Linguistics*, 43(4):683–722.

Terry Koo, Amir Globerson, Xavier Carreras, and Michael Collins. 2007. Structured prediction models via the matrix-tree theorem. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*.

Alex Lascarides and Nicholas Asher. 2008. Segmented discourse representation theory: Dynamic semantics with discourse structure. In *Computing meaning*, pages 87–124. Springer.

Yang Liu and Mirella Lapata. 2018. "Learning Structured Text Representations". *Transactions of the Association for Computational Linguistics*, 6:63–75.

Yang Liu, Ivan Titov, and Mirella Lapata. 2019. Single document summarization as tree induction. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1745–1755, Minneapolis, Minnesota. Association for Computational Linguistics.

Annie Louis and Ani Nenkova. 2013. What Makes Writing Great? First Experiments on Article Quality Prediction in the Science Journalism Domain. *TACL*.

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

Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational linguistics*, 19(2):313–330.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543. Association for Computational Linguistics.

Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind K Joshi, and Bonnie L Webber. 2008. The Penn Discourse TreeBank 2.0. In *LREC*.

Evan Sandhaus. 2008. The New York Times annotated corpus. *Linguistic Data Consortium, Philadelphia*, 6(12):e26752.

Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. 2018. Linguistically-informed self-attention for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5027–5038. Association for Computational Linguistics.

Duyu Tang, Bing Qin, and Ting Liu. 2015. "Learning Semantic Representations of Users and Products for Document Level Sentiment Classification". In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1014–1023. Association for Computational Linguistics.

Matt Thomas, Bo Pang, and Lillian Lee. 2006. "Get out the vote: Determining support or opposition from Congressional floor-debate transcripts". In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 327–335. Association for Computational Linguistics.

Adina Williams, Andrew Drozdov, and Samuel R. Bowman. 2018. Do latent tree learning models identify meaningful structure in sentences? *Transactions of the Association for Computational Linguistics*, 6:253–267.

Dani Yogatama, Phil Blunsom, Chris Dyer, Edward Grefenstette, and Wang Ling. 2017. Learning to compose words into sentences with reinforcement learning. *ICLR*.

Dani Yogatama and Noah A. Smith. 2014. "Linguistic Structured Sparsity in Text Categorization". In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 786–796. Association for Computational Linguistics.

A Appendices

Datasets Statistics for the datasets are listed in Table 5.

For WQ, the very good class was created by Louis and Nenkova (2013) using as a seed the 63 articles in the New York Times corpus (Sandhaus, 2008) deemed to be high-quality writing by a team of expert journalists. The class was then expanded by adding all other science articles in the NYT corpus that were written by the seed authors (4,253 articles). For the typical class, science articles by all other authors were included (19,520). Because the data is very imbalanced, we undersample the typical class to be the same size as the very good. We split this data into 80/10/10 for training, development and test, with both classes equally represented in each partition.

For WQTC, the original dataset authors provide a list of the 10 most topically similar articles for each article.[11] We make use of this list to explicitly sample topically similar documents.

[11] http://www.cis.upenn.edu/~nlp/corpora/scinewscorpus.html

Preprocessing For Debates and Yelp, we follow the same preprocessing steps as in L&L, but do not set a minimum frequency threshold when creating the word embeddings. For our three datasets, sentences are split and tokenized using Stanford Core NLP.

Training For all models, we use the Adagrad optimizer with a learning rate of 0.05. For WQ, WQTC, and WSJSO, gradient clipping is performed using the global norm with a ratio of 1.0. The batch size is 32 for all models except WSJSO uses 16. All models are trained for a maximum of 8 hours on a GeForce GTX 1080 Ti card.

Results Because our results hinge on multiple runs of experiments each initialized with different random weights, we include here more detailed versions of our results to more accurately illustrate their variability. Table 6 supplements Table 2 with tree statistics from L&L(orig), the model *without* the design change or bug fix, to illustrate the derived trees on this model are similar. Finally, Table 7 is a more detailed version of Table 4, which additionally includes maximum accuracy, standard deviation for accuracy, as well as the average parent entropy calculated over the latent trees.

Dataset	Classes	Number of documents				Vocab.
		Total	Train	Dev	Test	
Yelp	5	333K	266,522	33,333	33,317	53K
Debates	2	1.5K	1,050	102	374	21K
WQ	2	7.7K	6,195	775	763	150K
WQTC	2	7.8K	6,241	777	794	131K
WSJSO	-	2.4K	1,950 (35,165)	247 (4,392)	241 (4,383)	49K

Table 5: Statistics for the datasets used in the text classification and discrimination tasks (calculated after preprocessing). For WSJSO, the number of generated pairs are in parentheses.

	Yelp	Debates	WQ	WQTC	WSJSO
tree height	2.049 (2.248)	2.751 (2.444)	2.909 (2.300)	4.035 (2.468)	2.288 (2.368)
prop. of leaf nodes	0.825 (0.801)	0.849 (0.869)	0.958 (0.971)	0.931 (0.966)	0.892 (0.888)
norm. arc length	0.433 (0.468)	0.397 (0.377)	0.420 (0.377)	0.396 (0.391)	0.426 (0.374)
% vacuous trees	73% (68%)	38% (40%)	42% (28%)	14% (21%)	100% (56%)

Table 6: Statistics for the learned trees averaged across four runs on the L&L(ours) model with comparisons (in parentheses) to results using the original L&L code without the design change or bug fix.

	Accuracy	tree height	prop. of leaf	parent entr.	norm. arc length	% vacuous trees
Full	82.49 | **81.11** (0.95)	4.035	0.931	0.774	0.396	14%
-biLSTM	80.35 | 77.80 (1.72)	11.51	0.769	1.876	0.353	4%
-biLSTM, +p	78.72 | 77.11 (2.18)	10.430	0.790	0.349	0.349	3%
-biLSTM, +4p	82.75 | **81.71** (0.70)	9.588	0.811	1.60	0.353	3%
-biLSTM, +w	78.46 | 75.57 (2.52)	7.364	0.856	1.307	0.359	3%
-biLSTM, +w, +p	77.08 | 74.78 (2.58)	8.747	0.826	1.519	0.349	4%
parsed RST	-	25.084	0.567	2.711	0.063	0%

Table 7: Max | mean (standard deviation) test accuracy and tree statistics of the WQTC dev set (averaged across four training runs with different initialization weights). Bolded numbers are within 1 standard deviation of the best performing model. +w uses the weighted sum, +p adds 1 extra level of percolation, +4p adds 4 levels of percolation. The last row are the ('gold') parsed RST discourse dependency trees.