Multi-Task Learning for Coherence Modeling

Youmna Farag **Helen Yannakoudakis**

Department of Computer Science and Technology
The ALTA Institute
University of Cambridge
United Kingdom
{youmna.farag,helen.yannakoudakis}@cl.cam.ac.uk

Abstract

We address the task of assessing discourse coherence, an aspect of text quality that is essential for many NLP tasks, such as summarization and language assessment. We propose a hierarchical neural network trained in a multi-task fashion that learns to predict a document-level coherence score (at the network's top layers) along with word-level grammatical roles (at the bottom layers), taking advantage of inductive transfer between the two tasks. We assess the extent to which our framework generalizes to different domains and prediction tasks, and demonstrate its effectiveness not only on standard binary evaluation coherence tasks, but also on real-world tasks involving the prediction of varying degrees of coherence, achieving a new state of the art.

1 Introduction

Discourse coherence refers to the way textual units relate to one another and form a coherent whole. Coherence is an important aspect of text quality and therefore its modeling is essential in many NLP applications, including summarization (Barzilay et al., 2002; Parveen et al., 2016), question-answering (Verberne et al., 2007), question generation (Desai et al., 2018), and language assessment (Burstein et al., 2010; Somasundaran et al., 2014; Farag et al., 2018). A large body of work has investigated models for the assessment of inter-sentential coherence, that is, assessment in terms of transitions between adjacent sentences (Barzilay and Lapata, 2008; Yannakoudakis and Briscoe, 2012; Guinaudeau and Strube, 2013; Tien Nguyen and Joty, 2017; Joty et al., 2018). The properties of text that result in inter-sentential connectedness have been translated into a number of computational models – some of the most prominent ones include the entity-based approaches, inspired by Centering Theory (Grosz et al., 1995) and proposed in the pioneering work of Barzilay and Lapata (2005, 2008). Such approaches model local coherence in terms of entity transitions between adjacent sentences, where entities are represented by their syntactic role in the sentence (e.g., subject, object).

Current state-of-the-art deep learning adaptations of the entity-based framework involve the use of Convolutional Neural Networks (CNNs) over an entity-based representation of text to discriminate between a coherent document and its incoherent variants containing a random reordering of the document's sentences (Tien Nguyen and Joty, 2017); as well as lexicalized counterparts of such models that further incorporate lexical information regarding the entities, thereby distinguishing between different entities (Joty et al., 2018).

In contrast to existing approaches, we propose a more generalized framework that allows neural models to encode information about the types of grammatical roles all words in a sentence participate in, rather than focusing only on the roles of entities within a sentence. Inspired by recent advances in Multi-Task Learning (MTL) (Rei and Yannakoudakis, 2017; Sanh et al., 2018), we propose a simple, yet effective hierarchical model trained in a multi-task fashion that learns to perform two tasks: scoring a document's discourse coherence and predicting the type of grammatical role (GR) of a dependent with its head. We take advantage of inductive transfer between these tasks by giving a supervision signal at the bottom layers of a network with respect to the types of GRs, and a supervision signal at the top layers with respect to document-level coherence.

Our contributions are four-fold: (1) We propose a MTL approach to coherence assessment and compare it against a number of baselines. We experimentally demonstrate that such a framework allows us to exploit more effectively the inter-

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 629–639
Florence, Italy, July 28 - August 2, 2019. ©2019 Association for Computational Linguistics

dependencies between the two prediction tasks and achieve state-of-the-art results in predicting document-level coherence; (2) We assess the extent to which the information encoded in the network generalizes to different domains and prediction tasks, and demonstrate the effectiveness of our approach not only on standard binary evaluation tasks on the Wall Street Journal (WSJ), but also on more realistic tasks involving the prediction of varying degrees of coherence in people's everyday writing; (3) In contrast to existing work that has only investigated the impact of a specific set of grammatical roles (i.e., subject and object) on coherence, we instead investigate a large set of GR types, and train the model to predict the type of role dependents participate in. This allows the network to learn more generic patterns of language and composition, and a much richer set of representations than those induced by current approaches. In turn, this can be better exploited at the top layers of the network for predicting document-level coherence; (4) Finally, and contrary to previous work, our model does not rely on the availability of external linguistic tools at testing time as it directly learns to predict the GR types.

2 Related Work

Several studies have proposed frameworks for modeling the textual properties that coherent texts exhibit. A popular approach is one based on the entity-grid (egrid) representation of texts, proposed by Barzilay and Lapata (2005, 2008) and inspired by Centering Theory (Grosz et al., 1995). In the egrid model, texts are represented as matrices of entities (columns) and sentences (rows). Entities in the matrix are represented by their grammatical role (i.e., subject, object, neither), and entity transitions across sentences are used as features for coherence assessment. A large body of work has utilized and extended the egrid approach (Elsner and Charniak, 2008; Burstein et al., 2010; Elsner and Charniak, 2011; Guinaudeau and Strube, 2013). Other features have also been leveraged, such as syntactic patterns (Louis and Nenkova, 2012) and discourse relations (Lin et al., 2011; Feng et al., 2014). Deep learning architectures have also been successfully applied to the task of coherence scoring, achieving state-of-the-art results (Li and Jurafsky, 2017; Logeswaran et al., 2018; Cui et al., 2018). Some have exploited egrid features in a CNN model aimed at capturing long range entity transitions (Tien Nguyen and Joty, 2017; Joty et al., 2018); further details are provided in Section 4.2.

Traditionally, coherence evaluation has been treated as a binary task, where a model is trained to distinguish between a coherent document and its incoherent counterparts created by randomly shuffling the sentences it contains. The news domain has been a popular source of well-written, coherent texts. Among the popular datasets are articles about `EARTHQUAKES` and `AIRPLANES` accidents (Barzilay and Lapata, 2008; Guinaudeau and Strube, 2013; Li and Jurafsky, 2017) and the Wall Street Journal (WSJ) portion of the Penn Treebank (Elsner and Charniak, 2008; Lin et al., 2011; Tien Nguyen and Joty, 2017). Elsner and Charniak (2008) argue that the WSJ documents are normal informative articles, whereas the `AIRPLANES` and `EARTHQUAKES` ones have a more constrained style.

3 Approach

3.1 Neural Single-Task Learning (STL)

Our baseline model, shown in Figure 1, performs the single task of predicting an overall coherence score via a hierarchical model based on a Long Short-Term Memory (LSTM) network (Hochreiter and Schmidhuber, 1997). A document is composed of a sequence of sentences $\{s_1, s_2, ..., s_m\}$ and, in turn, each sentence consists of a sequence of words $\{w_1, w_2, ..., w_n\}$. The input words are initialized with vectors from a pre-trained embedding space. A bidirectional LSTM (Bi-LSTM) is applied to the words in each sentence to get contextualized representations, and the output vectors from both directions are concatenated:

$$\begin{aligned} \overrightarrow{h_t^w} &= LSTM(w_t, \overrightarrow{h_{t-1}^w}) \\ \overleftarrow{h_t^w} &= LSTM(w_t, \overleftarrow{h_{t+1}^w}) \\ h_t^w &= [\overrightarrow{h_t^w}, \overleftarrow{h_t^w}] \end{aligned} \tag{1}$$

To compose a sentence representation s, the hidden states $\{h_1^w, ..., h_n^w\}$ of its words are combined with an attention mechanism:

$$\begin{aligned} u_t^w &= tanh(W^w h_t^w) \\ a_t^w &= \frac{exp(v^w u_t^w)}{\sum_t exp(v^w u_t^w)} \\ s &= \sum_t a_t^w h_t^w \end{aligned} \tag{2}$$

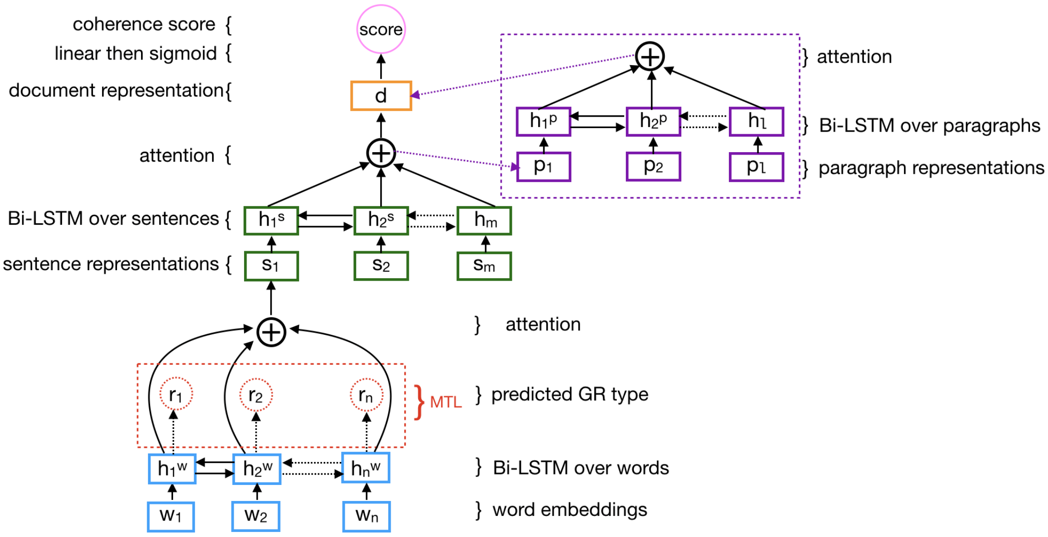


Figure 1: The hierarchical architecture of the STL and MTL models. The dotted red box is specific to the MTL framework. The dotted purple box is applied if the document contains paragraph boundaries (which is the case for the Grammarly Corpus in Section 4.1) in order to create paragraph representations prior to the document one.

where W^w and v^w are learnable parameters. Attention allows the model to focus on the salient words for coherence and build better sentence representations.

Constructing a document representation d is similar to the sentence one – a second Bi-LSTM is utilized over sentences $\{s_1, s_2, ..., s_m\}$ to generate contextually rich sentence representations:

$$\begin{aligned}\overrightarrow{h_i^s} &= LSTM(s_i, \overrightarrow{h_{i-1}^s}) \\ \overleftarrow{h_i^s} &= LSTM(s_i, \overleftarrow{h_{i+1}^s}) \\ h_i^s &= [\overrightarrow{h_i^s}, \overleftarrow{h_i^s}]\end{aligned} \tag{3}$$

Subsequently, attention is applied over the sentence embeddings $\{h_1^s, ..., h_m^s\}$ to allow the model to focus on sentences that contribute highly to the overall coherence of the document:

$$\begin{aligned}u_i^s &= tanh(W^s h_i^s) \\ a_i^s &= \frac{exp(v^s u_i^s)}{\sum_i exp(v^s u_i^s)} \\ d &= \sum_i a_t^s h_i^s\end{aligned} \tag{4}$$

where W^s and v^s are trainable weights in the network. If a document consists of paragraphs $\{p_1, p_2, ..., p_l\}$, a third Bi-LSTM is stacked over the sentence vectors and the output is aggregated with another attention layer to compose the document vector d.

Finally, the coherence score of a document is predicted by applying a linear transformation to the vector d followed by a sigmoid operation to bound the score in $[0, 1]$:

$$\hat{y} = \sigma(W^d \ d) \tag{5}$$

where $W^d \in \mathbb{R}^{dim}$ is the linear function weight and dim represents the dimensionality of the document vector. In a binary classification task, where the document is labeled as either coherent or incoherent, the model predicts one value for $\hat{y} \in [0, 1]$. In a multiclass classification setting where there are multiple classes $y \in C$ representing various degrees of coherence, a document is labeled with a one-hot vector with length $|C|$ with a value of 1 in the index of the correct class and 0 everywhere else. The model predicts $|C|$ scores, using Equation 5 with $W^d \in \mathbb{R}^{dim \times |C|}$, and learns to maximize the value corresponding to the gold label.

For the binary task, the network's parameters are optimized to minimize the negative log-likelihood of the document's ground-truth label y, given the networks prediction $\hat{y}$:

$$L_1 = -y \ log(\hat{y}) - (1 - y)log(1 - \hat{y}) \tag{6}$$

For the multiclass task, we use mean squared error to minimize the discrepancy between the one-hot gold vector and the estimated one:

$$L_1 = \frac{1}{|C|} \sum_{j=1}^{|C|} (y_j - \hat{y_j})^2 \tag{7}$$

An alternative approach to the multiclass problem is to apply a softmax over the predictions instead of a sigmoid, and minimize the categorical cross entropy; however, initial experiments on the development set showed that our formation yields better results.

3.2 Neural Multi-Task Learning (MTL)

The model described in 3.1 performs the single task of predicting a coherence score for a text; all model parameters are tuned to minimize the loss (L_1) in Equation 6 or 7 (depending on whether we are optimizing for a binary or a multiclass classification task respectively). We extend this model to a MTL framework by training it to optimize a secondary objective at the bottom layers of the network, along with the main one (L_1). Specifically, the model is trained to predict a document-level score along with word-level labels indicating the (predicted) GR type of dependents in the document.[1] The GRs are based on a predefined set R, generated from a dependency parser on the training set (Section 4.3). The set includes the types of GRs in which a word is a dependent (e.g., *nsubj*, *amod*, *xcomp*, *iobj*), and each type $r \in R$ is treated as a class (for the 'root' word, the type is *root*). In order to predict a probability distribution over R given a word representation h_t (Equation 1), a linear operation normalized by a softmax function is applied:

$$P(y_t^r|h_t^w) = softmax(W^r h_t^w) \tag{8}$$

The secondary objective and the word-level loss is defined as the categorical cross-entropy, i.e., the negative log-probability of the correct labels:

$$L_2 = -\sum_t \sum_r y_t^r log P(y_t^r|h_t^w) \tag{9}$$

Both the main (L_1) and secondary (L_2) objectives are optimized jointly (L_{total}), but with different weights to indicate the importance of each of these tasks during training:

$$L_{total} = \alpha L_1 + \beta L_2 \tag{10}$$

where $\alpha, \beta \in [0, 1]$ are the loss weight hyperparameters. Figure 1 (red-dotted box) presents the complete MTL framework. MTL allows us to take advantage of inductive transfer between these tasks and learn a rich set of representations at the bottom layers that can be exploited by the top layers of the network for predicting a document-level coherence score.

	#Docs	#Synthetic Docs	Avg #Sents
Train	1,376	25,767	21.0
Test	1,090	20,766	21.9

Table 1: Statistics for the WSJ data. #Docs represents the number of original articles and #Synthetic Docs the number of original articles + their permuted versions.

		#Docs	Avg #Sents
Yahoo	Train	1000	7.5
	Test	200	7.5
Clinton	Train	1000	6.6
	Test	200	6.6
Enron	Train	1000	7.7
	Test	200	7.8

Table 2: Statistics for the GCDC.

Current state-of-the-art approaches utilizing the entity-based framework (Joty et al., 2018) focus solely on the subject and object types. To further assess the impact of our extended set of GR types, we re-train the same MTL model but now only utilize subject (S) and object (O) GR types as our secondary training signal. Following the current entity-based approaches, all other types are mapped to X, to represent 'other' roles; specifically, $R = \{S, O, X\}$. We refer to this baseline model as MTL$_{\text{sox}}$.

4 Experiments

4.1 Data and Evaluation Metrics

Synthetic Data. The Wall Street Journal (WSJ) portion of the Penn Treebank (Elsner and Charniak, 2008; Lin et al., 2011; Tien Nguyen and Joty, 2017) is one of the most popular datasets for (binary) coherence assessment, given its size and the nature of the texts it contains; i.e. long articles not constrained in style (Elsner and Charniak, 2008; Tien Nguyen and Joty, 2017). Following previous work (Tien Nguyen and Joty, 2017), we also use the WSJ and specifically sections $00 - 13$ for training and $14 - 24$ for testing (documents consisting of one sentence are removed). We create 20 permutations per document, making sure to exclude duplicates or versions that happen to have the same ordering of sentences as the original article. Table 1 presents the data statistics.

To evaluate model performance on this dataset,

[1] We make our code publicly available at https://github.com/Youmna-H/coherence_mtl

we again follow previous work (Barzilay and Lapata, 2008; Tien Nguyen and Joty, 2017) and calculate pairwise ranking accuracy (PRA) between an original text and its 20 permuted counterparts. Specifically, PRA calculates the fraction of correct pairwise rankings in the test data (i.e., a coherent/original text should be ranked higher than its permuted counterpart). Following Farag et al. (2018), we also report the total pairwise ranking accuracy (TPRA) that extends PRA to comparing each original text to all permuted texts in the test set rather than only its own set of permuted counterparts.

Realistic Data. The Grammarly Corpus of Discourse Coherence (GCDC) is a newly-released dataset containing emails and reviews written with varying degrees of proficiency and care (Lai and Tetreault, 2018).[2] In addition to the WSJ, we employ this dataset in order to assess the effectiveness of our coherence model for tasks involving the prediction of varying degrees of coherence in people's everyday writing. Specifically, the dataset contains texts from four domains: **Yahoo** online forum posts, emails from Hillary **Clinton**'s office, emails from **Enron** and **Yelp** business reviews. As some of the reviews from the latter were subsequently removed by Yelp, we evaluate our model on each of the first three domains (Table 2).

Annotators were instructed to rate each document with a score $\in \{1, 2, 3\}$, representing *low*, *medium* and *high* levels of coherence respectively. For our experiments, we use the consensus rating of the expert scores as calculated by Lai and Tetreault (2018), and train the models to maximize the probability of the gold class within a multi-class classification framework (see Section 3). The gold label distribution is as follows: Yahoo 44.8% low, 17.9% medium, 37.25% high; Clinton 27.8% low, 20.3% medium, 51.8% high; Enron 30% low, 20.3% medium, 49.6% high. To evaluate model performance, we use three-way classification accuracy.

4.2 Models and Baselines

CNN Egrid (Egrid CNN$_{\text{ext}}$). We replicate the model proposed by Tien Nguyen and Joty (2017) using their source code.[3] The authors generate entity-grid representations of texts (i.e., matrices of entities as columns and sentences as rows, where entities are represented by their syntactic role: subject, object, or other) using the Brown coherence toolkit.[4] They then employ a CNN over the entity transitions across sentences in order to capture high-level features and long-range transitions. Training is performed in a pairwise fashion where the model learns to rank a coherent document higher than its incoherent counterparts. To further improve performance, they extend the model by including three entity-specific features, attached to entities' distributed representations: named entity type, salience (represented as the occurrence frequency of entities) and a binary feature indicating whether the entity has a proper mention.

Lexicalized CNN Egrid (Egrid CNN$_{\text{lex}}$). The aforementioned Egrid CNN model is agnostic to entities' lexical properties, which are useful features for the task. To remedy this, Joty et al. (2018) further extend it with lexical information about the entities: they represent each entity with its lexical presentation and attach it to its syntactic role (S, O, X). For instance, if "Obama" appears as a subject and an object, there will be two different representations for it in the input embedding matrix: Obama-S and Obama-O. Joty et al. (2018) achieve state-of-the-art results on the WSJ, outperforming Egrid CNN$_{\text{ext}}$ without including the three entity-specific features in their model. We also replicate their model using the authors' source code.[5]

Local Coherence Model (LC). This model, initially proposed by Li and Hovy (2014), applies a window approach to assess a text's local coherence. Sentences are encoded with a recurrent or recursive layer and a filter of weights is applied over each window of sentence vectors to extract "clique" scores that are aggregated to calculate the overall document coherence score. We use an improved variant that captures sentence representations via an LSTM and predicts an overall coherence score by averaging the local clique scores (Li and Jurafsky, 2017; Farag et al., 2018). Lai and Tetreault (2018) recently showed that the LC model achieves state-of-the-art results on the Clinton and Enron datasets.

[2] https://github.com/aylai/GCDC-corpus

[3] https://github.com/datienguyen/cnn_coherence

[4] https://bitbucket.org/melsner/browncoherence

[5] https://ntunlpsg.github.io/project/coherence/n-coh-acl18/

Paragraph sequence (PARSEQ). Lai and Tetreault (2018) implemented a hierarchical neural network consisting of three LSTMs to generate sentence, paragraph and document representations. The network's architecture is similar to our STL model; the key difference is the attention mechanism we use for aggregation. The model was tested on the GCDC and was found to outperform other feature-engineered methods and give state-of-the-art results on the Yahoo dataset.

Neural Single-Task Learning (STL). We implement the STL model as described in 3.1. For the WSJ data, the network utilizes two Bi-LSTMs to compose sentence and document representations. For the GCDC, we add a third Bi-LSTM, where sentence representations are aggregated via attention to form paragraph vectors. Given these paragraph vectors, we then apply a Bi-LSTM followed by attention to compose the document vectors that are to be scored for coherence.

Neural Multi-Task Learning (MTL). We implement the MTL model as described in 3.2. The same architecture variants as the STL ones are applied on the different datasets.

Neural S-O-X Multi-Task Learning ($\mathbf{MTL_{SOX}}$). As discussed in 3.2, we create another version of the MTL model where, for each word, we only predict subject (S), object (O) and 'other' (X) roles.

GR types Concatenation Model ($\mathbf{Concat_{grs}}$). Instead of learning to predict the GR types within a MTL framework, we incorporate them as input features to the model by concatenating them to the word representations in the STL framework. In this setup, we randomly initialize the types embedding matrix $E_{gr} \in \mathbb{R}^{q \times g}$, where g is the embedding size and q is the number of GR types in the training data. Each type is then mapped to a row in E_{gr} and concatenated to its corresponding word at the model's input layer. Here, the GRs are needed as input at both training and test time, unlike the MTL framework that only requires them during training. The concat$_{\text{grs}}$ model allows us to further assess whether the MTL framework has an advantage over feeding the GR types as input features.

4.3 Experimental setup

We extract the GR types of words using the Stanford Dependency Parser (v. 3.8) (Chen and Manning, 2014) and obtain a total of 39 different types of Universal Dependencies and their subtypes (see Appendix A for the full list). For the MTL$_{\text{SOX}}$ model, we consider direct objects, indirect objects and subjects of passive verbs as objects (O). Our models are initialized with pre-trained GloVe embeddings (Pennington et al., 2014). We use mini-batches of size 32, optimize the models using RMSProp (Tieleman and Hinton, 2012), and set the learning rate to 0.001. Dropout (Srivastava et al., 2014) is used for regularization with probability 0.5 and applied to the word embedding layer and the output of the Bi-LSTM sentence layer. Table 3 shows the different hyperparameters used for training.[6]

	word embed dim	LSTM hidden dim			α	β
		h^w	h^s	h^p		
WSJ	50	100	100	-	0.7	0.3
Yahoo	300	100	100	100	1	0.1
Clinton	300	100	200	100	1	0.1
Enron	300	100	100	100	1	0.2

Table 3: Model hypermarameters: w, s and p refer to word, sentence and paragraph hidden layers respectively; α is the main and β the secondary loss weight.

Training is done for 30 epochs and performance is monitored over the development set; the model with the highest performance (highest PRA on the synthetic data and highest classification accuracy on GCDC) on the development set is selected and applied at testing time. To reduce model variance, we run the WSJ experiments 5 times with different random initializations and the GCDC ones 10 times (following Lai and Tetreault (2018)), and average the predicted scores of the ensembles for the final evaluation. For the WSJ data, we use the same train/dev splits as Tien Nguyen and Joty (2017), and for GCDC, we follow Lai and Tetreault (2018) and split the training data with a 9:1 ratio for tuning.

5 Results and Discussion

Binary Classification. Table 4 shows the results of the binary discrimination task on the WSJ. The results demonstrate the effectiveness of our MTL approach using a supervision signal at the bottom layers based on the words' GR types, which significantly outperforms all other approaches and achieves state-of-the-art results on

[6] We note that hyperparameters are tuned per domain.

Model	PRA	TPRA
Egrid CNN$_{\text{ext}}$	0.876	0.656
Egrid CNN$_{\text{lex}}$	0.846	0.566
LC	0.741	0.728
STL	0.877	0.893
MTL	**0.932***	**0.941***
MTL$_{\text{SOX}}$	0.899	0.913
Concat$_{\text{grs}}$	0.896	0.908

Table 4: Results of the binary discrimination task on the WSJ. * indicates significance ($p < 0.01$) over all the other models based on the randomization test. Egrid models are significantly worse than MTL$_{\text{SOX}}$ and Concat$_{\text{grs}}$ on the PRA metric and significantly worse than all models on TPRA.[8]

Model	Yahoo	Clinton	Enron
LC	0.535	0.610	0.544
PARSEQ	0.549	0.602	0.532
STL	0.550	0.590	0.505
MTL	**0.560**	**0.620***	**0.560***
MTL$_{\text{SOX}}$	0.505	0.585	0.510
Concat$_{\text{grs}}$	0.455	0.570	0.460

Table 5: Model accuracy on the three-way classification task on GCDC. * indicates significance over STL with $p < 0.01$ using the randomization test. Results for PARSEQ and LC are those reported in Lai and Tetreault (2018) on the same data.

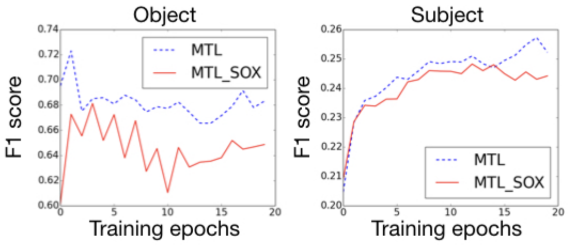


Figure 2: F1 scores for subject and object predictions with the MTL and MTL$_{\text{SOX}}$ models over the first 20 epochs of training. Y-axis: F1 scores; x-axis: epochs. The graphs are based on the WSJ dev set.

the WSJ (0.932 PRA and 0.941 TPRA).[7] The performance of the Egrid neural models shows that despite their ability to rank a document higher than its incoherent counterparts (0.876 and 0.846 PRA), they do not generalize when documents are compared against counterparts from the whole test set (0.656 and 0.566 TPRA). This could be partly attributed to the pairwise training strategy adopted by these models and their inability to compare entity-transition patterns across different topics. The table also shows that models that utilize compositions over textual units to form document representations (the last four models) are significantly more effective than those explicitly utilizing only the local transitions between sentences (LC model). Furthermore, we observe that incorporating GR types (MTL, MTL$_{\text{SOX}}$ and Concat$_{\text{grs}}$) gives significantly better results compared to the STL model that is GR-agnostic. The superiority of the MTL model over Concat$_{\text{grs}}$ and MTL$_{\text{SOX}}$ demonstrates that learning the GR types, within an MTL framework, allows the model to learn richer contextual representations (but also to be more efficient at testing time compared to e.g., Concat$_{\text{grs}}$ since it does not require external linguistic tools).

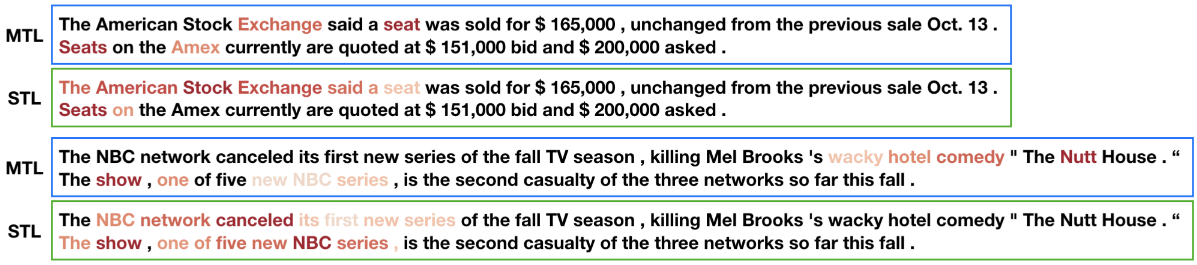
To further analyze performance, we calculate the Pearson correlation between: a) the similarity between a permuted document and its original counterpart in terms of the minimum number of adjacent transpositions needed to transform the former back to its original version (Lapata, 2006), and b) the predicted coherence score for the permuted document. This allows us to investigate whether a higher similarity is linked to a higher coherence score. We observe that MTL, MTL$_{\text{SOX}}$, Concat$_{\text{grs}}$ and STL have the highest correlations (0.260, 0.232, 0.227, 0.225 respectively), followed by LC (0.076), Egrid CNN$_{\text{ext}}$ (-0.0126) and Egrid CNN$_{\text{lex}}$ (-0.069).[9] In order to further analyze the strengths of MTL, we plot in Figure 2 the F1 scores over the training epochs for predicting the subject and object types using MTL or MTL$_{\text{SOX}}$. We can see that learning to predict a larger set of GR types enhances the model's predictive power for the subject and object types, corroborating the value of entity-based properties for coherence.

Three-way Classification. On GCDC (Table 5) we can see that MTL achieves state-of-the-art performance across all three datasets. Although different evaluation metrics are employed, we note that the numbers obtained on this dataset are quite low compared to those on the WSJ. Assessing

[7] Significance is calculated based on the randomization test (Yeh, 2000).

[8] Joty et al. (2018) reported 0.885 PRA for their Egrid CNN$_{\text{lex}}$, which we were unable to replicate using their code; however, this is still lower compared to our results.

[9] We note that the low correlation is due to the nature of the task: binary evaluation rather than absolute scoring of coherence.


MTL
STL
MTL
STL

MTL: The American Stock Exchange said a seat was sold for $ 165,000 , unchanged from the previous sale Oct. 13 . Seats on the Amex currently are quoted at $ 151,000 bid and $ 200,000 asked .

STL: The American Stock Exchange said a seat was sold for $ 165,000 , unchanged from the previous sale Oct. 13 . Seats on the Amex currently are quoted at $ 151,000 bid and $ 200,000 asked .

MTL: The NBC network canceled its first new series of the fall TV season , killing Mel Brooks 's wacky hotel comedy " The Nutt House . " The show , one of five new NBC series , is the second casualty of the three networks so far this fall .

STL: The NBC network canceled its first new series of the fall TV season , killing Mel Brooks 's wacky hotel comedy " The Nutt House . " The show , one of five new NBC series , is the second casualty of the three networks so far this fall .

Figure 3: Visualization of the model's gradients with respect to the input word embeddings for MTL and STL on the WSJ dev set. Words that contribute the most to coherence scoring (i.e., those with high gradient norms) are colored: the contribution of words decreases from dark red to lighter tones of orange.

varying degrees of coherence is a more challenging task: differences in coherence between different documents is less pronounced than when taking a document and randomly shuffling its sentences. When comparing MTL to STL, the former is consistently better across all datasets, with significant improvements for two of them.[10] Interestingly, we observe that MTL$_{\text{SOX}}$ and Concat$_{\text{grs}}$ do not generalize to the more realistic domain. As shown in Table 3, our best MTL model uses smaller β and higher α values on the GCDC compared to the WSJ. This could be attributed to the performance of the parser and/or the nature of the GCDC and the properties of (in)coherence it exhibits, compared to the WSJ data. MTL allows the model more flexibility and control with respect to the features it learns in order to enhance performance on the main task, in contrast to Concat$_{\text{grs}}$ where the GRs are given directly as input to the model (yielding the worst performance across all the GCDC datasets).

The results on GCDC demonstrate that our main MTL approach generalizes to tasks involving the prediction of varying degrees of coherence in everyday writing. In general, however, we observe that, out of the three gold coherence labels (low, medium, high) both MTL and STL have difficulty in correctly classifying documents of medium coherence, which can be attributed to the smaller number of training examples for that class (Section 4.1).

Visualization. In an attempt to better understand what the models have learnt, we visualize the words that contribute the most to coherence prediction. We calculate the model's gradients with respect to the input word embeddings (similarly to Li et al. (2016)) to determine which words maximize the model's prediction (more influential words should have higher gradient norms). Figure 3 presents example visualizations obtained with STL and MTL. We observe that for MTL, important words are those that are considered the center of attention: in the first example (top two sentences) where the document is about seats in the stock exchange, "seat" and "Seats" are considered more important than the subject entities. On the other hand, the STL model considers the subject of the first sentence ("The American Stock Exchange") more important than the object "seat". In the second example (last two sentences) where the document is about a canceled show by the NBC, for the MTL model, the name of the show (or part of it) in the first sentence ("Nutt") is considered important, as well as "comedy" which also refers to the show; in addition to "show" in the second sentence. On the other hand, STL fails to identify the name of the show as important. In general, STL seems to be more distracted, focusing on words that do not necessarily contribute to coherence (e.g., determiners and prepositions), whereas MTL seems to be considering more informative parts of the text.

Qualitative Analysis. Following previous work (Miltsakaki and Kukich, 2004; Li and Jurafsky, 2017), we perform a small-scale qualitative analysis: we apply our best model to a number of discourses that exhibit different types of coherence and investigate the predicted coherence scores. We observe that MTL can capture some aspects of lexical and centering/referential coherence:

Mary ate some apples. She likes apples. **0.790**
Mary ate some apples. She likes pears. **0.720**
Mary ate some apples. She likes Paris. **0.742**
She ate some apples. Mary likes apples. **0.747**

[10] We also note that GR prediction is only required during training; therefore, at inference time, MTL uses the same number of parameters as STL.

John went to his favorite music store to buy a piano. He had frequented the store for many years. **0.753**
John went to his favorite music store to buy a piano. It was a store John had frequented for many years. **0.743**

On the other hand, it is not as good at recognizing temporal order and causal relationships; for example:

Bret enjoys video games; therefore, he sometimes is late to appointments. **0.491**
Bret sometimes is late to appointments; therefore, he enjoys video games. **0.499**

6 Conclusion

We have presented a hierarchical multi-task learning framework for discourse coherence that takes advantage of inductive transfer between two tasks: predicting the GR type of words at the bottom layers of the network and predicting a document-level coherence score at the top layers. We assessed the extent to which our framework generalizes to different domains and prediction tasks, and demonstrated its effectiveness against a number of baselines not only on standard binary evaluation coherence tasks, but also on tasks involving the prediction of varying degrees of coherence, achieving a new state of the art. As part of future work, we would like to investigate the use of contextualized embeddings (e.g., BERT, Devlin et al. (2018)) for coherence assessment – as such representations have been shown to carry syntactic information of words (Tenney et al., 2019) – and whether they allow multi-task learning frameworks to learn complementary aspects of language.

Acknowledgments

We thank Ted Briscoe and Marek Rei for their valuable suggestions and feedback. We also thank Paula Buttery, Andrew Caines, James Thorne, Christopher Bryant, Simone Teufel and the anonymous ACL reviewers for their insightful comments. We thank the NVIDIA Corporation for the donation of the Titan X Pascal GPU used in this research. We gratefully acknowledge our funding bodies: Youmna Farag was supported by the EPSRC and Cambridge Trust; Helen Yannakoudakis was supported by Cambridge Assessment, University of Cambridge.

References

Regina Barzilay, Noemie Elhadad, and Kathleen R. McKeown. 2002. Inferring strategies for sentence ordering in multidocument news summarization. *J. Artif. Int. Res.*, 17(1):35–55.

Regina Barzilay and Mirella Lapata. 2005. Modeling local coherence: An entity-based approach. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 141–148. Association for Computational Linguistics.

Regina Barzilay and Mirella Lapata. 2008. Modeling local coherence: An entity-based approach. *Computational Linguistics*, 3(1):1–34.

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*, pages 681–684. Association for Computational Linguistics.

Danqi Chen and Christopher Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 740–750. Association for Computational Linguistics.

Baiyun Cui, Yingming Li, Ming Chen, and Zhongfei Zhang. 2018. Deep attentive sentence ordering network. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4340–4349. Association for Computational Linguistics.

Takshak Desai, Parag Dakle, and Dan Moldovan. 2018. Generating questions for reading comprehension using coherence relations. In *Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications*, pages 1–10.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Micha Elsner and Eugene Charniak. 2008. Coreference-inspired coherence modeling. In *Proceedings of ACL-08: HLT, Short Papers*, pages 41–44. Association for Computational Linguistics.

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*, pages 125–129. Association for Computational Linguistics.

Youmna Farag, Helen Yannakoudakis, and Ted Briscoe. 2018. Neural automated essay scoring and coherence modeling for adversarially crafted input. In *Proceedings of the 2018 Conference of the North*

American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pages 263–271. 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 *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 940–949. Dublin City University and Association for Computational Linguistics.

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

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

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Shafiq Joty, Muhammad Tasnim Mohiuddin, and Dat Tien Nguyen. 2018. Coherence modeling of asynchronous conversations: A neural entity grid approach. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 558–568. Association for Computational Linguistics.

Alice Lai and Joel Tetreault. 2018. Discourse coherence in the wild: A dataset, evaluation and methods. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 214–223. Association for Computational Linguistics.

Mirella Lapata. 2006. Automatic evaluation of information ordering: Kendall's tau. *Comput. Linguist.*, 32(4):471–484.

Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. 2016. Visualizing and understanding neural models in NLP. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 681–691. Association for Computational Linguistics.

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. 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. 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*, pages 997–1006. Association for Computational Linguistics.

Lajanugen Logeswaran, Honglak Lee, and Dragomir R. Radev. 2018. Sentence ordering and coherence modeling using recurrent neural networks. In *AAAI*, pages 5285–5292. AAAI Press.

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*, pages 1157–1168. Association for Computational Linguistics.

Eleni Miltsakaki and Karen Kukich. 2004. Evaluation of text coherence for electronic essay scoring systems. *Natural Language Engineering*, 10(1):25–55.

Daraksha Parveen, Mohsen Mesgar, and Michael Strube. 2016. Generating coherent summaries of scientific articles using coherence patterns. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 772–783. Association for Computational Linguistics.

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.

Marek Rei and Helen Yannakoudakis. 2017. Auxiliary objectives for neural error detection models. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*.

Victor Sanh, Thomas Wolf, and Sebastian Ruder. 2018. A hierarchical multi-task approach for learning embeddings from semantic tasks. *arXiv preprint arXiv:1811.06031*.

Swapna Somasundaran, Jill Burstein, and Martin Chodorow. 2014. Lexical chaining for measuring discourse coherence quality in test-taker essays. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 950–961. Dublin City University and Association for Computational Linguistics.

Nitish Srivastava, Geoffrey E 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(1):1929–1958.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R Bowman, Dipanjan Das, et al. 2019. What do you learn from context? probing for sentence structure in contextualized word representations. *arXiv preprint arXiv:1905.06316*.

Tijmen Tieleman and Geoffrey Hinton. 2012. Lecture 6.5 - rmsprop. *Technical report*.

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)*, pages 1320–1330. Association for Computational Linguistics.

Suzan Verberne, LWJ Boves, NHJ Oostdijk, and PAJM Coppen. 2007. Evaluating discourse-based answer extraction for why-question answering. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.*, pages 735–736.

Helen Yannakoudakis and Ted Briscoe. 2012. Modeling coherence in esol learner texts. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 33–43. Association for Computational Linguistics.

Alexander Yeh. 2000. More accurate tests for the statistical significance of result differences. In *Proceedings of the 18th conference on Computational linguistics-Volume 2*, pages 947–953. Association for Computational Linguistics.

A Grammatical roles

Type	Description
acl [relcl]	clausal modifier of noun (adjectival clause)
advcl	adverbial clause modifier
advmod	adverbial modifier
amod	adjectival modifier
appos	appositional modifier
aux	auxiliary
auxpass	passive auxiliary
case	case marking
cc [preconj]	coordinating conjunction
ccomp	clausal complement
compound [prt]	compound
conj	conjunct
cop	copula
csubj	clausal subject
csubjpass	clausal passive subject
dep	unspecified dependency
det [predet]	determiner
discourse	discourse element
dobj	direct object
expl	expletive
iobj	indirect object
mark	marker
mwe	multi-word expression
neg	negation modifier
nmod [tmod, poss, npmod]	nominal modifier
nsubj	nominal subject
nsubjpass	passive nominal subject
nummod	numeric modifier
parataxis	parataxis
punct	punctuation
root	root
xcomp	open clausal complement

Table 6: The GR types (UDs) extracted from the WSJ training data. The text inside [] (left column) denotes the extracted subtypes (language specific types).[a] The total number of main types and their subtypes is 39.[b]

[a] For more details about subtypes please see http://universaldependencies.org/docsv1/ext-dep-index.html.

[b] For the full list of UDs please see http://universaldependencies.org/docsv1/u/dep/index.html.