Fact Aware Multi-task Learning for Text Coherence Modeling

Tushar Abhishek(✉), Daksh Rawat, Manish Gupta, and Vasudeva Varma

Information Retrieval and Extraction Lab, IIIT, Hyderabad, India
tushar.abhishek@research.iiit.ac.in,
daksh.rawat@students.iiit.ac.in, {manish.gupta,vv}@iiit.ac.in

Abstract. Coherence is an important aspect of text quality and is crucial for ensuring its readability. It is essential for outputs from text generation systems like summarization, question answering, machine translation, question generation, table-to-text, etc. An automated coherence scoring model is also helpful in essay scoring or providing writing feedback. A large body of previous work has leveraged entity-based methods, syntactic patterns, discourse relations, and traditional deep learning architectures for text coherence assessment. However, these approaches do not consider factual information present in the documents. The transitions of facts associated with entities across sentences could help capture the essence of textual coherence better. We hypothesize that coherence assessment is a cognitively complex task that requires deeper fact-aware models and can benefit from other related tasks. In this work, we propose a novel deep learning model that fuses document-level information with factual information to improve coherence modeling. We further enhance the model efficacy by training it simultaneously with Natural Language Inference task in multi-task learning setting, taking advantage of inductive transfer between the two tasks. Our experiments with popular benchmark datasets across multiple domains demonstrate that the proposed model achieves state-of-the-art results on a synthetic coherence evaluation task and two real-world tasks involving prediction of varying degrees of coherence.

1 Introduction

Coherence is a crucial metric for text quality analysis. It assimilates how well the sentences are connected and how well the document is organized. Coherent documents have clear topic transitions that are discussed throughout the text with a smooth flow of concepts, typically in an increasing order of complexity. Ideas are first introduced in preceding sentences and are referred to later in document. Connectives are often used to assist the structure and for smooth transitions within the document. Overall, coherence leads to better clarity.

M. Gupta—The author is also a Principal Applied Scientist at Microsoft.

J. Gama et al. (Eds.): PAKDD 2022, LNAI 13281, pp. 340–353, 2022.
https://doi.org/10.1007/978-3-031-05936-0_27

Coherence is vital for multiple Natural Language Processing (NLP) applications like summarization [3,44], question answering [51], machine translation [38,55], question generation [10], language assessment for essay scoring [8,16,46], story generation [34], readability assessment [41,45] and other text generation [22,26,43].

Many formal theories of coherence [2,19,33] have been proposed leading to further development of various coherence models. Based on such theories, multiple text coherence models like entity-grid [4] and its extensions have been proposed. Other linguistic approaches for text coherence include coreference resolution, discourse relations, lexical cohesion, and syntactic features. However, feature engineering is decoupled from the prediction task thus limiting model performance. Recently, various models have been proposed which leverage deep learning architectures like convolutional neural networks (CNNs), recurrent neural networks (RNNs), long short-term memory networks (LSTMs). Transformer [50] based approaches [23–25] have also been proposed that achieve better results on coherence modeling and its downstream tasks.

However, these approaches do not consider the factual information present in the document. Recent work has demonstrated usefulness of fact triples ⟨subject, verb, object⟩ for improving result on various NLP tasks, such as summarization [20], Question Answering (QA) [47], Natural Language Inference (NLI) [1] and language modeling [53]. In this work, we propose a novel architecture that fuses document-level information with factual information to improve coherence modeling. Further, we enhance the accuracy of coherence prediction by jointly modeling coherence and Natural Language Inference (NLI) in a multi-task learning (MTL) setting.

Overall, in this paper, we make the following main contributions. (1) We investigate the effectiveness of novel fact-aware MTL architecture. (2) We assess the extent to which the information encoded in the network generalizes to multiple domains and demonstrate the effectiveness of our approach not only on popular sentence order discrimination task but also on more realistic task like predicting coherence of varying degrees in people's everyday writings. (3) Experiments on popular benchmark datasets (GCDC and WSJ) indicate that our proposed methods establish SOTA across multiple (task, dataset) combinations. (4) On an automated essay scoring (AES) task, we demonstrate that addition of coherence signal from our model significantly improves AES accuracy.

2 Related Work

Entity-Grid Based Methods: Discourse coherence has been studied widely using both deep learning as well as non-deep learning models. Barzilay et al. [4] proposed the entity grid model, which is based on Centering Theory [19]. It captures the distribution of discourse entities and transition of grammatical roles (subject, object, neither) across the sentences. Several extensions were proposed by utilising entity specific features [13], modifying ranking scheme [17] or transforming problem into bipartite graph [35]. The entity grid method as well as extensions suffer from two main drawbacks: (1) they use discrete representation for grammatical roles and features, which prevents the model from considering sufficiently long transitions due to the curse of dimensionality problem. (2) Feature engineering is decoupled from the prediction task, which limits the model's capacity to learn task-specific features.

Other Feature Engineering Methods: Besides entity grid, other linguistic approaches for text coherence include coreference resolution, discourse relations, lexical cohesion, and syntactic features. Elsner et al. [13] proposed a maximum-entropy based discourse-new classifier that classifies mentions of all referring expression as first mention (discourse-new) or subsequent (discourse-old) mentions. Louis et al. [32] proposed a coherence model based on syntactic patterns by assuming that sentences in a coherent discourse should share the same structural syntactic patterns. Other approaches have used syntactic patterns [32], lexical cohesion [40,46] or capture topic shifts via HMMs [5].

Deep Learning Methods: Recently, multiple deep learning approaches have been proposed. Li et al. [29] propose a neural framework to compute the coherence score of a document by estimating a coherence probability for each clique of L sentences. Li et al. [30] propose generative methods to capture global topic information. Nguyen et al. [42] and Mohiuddin et al. [37] transform entity-grid based methods into deep learning versions that obtain better results than traditional counterparts. Farag et al. [15] propose a hierarchical attention model with multi-task learning objective. Xu et al. [56] and Moon et al. [39] show that modeling local coherence with discriminative models could capture both the local and the global contexts of coherence. Guz et al. [21] propose an RST-Recursive model, which takes advantage of the text's RST features. Farag et al. [14] extend some of the previous discriminative models using BERT (Bidirectional Encoder Representations from Transformers) [11] embeddings. Recently, Transformer [50] based approaches [23–25] have been proposed that achieve better results.

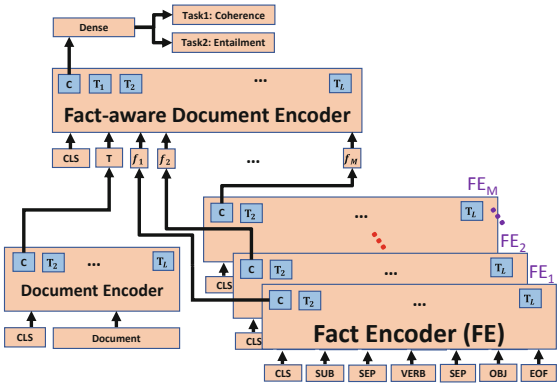

Fig. 1. An overview of our proposed fact-aware multi-task learning architecture. M distinct facts extracted from the document are fed to Fact Encoder individually to get permutation invariant representation. Fact-aware document encoder combines the document representation with M factual representation to obtain the fact-aware document representation.

3 Proposed Model

Given a document D, our goal is to assess its coherence according to the downstream task (binary classification, multi-class classification or regression task). Figure 1 provides an overview of our novel fact-aware multi-task learning model. It consist of three components: (i) Fact extractor to extract facts from textual content, (ii) Fact-aware document encoder that fuses the textual information with factual information, and (iii) Multi-task learning (MTL) framework that add auxiliary objective of textual entailment prediction to coherence objective. We discuss these components in detail in the following.

3.1 Fact Extractor

We leverage MinIE, an Open Information Extraction (IE) system [18] to generate a set of facts for each sentence. Open IE systems aim to exploit linguistic information including dependency relations in sentences to extract facts in a knowledge-agnostic manner. A fact is essentially an ordered 3-tuple ⟨subject, verb, object⟩ extracted from a particular sentence. A single sentence can produce multiple facts. Consider the sentence "They are trying to determine whether it was used to attack Steenkamp, if she used the bat in self-defense." Two facts that can be extracted from this sentence are ("it", "was used to attack", "Steenkamp") and ("she", "used bat in", "self-defense"). Each of the three components of a fact triple can contain multiple words.

For a given document D we pass the textual content through fact extractor (MinIE) to extract in-domain facts. Let M be the number of distinct facts obtained from the document D using MinIE.

3.2 Fact-Aware Document Encoder

This module follows a hierarchical structure with the following two encoders at the bottom level: (i) document encoder, and (ii) fact encoder. Each encoder uses a transformer model. Document encoder and fact encoder share weights. For i^{th} fact triple obtained from fact extractor for given document D, we create linear fact string by concatenating the subject, predicate and object delimited by separator token (SEP). The linear fact string is then fed to fact encoder FE_i individually to produce permutation invariant fact representation f_i. The document encoder encodes the document expressed using standard sub-word tokens to obtain document-level representation T. These fact and the document representations, T and f_i respectively, form the input for the fact-aware document encoder. Finally, we obtain fact-aware document representation as the CLS token vector from the last layer of the fact-aware document encoder. This is then fed to a fully-connected layer with ReLU, and then to a task specific output layer.

3.3 MTL Framework

When multiple related prediction tasks need to be performed, multi-task learning (MTL) has been found to be very effective. We experimented with various Natural

Language Understanding (NLU) tasks as auxiliary task and empirically found MTL combination of textual entailment and text coherence task provides better generalization and robustness. For a given a pair of sentences, the textual entailment task aims to predict whether the second sentence (hypothesis) is an entailment with respect to the first one (premise) or not. We share the fact-aware document encoder weights across the two tasks. Task specific layers for each task are conditioned on the shared fact-aware document encoder. For the sentence entailment task, we form input by concatenating the hypothesis and premise with sentence separator token SEP placed between them. For both the tasks (coherence and entailment), we use a fully-connected layer with ReLU, and then a softmax output layer. The final loss is computed as a sum of the individual losses for the two tasks. In the multi-task learning, we use mini-batch based stochastic gradient descent (SGD) to learn the parameters of our model (i.e., the parameters of all shared layers and task-specific layers) as shown in Algorithm 1.

Algorithm 1: Training a fact-aware MTL model

```
Model trainable parameters : θ (initialized to pretrained weights)
Set the max number of epochs: epoch_max.
for epoch in 1, 2, ..., epoch_max do
    Merge coherence and entailment dataset: D_global = D_coh ∪ D_entail
    Shuffle D_global
    for batch in D_global do
        Initialize losses: L_coh = 0, L_entail = 0
        if batch_coh ∈ batch then
            L_coh = Compute text coherence loss on batch_coh
        if batch_entail ∈ batch then
            L_entail = Compute text entailment loss on batch_entail
        Combine loss: L_total = L_coh + L_entail.
        Update the gradients and θ
```

4 Evaluation Tasks and Datasets

We experiment with two popular benchmark datasets: Wall Street Journal (WSJ) and Grammarly Corpus of Discourse Coherence (GCDC). GCDC is a real dataset while WSJ is a synthetic dataset. We use the Recognizing Textual Entailment (RTE) dataset [52] for training the auxiliary task head for our MTL model (2490 train and 277 validation instances) for experiments on GCDC. For WSJ, we found MTL to perform better when we use the Multi-Genre Natural Language Inference (MNLI) dataset [54] (21560 train and 6692 validation instances) for training the auxiliary task. We also evaluated the efficiency of proposed architecture on one downstream task: Automated Essay Scoring (AES). For AES task we use Automated Student Assessment Prize (ASAP) dataset. We make the code and dataset publicly available[1].

WSJ Sentence Order Discrimination Task. The WSJ portion of the Penn Treebank [13,42] is one of the most popular datasets for the sentence order discrimination task. It contains long articles without any constraint on style. Following previous

[1] https://www.dropbox.com/s/wolrmesgr4k1lf8/fact-aware-mtl-text-coh.zip.

work [4,42], we also use the sections 00–13 for training and 14–24 for testing (documents consisting of only 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. We labeled these permuted documents as negative samples. The dataset is created by pairing the original document and the permuted document. The task is to rank the original document higher than the permuted one in terms of coherence. We present the basic statistics of the dataset in Table 1.

We evaluate model performance on this dataset using pairwise ranking accuracy (PRA) between original text and its 20 permuted counterparts, similar to previous work. PRA calculates the fraction of correct pairwise rankings in the test data (i.e., the original coherent text should be ranked higher than its permuted non-coherent counterpart).

For this task, the coherent and incoherent document representations are obtained by using proposed fact-aware document encoder using the architecture shown in Fig. 1. Further, on top of these representations, we apply Siamese network [7] as illustrated in Fig. 2. The document encoder for the coherent as well as the incoherent document, share weights. Both the document representations are separately connected to a dense layer with shared weights. Outputs of the dense layers are used to calculate margin ranking loss.

Table 1. Basic statistics of the WSJ dataset. #Docs represents the number of original articles and #Synthetic Docs represents the number of original articles and their permuted versions.

	#Docs	#Syn. Docs	Avg #Sents	Avg #Words
Train	1376	29720	21.0	529.8
Test	1090	21800	21.9	564.3

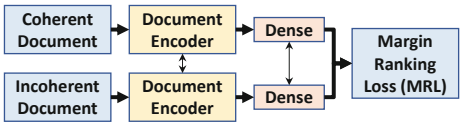

Fig. 2. Overview of Siamese neural approach applied for sentence order discrimination task. Document encoder weights are shared. Dense layer weights are also shared.

GCDC 3-Way Classification. The GCDC dataset contains emails and reviews written with varying degrees of proficiency and care [28]. The WSJ dataset contains documents that have been professionally written and extensively edited. In contrast to WSJ, the GCDC dataset contains writing from non-professional writers in everyday contexts. Rather than using permuted or machine generated texts as examples of low coherence, GCDC has real sentences in which people try but fail to write coherently. GCDC is a corpus that contains texts from four domains, covering a range of coherence, each annotated with a document-level coherence score. 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. We present the basic statistics of the dataset in Table 2.

Given a document, the task is to classify it into one of the three different labels (high, medium and low) which denotes the textual coherence level of the given document. For each of these domains, a fixed split of 1000 and 200 was used for train and test respectively as specified in [28]. Of the 1000 documents, we use 200 documents for validation and remaining 800 for training. For our experiments, we use the consensus rating of the expert scores as calculated by [28], and train our models for all the four domains. To evaluate model performance, we use 3-way classification accuracy.

Table 2. Basic statistics of the GCDC dataset. For each of these domains, a fixed split of 1000 and 200 was used for train and test respectively as specified in [28]

	#Docs	Avg #Words	Avg #Sents	Low, medium, high instances (%)
Yahoo	1200	162.1	7.5	46.6,17.4,37.0
Clinton	1200	189.0	6.6	28.2,20.6,51.2
Enron	1200	196.2	7.7	29.9,19.4,50.7
Yelp	1200	183.1	7.5	27.1,21.8,51.1

Table 3. Statistics of ASAP dataset.

Prompt	#Essays	Genre	#Avg length	Range of scores
1	1783	Argumentative	350	2–12
2	1800	Argumentative	350	2–12
3	1726	Response	150	0–3
4	1772	Response	150	0–3
5	1805	Response	150	0–4
6	1800	Response	150	0–4
7	1569	Narrative	250	0–30
8	723	Narrative	650	0–60

ASAP Automated Essay Scoring. Automated Student Assessment Prize (ASAP) dataset is taken from the Kaggle competition[2] which was organized and sponsored by the William and Flora Hewlett Foundation and ran on Kaggle from 10-Feb-12 to 30-Apr-12. The essays are associated with scores given by humans and categorized in eight prompts. Table 3 summarizes some properties of this dataset. The task is to assign an automatic score for a given essay, aiming to replicate human scoring results. Essays are segregated into different prompts based on essay topic and genre. We normalize all score range to within [0, 1]. The scores are re-scaled back to the original prompt-specific scale for calculating Quadratic Weighted Kappa (QWK) scores. The reader can refer [48] to get more details on QWK. We conduct the evaluation in prompt-specific fashion as done in [48].

For this task, we follow previous studies [36,57]. First, we obtain the essay's feature vector v_1 by training a Longformer model for AES task, and take CLS token representation from the last layer. Next, without any AES-task-specific finetuning, we obtain a coherence vector v_2 produced by our model finetuned on WSJ task. The concatenation of v_1 and v_2 is now "coherence augmented representation" of the essay. This representation is passed to a linear layer with sigmoid activation for final essay scoring. We hope that augmentation by v_2 obtained from our model will improve AES scoring accuracy.

5 Experiments

5.1 Baselines

For WSJ and GCDC Related Tasks. We perform extensive comparisons with the following baselines. While Flesch-Kincaid grade level (FKGL) [27] is a readability measure, previous work has treated readability and text coherence as overlapping tasks [4,35]. For coherence classification, Mesgar et al. [35] search over the grade level scores on the training data and select thresholds that result in the highest accuracy. Entity grid (EGRID) [4] builds an entity grid which is a matrix that tracks entity mentions over sentences. Random forest classifier is trained over features extracted from entity grid. CNN-Egrid [42] is a local coherence model that employs a CNN that operates over the entity grid representation. LCNN-Egrid [37] extends CNN-Egrid with lexical information about the entities. In Local Coherence Model (LC) [29], sentences are

[2] https://www.kaggle.com/c/asap-aes/.

encoded with a recurrent or recursive layer and a filter of weights is applied over each window of sentence vectors to extract scores that are aggregated to calculate overall document coherence score. Paragraph sequence (PARSEQ) [28] contains three stacked LSTMs to represent sentence, paragraph and document. Hierachical LSTM [15] is very similar to PARSEQ, but with attention and uses BiLSTMs. Coh+GR [15] extends Hierachical LSTM by training it to predict word-level labels indicating the predicted grammatical role (GR) type at the bottom layers of the network, along with the document-level coherence score. Coh+SOX [15] is same as Coh+GR where, for each word, we only predict subject (S), object (O) and 'other' (X) roles. Seq2Seq [30] consists of two LSTM generative language models and uses the difference between conditional log likelihood of a sentence given its preceding/succeeding context, and the marginal log likelihood of the current/next sentence to assess coherence. Local Coherence Discriminator (LCD-L) [56] uses max-pooling on the hidden state of the language model to get the sentence representation. A representation for two consecutive sentences is then computed by concatenating the output of a set of linear transformations applied to the two sentences. This is fed to a dense layer and used to predict a local coherence score. Coh+GR_BERT [14] is similar to Coh+GR, except that BERT embeddings are used instead of GloVe embeddings as input to BiLSTMs. LCD_BERT [14] is similar to LCD-L but uses averaged BERT (instead of GloVe) embeddings as the sentence representations. We also included LCD_RoBERTa which similar to LCD_BERT but uses RoBERTa embeddings instead of BERT. Unified [39] uses a combination of LSTMs and CNNs. Inc-lex-Coh [24] extracts sentence representations using a pretrained language model and combines the semantic centroid vector with semantic similarity vector to obtain coherence output. They also created another variant Avg-XLNET-Doc that encodes an text content at the document level and averages the encoded representations. We created RoBERTa variant of this model Avg-RoBERTa-Doc where we used RoBERTa embedding instead of XLNET.

For AES/ASAP Task. We perform extensive comparisons with the following baselines. EASE is publicly available, open-source[3] software which ranked third amongst 154 participants in the ASAP competition. It uses manual feature engineering with Support Vector Regression (SVR) and Bayesian Linear Ridge Regression (BLRR). EASE+cohLSTM [36] combines the feature vector computed by EASE, and the coherence vector produced by LSTM-based coherence model to obtain a more reliable representation of an essay. Constraint MTL [9] uses a constrained multi-task pairwise preference learning approach that enables the data from multiple tasks to be combined effectively. Attention based RCNN [12] uses hierarchical sentence-document model to represent essays, using the attention mechanism to learn the relative importance of words and sentences. SkipFlow [49] models coherence using the similarity between multiple states of an LSTM over time with a bounded window.

5.2 Experimental Settings and Reproducibility Information

All experiments were run on a machine equipped with four 32GB V100 GPUs. For all our models, we use 12-layer models, and embedding layer was frozen except for

[3] https://github.com/edx/ease.

Table 4. Sentence order discrimination task Pairwise Ranking Accuracy (PRA) results on WSJ

	Model	PRA
Baselines	LC	74.10
	PARSEQ	74.10
	Seq2Seq	86.95
	CNN-Egrid	88.69
	Unified (ELMo)	93.19
	Coh+GR	93.20
	LCD-L	95.49
	Coh+GR_BERT	96.10
	LCD_RoBERTa	96.45
	LCD_BERT	97.10
Ours	Vanilla Transformer	97.34
	Fact-aware Transformer	97.81
	Fact-aware MTL Trans	**98.22**

Table 5. 3-way classification accuracy results on GCDC.

	Models	Yahoo	Clinton	Enron	Yelp	Average
Baselines	EGRID+coref	41.5	48.0	47.0	49.0	46.4
	EGRAPH+coref	42.5	55.0	44.0	54.0	48.9
	LCNN-Egrid+coref	51.0	56.6	44.7	54.0	51.6
	FKGL	43.5	56.0	52.5	55.0	51.8
	Coh+SOX	50.5	58.5	51.0	–	53.3
	Hierachical LSTM	55.0	59.0	50.5	–	54.8
	PARSEQ	54.9	60.2	53.2	54.4	55.7
	LC	53.5	61.0	54.4	–	56.3
	PARSEQ (A)	58.5	61.0	53.9	56.5	57.5
	Coh+GR	56.0	62.0	56.0	–	58.0
	Inc-lex-Coh	57.3	61.7	54.5	**59.0**	58.1
	Avg-RoBERTa-Doc	60.0	65.3	55.0	58.8	59.8
	Avg-XLNet-Doc	60.5	65.9	**56.9**	**59.0**	60.6
Ours	Vanilla Trans.	58.1	63.9	55.3	57.6	58.7
	Fact-aware Trans.	59.2	67.2	56.3	58.5	60.3
	Fact-aware MTL Trans.	**60.7**	**67.4**	56.4	**59.0**	**60.8**

Table 6. Experimental results on ASAP dataset of our approach versus the baseline methods. Results are reported in terms of the quadratic weighted kappa (QWK) measure, using 5-fold cross-validation. Best QWK for each prompt is highlighted in bold.

	Models	Prompts								
		1	2	3	4	5	6	7	8	Average
Baselines	CohLSTM	0.669	0.634	0.591	0.710	0.639	0.716	0.729	0.641	0.666
	EASE (SVR)	0.781	0.630	0.621	0.749	0.782	0.771	0.727	0.534	0.699
	EASE (BLRR)	0.761	0.606	0.621	0.742	0.784	0.775	0.730	0.617	0.705
	EASE+CohLSTM	0.784	0.654	0.663	0.788	0.793	0.794	0.756	0.646	0.735
	Constraint MTL	0.816	0.667	0.654	0.783	0.801	0.778	0.787	0.692	0.747
	Attention based RCNN	0.822	0.682	0.672	0.814	0.803	0.811	0.801	**0.705**	0.764
	SkipFlow	**0.832**	**0.684**	0.695	0.788	**0.815**	0.810	0.800	0.697	0.765
Ours	Longformer	0.824	0.660	0.693	0.820	0.795	0.810	0.817	0.701	0.765
	Longformer+Fact aware MTL Trans.	0.822	0.674	**0.696**	**0.821**	0.798	**0.812**	**0.822**	0.699	**0.768**

the sentence order discrimination task on WSJ. For fact-aware document encoder, we used pretrained model for the fact encoders and document encoder, and a randomly initialized RoBERTa for fact-aware document encoder. For all experiments we cap the maximum number of facts to 100.

For all experiments, we run 10 epochs except ASAP where we use 5-fold cross validation, weight decay of 0.01 and use a dropout of 0.1. We use Adam optimizer for experiments on GCDC, and use AdamW for WSJ and ASAP experiments. For all the baseline models, we report results from their original papers. For all of our models, the reported results on WSJ and GCDC dataset, are the mean of 10 runs with different random seeds. Margin for the margin ranking loss is set to 1. For MTL framework, categorical cross entropy loss was used for the auxiliary task. We use Longformer based

models for WSJ and ASAP dataset to handle the long input documents. For Longformer, we fixed max sequence length to 2048. For RoBERTa, we fixed it to 512. We use learning rate of 2e−5 for all experiments. We use batch size of 2 for all the models on all the tasks.

For model proposed for Automated Essay Scoring (ASAP), we use 5-fold cross validation to evaluate all systems with a 60/20/20 split for train, dev and test sets. We use the splits provided by [48] and closely follow the same experimental procedure. We train our models on ASAP using mean square error (MSE) for 10 epochs and select the best model based on the performance on the validation set.

5.3 Results

Tables 4 and 5 show the results for the two text coherence tasks for WSJ and GCDC datasets respectively. Broadly we observe that our proposed approach significantly outperforms baselines, establishing a new SOTA across all tasks. Across all tasks, the results using our method are statistically significantly better compared to the best baseline with $p \leq 10^{-3}$ at 95% confidence.

Sentence Order Discrimination Results: Table 4 shows results for the sentence order discrimination task for WSJ dataset. We make the following observations: (1) Fact aware transformer outperforms vanilla transformer model as it can incorporate the factual information flow (subject in discourse) in addition to textual information which helps it to correctly determine the coherent sentences. (2) fact-aware MTL model outperforms other variants as the auxiliary task helps in better generalization over test set.

3-Way Classification Results: Table 5 shows 3-way classification results on GCDC. We make the following observations: (1) The Fact-aware model performs better than the vanilla model across all the domains, demonstrating that transitions of facts associated with entities across sentences benefit the model in capturing textual coherence signals. (2) Out of the three gold coherence labels (low, medium, high) all the models have difficulty in correctly classifying documents of medium level coherence, which can be attributed to the smaller number of training examples for that particular class.

AES Results: From Table 6 we observe that Vanilla Longformer finetuned on ASAP dataset performs better than or comparable to previous baseline approaches. Among our models, the "coherence augmented representation" from Fact aware MTL obtains the best result. To understand this a little better we computed the correlation between the coherence score predicted by the Fact aware MTL Transformer and the essay scores in ASAP dataset. We found it to be 0.48 and 0.53 for Longformer and Longformer with fact aware MTL respectively, thereby explaining why our model outperforms vanilla Longformer model.

Qualitative Analysis: We also explore our model qualitatively, examining the coherence scores assigned to some artificial miniature discourses that exhibit various kinds of coherence. The score varies from 0 to 3 and higher score denotes higher level of textual coherence. (1) Case 1: Lexical Coherence. The examples in Table 7 (type = LC) suggest that the models handle lexical coherence, correctly favoring the first over the second, and the third over the fourth and fifth examples (for all our models except the

Table 7. Qualitative analysis: Lexical Coherence (LC), Temporal Order (TO), Centering/Referential Coherence (CRC) examples. Ours = Fact-aware MTL.

Type	S. No	Text	Vanilla	Ours
LC	1	Pinochet was arrested. His arrest was unexpected	1.81	2.76
	2	Pinochet was arrested. His death was unexpected	1.67	1.56
	3	Mary ate some apples. She likes apples	1.45	2.30
	4	Mary ate some apples. She likes pears	1.47	1.45
	5	Mary ate some apples. She likes Paris	1.36	1.27
TO	1	Washington was unanimously elected president in the first two national elections. He oversaw the creation of a strong, well financed national government	1.93	2.79
	2	Washington oversaw the creation of a strong, well-financed national government. He was unanimously elected president in the first two national elections	1.88	2.36
CRC	1	Mary ate some apples. She likes apples	1.45	2.52
	2	She ate some apples. Mary likes apples	1.31	2.49
	3	John went to his favorite music store to buy a piano. He had frequented the store for many years. He was excited that he could finally buy a piano. He arrived just as the store was closing for the day	2.38	2.86
	4	John went to his favorite music store to buy a piano. It was a store John had frequented for many years. He was excited that he could finally buy a piano. It was closing just as John arrived	2.45	2.67

fact-aware one). (2) Case 2: Temporal Order. We show an example of temporal order in Table 7 (type = TO). (3) Case 3: Centering/Referential Coherence. We show a few examples of Centering/Referential Coherence in Table 7 (type = CRC). We observe that our model provides intuitive results while the Vanilla Transformer does not. This suggests that straight-forward adaptation of Transformer models for coherence assessment may not be the best approach.

6 Conclusion

In this paper, we proposed a fact-aware MTL model for text coherence assessment. The proposed model incorporates factual information with document-level information to capture transitions of facts associated with entities across sentences. We observe that our Fact aware approaches outperform existing models on synthetic data (WSJ) as well as real-world data (GCDC). Our work also demonstrates that inductive transfer between tasks: textual coherence assessment and textual entailment, provides better generalization and robustness. Coherence vector obtained from our proposed coherence models also improves the effectiveness of simple models on the automated essay scoring downstream task. In the future, we plan to extend this work to evaluate the text coherence in an open domain setting.

References

1. Annervaz, K., Chowdhury, S.B.R., Dukkipati, A.: Learning beyond datasets: Knowledge graph augmented neural networks for natural language processing. arXiv:1802.05930 (2018)
2. Asher, N., Asher, N.M., Lascarides, A.: Logics of Conversation. Cambridge University Press, Cambridge (2003)
3. Barzilay, R., Elhadad, N.: Inferring strategies for sentence ordering in multidocument news summarization. JAIR **17**, 35–55 (2002)
4. Barzilay, R., Lapata, M.: Modeling local coherence: an entity-based approach. COLING **34**(1), 1–34 (2008)
5. Barzilay, R., Lee, L.: Catching the drift: probabilistic content models, with applications to generation and summarization. In: NAACL-HLT, pp. 113–120 (2004)
6. Beltagy, I., Peters, M.E., Cohan, A.: Longformer: the long-document transformer. arXiv:2004.05150 (2020)
7. Bromley, J., et al.: Signature verification using a "siamese" time delay neural network. Int. J. Pattern Recognit. Artif. Intell. **7**(04), 669–688 (1993)
8. Burstein, J., Tetreault, J., Andreyev, S.: Using entity-based features to model coherence in student essays. In: NAACL, pp. 681–684 (2010)
9. Cummins, R., Zhang, M., Briscoe, T.: Constrained multi-task learning for automated essay scoring. In: ACL, pp. 789–799 (2016)
10. Desai, T., Dakle, P., Moldovan, D.: Generating questions for reading comprehension using coherence relations. In: Workshop on NLP Techniques for Educational Applications, pp. 1–10 (2018)
11. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805 (2018)
12. Dong, F., Zhang, Y., Yang, J.: Attention-based recurrent convolutional neural network for automatic essay scoring. In: CoNLL, pp. 153–162 (2017)
13. Elsner, M., Charniak, E.: Coreference-inspired coherence modeling. In: ACL, pp. 41–44 (2008)
14. Farag, Y., Valvoda, J., Yannakoudakis, H., Briscoe, T.: Analyzing neural discourse coherence models. arXiv:2011.06306 (2020)
15. Farag, Y., Yannakoudakis, H.: Multi-task learning for coherence modeling. arXiv:1907.02427 (2019)
16. Farag, Y., Yannakoudakis, H., Briscoe, T.: Neural automated essay scoring and coherence modeling for adversarially crafted input. arXiv:1804.06898 (2018)
17. Feng, V.W., Hirst, G.: Extending the entity-based coherence model with multiple ranks. In: EACL, p. 315–324 (2012)
18. Gashteovski, K., Gemulla, R., Del Corro, L.: MinIE: minimizing facts in open information extraction. In: EMNLP, pp. 2620–2630. ACL (2017)
19. Grosz, B.J., Weinstein, S., Joshi, A.K.: Centering: a framework for modeling the local coherence of discourse. COLING **21**(2), 203–225 (1995)
20. Gunel, B., Zhu, C., Zeng, M., Huang, X.: Mind the facts: knowledge-boosted coherent abstractive text summarization. arXiv:2006.15435 (2020)
21. Guz, G., Bateni, P., Muglich, D., Carenini, G.: Neural RST-based evaluation of discourse coherence. arXiv:2009.14463 (2020)
22. Holtzman, A., Buys, J., Forbes, M., Bosselut, A., Golub, D., Choi, Y.: Learning to write with cooperative discriminators. arXiv:1805.06087 (2018)
23. Jeon, S., Strube, M.: Centering-based neural coherence modeling with hierarchical discourse segments. In: EMNLP (1), pp. 7458–7472 (2020)

24. Jeon, S., Strube, M.: Incremental neural lexical coherence modeling. In: Proceedings of the 28th International Conference on Computational Linguistics, pp. 6752–6758 (2020)
25. Jeon, S., Strube, M.: Countering the influence of essay length in neural essay scoring. In: Second Workshop on Simple and Efficient Natural Language Processing, pp. 32–38 (2021)
26. Kiddon, C., Zettlemoyer, L., Choi, Y.: Globally coherent text generation with neural checklist models. In: EMNLP, pp. 329–339 (2016)
27. Kincaid, J.P., Fishburne Jr., R.P., Rogers, R.L., Chissom, B.S.: Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for Navy enlisted personnel. Technical report, Naval Technical Training Command Millington TN Research Branch (1975)
28. Lai, A., Tetreault, J.: Discourse coherence in the wild: a dataset, evaluation and methods. arXiv:1805.04993 (2018)
29. Li, J., Hovy, E.: A model of coherence based on distributed sentence representation. In: EMNLP, pp. 2039–2048 (2014)
30. Li, J., Jurafsky, D.: Neural net models of open-domain discourse coherence. In: EMNLP, pp. 198–209 (2017)
31. Liu, Y., et al.: RoBERTa: a robustly optimized BERT pretraining approach. arXiv:1907.11692 (2019)
32. Louis, A., Nenkova, A.: A coherence model based on syntactic patterns. In: EMNLP, pp. 1157–1168 (2012)
33. Mann, W.C., Thompson, S.A.: Rhetorical structure theory: toward a functional theory of text organization. Text **8**(3), 243–281 (1988)
34. McIntyre, N., Lapata, M.: Plot induction and evolutionary search for story generation. In: ACL, pp. 1562–1572 (2010)
35. Mesgar, M., Strube, M.: Graph-based coherence modeling for assessing readability. In: Joint Conference on Lexical and Computational Semantics, pp. 309–318 (2015)
36. Mesgar, M., Strube, M.: A neural local coherence model for text quality assessment. In: EMNLP, pp. 4328–4339 (2018)
37. Mohiuddin, T., Joty, S., Nguyen, D.T.: Coherence modeling of asynchronous conversations: a neural entity grid approach. In: ACL, pp. 558–568 (2018)
38. Mohiuddin, T., Jwalapuram, P., Lin, X., Joty, S.: CohEval: benchmarking coherence models. arXiv:2004.14626 (2020)
39. Moon, H.C., Mohiuddin, M.T., Joty, S., Xu, C.: A unified neural coherence model. In: EMNLP-IJCNLP, pp. 2262–2272 (2019)
40. Morris, J., Hirst, G.: Lexical cohesion computed by thesaural relations as an indicator of the structure of text. COLING **17**(1), 21–48 (1991)
41. Muangkammuen, P., Xu, S., Fukumoto, F., Saikaew, K.R., Li, J.: A neural local coherence analysis model for clarity text scoring. In: COLING, pp. 2138–2143 (2020)
42. Nguyen, D.T., Joty, S.: A neural local coherence model. In: ACL, pp. 1320–1330 (2017)
43. Park, C.C., Kim, G.: Expressing an image stream with a sequence of natural sentences. NIPS **28**, 73–81 (2015)
44. Parveen, D., Mesgar, M., Strube, M.: Generating coherent summaries of scientific articles using coherence patterns. In: EMNLP, pp. 772–783 (2016)
45. Pitler, E., Louis, A., Nenkova, A.: Automatic evaluation of linguistic quality in multi-document summarization. In: ACL, pp. 544–554 (2010)
46. Somasundaran, S., Burstein, J., Chodorow, M.: Lexical chaining for measuring discourse coherence quality in test-taker essays. In: COLING, pp. 950–961 (2014)
47. Sorokin, D., Gurevych, I.: Modeling semantics with gated graph neural networks for knowledge base question answering. arXiv:1808.04126 (2018)
48. Taghipour, K., Ng, H.T.: A neural approach to automated essay scoring. In: EMNLP, pp. 1882–1891 (2016)

49. Tay, Y., Phan, M.C., Tuan, L.A., Hui, S.C.: SkipFlow: incorporating neural coherence features for end-to-end automatic text scoring. In: AAAI (2018)
50. Vaswani, A., et al.: Attention is all you need. In: NIPS, pp. 5998–6008 (2017)
51. Verberne, S., Boves, L., Oostdijk, N., Coppen, P.A.: Evaluating discourse-based answer extraction for why-question answering. In: SIGIR, pp. 735–736 (2007)
52. Wang, A., et al.: SuperGLUE: a stickier benchmark for general-purpose language understanding systems. In: NIPS, pp. 3266–3280 (2019)
53. Wang, X., et al.: KEPLER: a unified model for knowledge embedding and pre-trained language representation. TACL **9**, 176–194 (2021)
54. Williams, A., Nangia, N., Bowman, S.: A broad-coverage challenge corpus for sentence understanding through inference. In: NAACL-HLT, pp. 1112–1122 (2018)
55. Xiong, H., He, Z., Wu, H., Wang, H.: Modeling coherence for discourse neural machine translation. In: AAAI, vol. 33, pp. 7338–7345 (2019)
56. Xu, P., et al.: A cross-domain transferable neural coherence model. arXiv:1905.11912 (2019)
57. Zesch, T., Wojatzki, M., Scholten-Akoun, D.: Task-independent features for automated essay grading. In: 10th Workshop on Innovative Use of NLP for Building Educational Applications, pp. 224–232 (2015)