Utilizing Attention to improve and explain Discourse Coherence classification models

Aditya Dhara
School of Information
University of California Berkeley
adityadhara@berkeley.edu

Abstract

In this paper, feed-forward Attention layers are used to improve the accuracy of classification models predicting discourse coherence of a text. Discourse coherence is an important aspect of the quality of a large text - ensuring it is connected and organized well. The quality of prediction depends heavily on context and the overarching structure across words, sentences, and paragraphs of a document. As such, the ParSeq model of stacked LSTM encoder layers (Lai and Tetreault, 2018), and a smaller variation of it, are compared against the same models with Attention in the encoder layers. We have found 2.5% improvement in accuracy over ParSeq, and reduction in over-fitting. The attention mechanism in the sentence encoder layer is also found to be useful in indicating words important to a prediction.

1 Introduction

The coherence of a discourse is an important aspect of its quality. Measuring it has many practical uses, (Lai and Tetreault, 2018) suggest an automated coherence scoring model can be useful for providing feedback on a users writing, such as missing transition between topics, poorly organized paragraphs etc. (Farag and Yannakoudakis, 2019) point out that it can inform the quality of generated language in text generation, summarization, question-answering, question generation and language assessment.

In this paper, we explore adding attention mechanisms to neural network models for predicting discourse coherence. We use the Grammarly Corpus of Discourse Coherence (Lai and Tetreault, 2018) and try to improve accuracy in predicting expert annotated labels on that corpus. There is a hierarchical nature to the task of discourse coherence (Wang and Guo, 2014), and so we will focus on hierarchical models like ParSeq introduced in (Lai and Tetreault, 2018). We will also explore the effectiveness of attention mechanisms in explaining which parts of each document were given importance in predicting a label.

2 Background

Discourse Coherence is a description of the structure of a document, specifically how sentences are connected, and how the whole document is organized (Lai and Tetreault, 2018). (Wang and Guo, 2014) show that discourses can be defined as coherent combinations of sentences or sentence fragments that result from communication between participants whether speaker and listener or writer and reader. They further explain that while individual sentences rely on grammar and regularity for structure, discourse is a choice in a semantic and pragmatic network (Wang and Guo, 2014). For a text to be coherent, it must be coherent with its context, and all parts of the text must be connected by cohesive linguistic devices, since the interpreter is involved in figuring out the implied semantic structure of the discourse (Wang and Guo, 2014).

There is a hierarchical nature to the coherence of a discourse. (Wang and Guo, 2014) point out two levels of coherence: local coherence of sentence level prepositions and composition; and global coherence of the discourse as a whole. In a similar vein, (Grosz et al, 1995) point out that a discourse and each segment of that discourse contain purpose, and that the global coherence of a discourse depends on the relationships among the discourse purpose and the discourse segments' purposes.

2.1 Existing methods

A popular model of discourse coherence is EGrid, which uses an entity-grid representation of text inspired by centering theory (Farag and Yannakoudakis, 2019). Specifically, it leverages the notion that sentence-level coherence is dependent on it's center, which is defined as an utterance that serves to link the other utterances in the discourse segment containing it (Grosz et al, 1995). In an entity grid, a text is represented as a matrix of entities and sentences. Each of those entities are represented by their grammatical role, and transitions of entities across sentences are used as features for coherence assessment (Farag and Yannakoudakis, 2019). Some other approaches mentioned in (Lai and Tetreault, 2018) include:

- EGraph: A similar model to EGrid, except entities and sentences are interpreted as a graph
- LexGraph: This uses a graph of sentences as nodes and common words as edges. K-node subgraphs are used as features in a classifier
- Clique: This encodes sentences using an LSTM, passes k-cliques of encoded sentences through a classifier, and averages the score.
- SentAvg: A neural network model that encodes sentences using an LSTM and averages them to form the document vector. The document vector is then passed through a hidden layer to classify coherence
- ParSeq: a neural network model that stacks three LSTMs to encode sentences, paragraphs and the document. The document vector is then passed through a hidden layer to classify coherence

Finally, (Farag and Yannakoudakis, 2019) constructed a multi-task learning model that is similar to ParSeq, but varies in two regards. It uses an attention mechanism between each stack of LSTMs, and also labels words with semantic roles in the bottom sentence encoder LSTM (Farag and Yannakoudakis, 2019).

3 Methods

3.1 Data and metrics

The Grammarly Corpus of Discourse Coherence (GCDC) by (Lai and Tetreault, 2018) is used below for modeling and evaluation. It contains subsets of four other data sets to represent common human tasks like forum posts, emails and product reviews (Lai and Tetreault, 2018):

- Yahoo's Answers L6 Corpus
- Enron Emails
- Clinton Emails
- Yelp reviews

(Lai and Tetreault, 2018) filtered these data sets with criteria such as being 100-300 words in length, containing few line breaks and containing no URLs. They did so on the basis of having text long enough to exhibit a range of characteristics of local and global coherence, while also being reasonable in size for expert annotators (Lai and Tetreault, 2018).

Table 1 contains counts of words, sentences, paragraphs and documents in the training data. Figure 1 shows a long tail distribution of the lengths of words, sentences and paragraphs in each document. This is not unusual due to Zipf's law, but it will necessitate some pre-processing steps addressed in section 3.4

	Count across training data
Documents	2000
Paragraphs	7051
Sentences	19475
Words	368067

Table 1: Counts of document parts across training data

Many previous approaches used sentences in large corpora as correct examples, and shuffled versions of the same sentence as negative examples for binary classification (Lai and Tetreault, 2018). In contrast, the GCDC corpus averaged the opinions of 13 expert annotators who labelled each text with "low", "medium" and "high" coherence (Lai and Tetreault, 2018). The corpus also contains labels from 1-5 generated using Amazon Mechanical Turk workers, however, they are non-experts who were given a qualification test and annotation guidelines (Lai and Tetreault, 2018).

The data used for modeling below will be the Clinton and Enron emails in GCDC. This is done because reviews, emails and forum posts each have biases in writing styles, and focusing on one

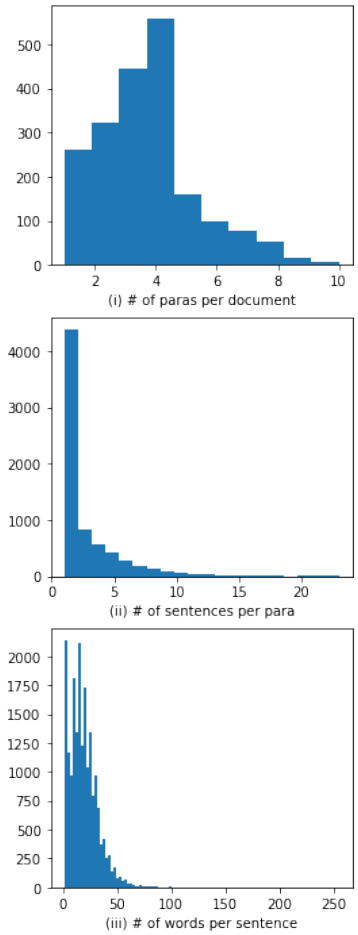

Figure 1: Histograms of lengths of parts of documents in the training data set

style can help explore the effectiveness of Attention without letting those biases color results. The data set has already been split into train and test sets, each containing 2000 and 400 documents respectively. We will focus on optimizing for the accuracy of predicting the expert annotated labels on the test set, since we are interested in the correctness of our predicted values.

When pre-processing GCDC text input, documents were first separated into paragraphs by splitting on new-lines. Each paragraph was then tokenized into sentences and words using NLTK's punkt tokenizer. Finally, TensorFlow's tokenizer, was used to convert words into indexes. These indexes were also used to populate an embedding matrix of GloVe embeddings using the publicly available pre-trained map of words to embeddings at (Pennington et al, 2014).

3.2 Models

3.3 Baseline - most frequent class

We start with a baseline of predicting the most common label in our training data. From above, that's a label of "high" coherence for all input text. In the training data, that's 50.9% of all labels.

3.3.1 Document classification models

The first model to compare is the ParSeq stacked LSTM model described in (Lai and Tetreault, 2018). It uses LSTMs with 300 hidden units to encode sentences from word embeddings, paragraphs from encoded sentences, and documents from their encoded paragraphs. The document vector is fed to a 300 unit hidden layer with $tanh$ activation before it is given to an output layer with $sigmoid$ activation. (Lai and Tetreault, 2018) also used a 50% dropout on the hidden layer during training time.

(Lai and Tetreault, 2018) used GloVe embeddings to encode words, and so 100 dimensional GLoVe embeddings are used here. They are not re-trained since models quickly over-fit on training data if that was allowed. The below implementation also uses bidirectional LSTMs with 300 hidden units each, which are concatenated outputs of LSTMs with 150 units going from left-to-right and right-to-left

$$\overrightarrow{h_t} = LSTM(x_t, h_{t-1})$$

$$\overleftarrow{h_t} = LSTM(x_t, h_{t-1})$$

$$h_t = [\overrightarrow{h_t}, \overleftarrow{h_t}]$$

In addition, a similar model but without a paragraph encoder will also be trained and compared. This model is referred to below as ParSeq-small.

3.3.2 Attention mechanisms

(Yang et. al, 2016) proposed the Hierarchical Attention Networks since different words and sentences are differentially informative to the overall coherence score, and can have different levels of importance in different contexts. In a similar manner, due to the contextual and hierarchical nature of discourse coherence, an attention mechanism might improve performance in classifying the level of coherence in a text. Additionally, (Bahdanau et al, 2015) conjectured that soft-searching over encoded input words freed their model from encoding a whole sentence into a fixed-length vector, and also lets the decoder focus on words relevant to the generation of the next word. Since discourse coherence requires looking for different pieces of a sentence in predicting its local coherence, an attention mechanism might prove helpful in a similar manner.

Thus, variations of the above ParSeq and ParSeq-small models, that use an attention layer on top of the LSTM outputs at each time-step, will also be trained and compared below. Particularly, feed-forward layers W and V are used for attention as seen in (Bahdanau et al, 2015).

$$l_i = BidirectionalLSTM(x_i, l_{i-1})$$

$$L = [l_1..l_n]$$

$$\overrightarrow{\alpha} = sigmoid(V \cdot tanh(W \cdot L))$$

$$h_i = \sum_i \alpha_i l_i$$

A good number of units in the W and V layers were found to be 200 for ParSeq with attention, and 100 for ParSeq-small with attention. Note that the ParSeq-small model with attention is identical to the model described in Hierarchical Attention Networks (Yang et. al, 2016).

3.4 Masks

When training these models, TensorFlow's keras functional API was used to create the above models. One limitation in doing so was we needed to use fixed shape tensors as input for the models. In our case since documents, paragraphs and sentences have variable lengths, they need to be padded to accommodate the largest lengths - specifically $(12, 32, 255)$. Since documents, paragraphs, and sentences all have lengths following a long tail distribution (Figure 1), most of the padded values would be 0. Thus, masks need to be pre-computed for the LSTMs to ignore the padded time-steps. As a result, each model receives a document tensor, document mask, paragraph mask and sentence mask as input during training and evaluation.

4 Results and discussion

Table 2 shows the accuracies of the models evaluated against the test set. The baseline accuracy is equivalent to the relative frequency of the label 'high' in the test data set.

Model	Accuracy
Baseline	0.4975
ParSeq	0.5400
ParSeq-small	0.4950
ParSeq + Attention	0.5650
ParSeq-small + Attention	0.5000

Table 2: Accuracy for each model

As it can be seen, adding attention to ParSeq and ParSeq-small improved their respective accuracies by 2.5% and 0.5%. However, it must be noted that the ParSeq-small performed worse than the random baseline after 10 epochs of training. ParSeq and ParSeq-small also exhibited overfitting when observing accuracy on training and test sets over each epoch of training. As can be seen in Figure 2, the training accuracy rose as test accuracy fell over time.

One limitation in the training process that couldn't be avoided is that encoders lower in the model received more input data. This is because across all documents in the training set, there are 368067 words, 19475 sentences, and 7051 paragraphs. This might have resulted in poor performance of the document encoding layers, the hidden layer and the output layer.

4.1 Attention for Explainability

While the accuracy of ParSeq with Attention had shown gains, the Attention model was less prone to over fitting, as seen in Figure 2. However, apart from performance gains, attention can be useful to determine what parts of a document contributed to a predicted label. To demonstrate, figure 3 shows

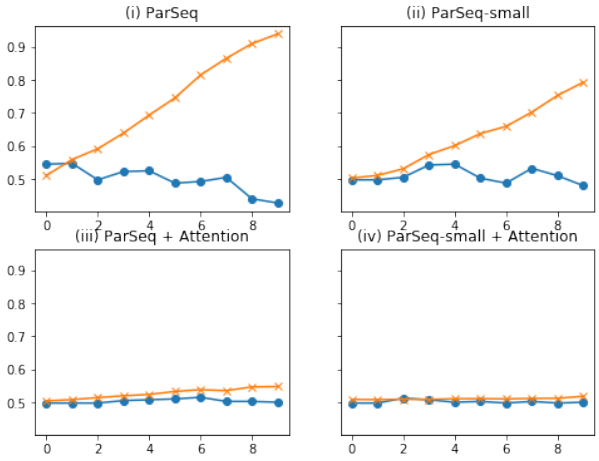

Figure 2: Train (x) and text (o) accuracies for model over 10 epochs of training

highlighted attention weights over text correctly labelled as 'low' coherence, and figured 4 shows the same over text labelled as 'high' coherence.

It appears as though the attention mechanism did not work well on the document or paragraph level. The weights appear to be evenly distributed on any prediction made using the model. This might once again be because of how document and paragraph encoders received much less training examples than sentence encoders.

On the other hand, sentence level attention across words had enough variation to demonstrate the model gives importance to some words over others. For example, in 4 the emphasis on words like "Please", and "presentation" shows the model favors words that are often found in more formal emails. In contrast, 3 had such words as "think" and "much", which might be more common in informal conversation.

4.2 Further work

It appears as though the biggest drawbacks in the above models are the lack of training of higher layers of the model, and inability of attention mechanisms to weight encoded sentences or paragraphs. The latter might be solved by using the tensor of word embeddings, as well as encoded sentences, as input into the paragraph encoder. Such contextual information might help inform the attention mechanism of which encoded sentences are to be weighted more.

Alternatively, we could also utilize an architecture that is more suited to hierarchical, and contextually interlinked structures like transformer models. Since transformers use multi-headed attention, their weights might also give clues of which words, sentences and paragraphs were useful in determining the classified label.

5 Conclusion

The coherence of a discourse is an important aspect of its quality. Using the Grammarly Discourse Coherence dataset, we incorporated an attention mechanism into each hierarchical encoder of the ParSeq model to predict discourse coherence. We were able to get a 2.5% increase in accuracy over ParSeq, and we also seemed to reduce over-fitting. We also explored whether the attention mechanisms in each encoder would contribute to explaining why a label was chosen. However, it seems the attention mechanism only shows varying weights in the sentence encoder, possibly because it received the most input data to train over.

References

Alice Lai and Joel Tetreault. 2018. *Discourse Coherence in the Wild: A Dataset, Evaluation and Methods*. Association for Computational Linguistics, Melbourne, Australia.

Barbara J. Grosz, Aravind K. Joshi and Scott Weinstein. 1995. *Centering: A Framework for Modeling the Local Coherence of Discourse*

Dzimitry Bahdanau and Kyunghyun Cho 2014. *Neural Machine Translation by Jointly Learning to Align and Translate*

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. *GloVe: Global Vectors for Word Representation*.

Youmna Farag and Helen Yannakoudakis. 2019. *Multi-Task Learning for Coherence Modeling*. Association for Computational Linguistics, Florence, Italy.

Yuan Wang and Minghe Guo. 2014. *A Short Analysis of Discourse Coherence*. Journal of Language Teaching and Research.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola and Eduard Hovy. 2016. *Hierarchical Attention Networks for Document Classification*. Association for Computational Linguistics, San Diego, CA.

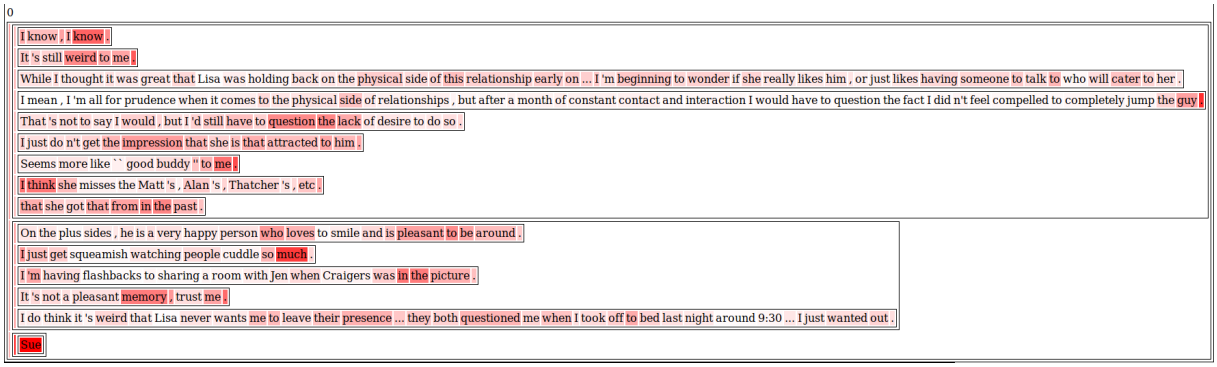

Figure 3: Attention weights for a document correctly labelled ’low’ coherence

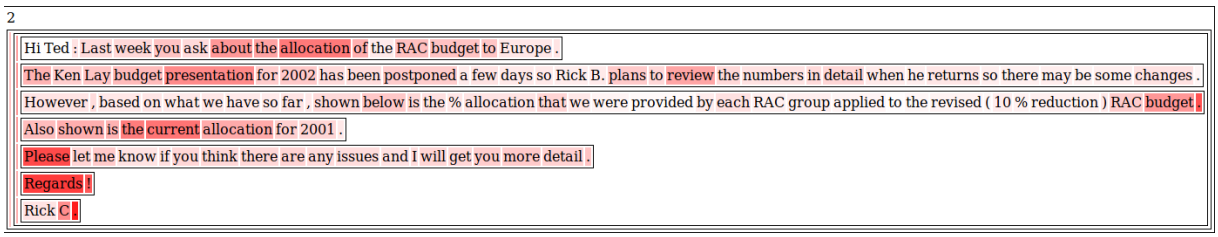

Figure 4: Attention weights for a document correctly labelled ’high’ coherence