Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 1067–1072
Marseille, 11–16 May 2020

On the Creation of a Corpus for Coherence Evaluation of Discursive Units

Elham Mohammadi, Timothe Beiko, and Leila Kosseim
Computational Linguistics at Concordia (CLaC) Laboratory
Department of Computer Science and Software Engineering
Concordia University, Montréal, Québec, Canada
elham.mohammadi@concordia.ca t.beiko23@gmail.com
leila.kosseim@concordia.ca

Abstract

In this paper, we report on our experiments towards the creation of a corpus for coherence evaluation. Most corpora for textual coherence evaluation are composed of randomly shuffled sentences that focus on sentence ordering, regardless of whether the sentences were originally related by a discourse relation. To the best of our knowledge, no publicly available corpus has been designed specifically for the evaluation of coherence of known discursive units. In this paper, we focus on coherence modeling at the intra-discursive level and describe our approach to build a corpus of incoherent pairs of sentences. We experimented with a variety of corruption strategies to create synthetic incoherent pairs of discourse arguments from coherent ones. Using discourse argument pairs from the Penn Discourse Tree Bank (Prasad et al., 2008), we generate incoherent discourse argument pairs, by swapping either their discourse connective or a discourse argument. To evaluate how incoherent the generated corpora are, we use a convolutional neural network to try to distinguish the original pairs from the corrupted ones. Results of the classifier as well as a manual inspection of the corpora show that generating such corpora is still a challenge as the generated instances are clearly not "incoherent enough", indicating that more effort should be spent on developing more robust ways of generating incoherent corpora.

Keywords: Discourse Analysis, Discourse Coherence, Corpus Creation

1. Introduction

A common assumption in natural language analysis is that the input text is coherent. However, this premise may not always hold, especially in the case of automatically generated texts or texts written by humans with lower language skills or with health issues affecting language. In these cases, the automatic evaluation of textual coherence can help towards improving the quality of automatically-generated text or detecting authors with specific linguistic deficiencies (Abdalla et al., 2018).

In order to perform automatic coherence evaluation, a corpus including both coherent and incoherent samples is needed. Coherent texts, are easy to find; however incoherent texts are not. Most corpora for textual coherence evaluation are synthetic data sets composed of randomly shuffled sentences (Lapata and Barzilay, 2005; Li and Jurafsky, 2017a; Logeswaran et al., 2018) which are commonly used for sentence ordering tasks (Logeswaran et al., 2018; Cui et al., 2018; Gong et al., 2016; Chen et al., 2016). However, these corpora do not consider if the original pairs of sentences are related by a discourse relation or not; hence, the difficulty of the sentence ordering task may vary significantly. To our knowledge, no publicly available corpus exists for coherence evaluation of known discursive units where the sentence pairs are known to have a specific discourse relation.

In this paper, we describe our approach to build a corpus of grammatically correct, but incoherent pairs of sentences. We experimented with a variety of corruption strategies to create synthetic incoherent pairs of sentences from coherent sentences with a known discourse relation. The corpora were created by swapping discourse arguments from original coherent discursive units and reconstructing new units, on the grounds that these new units would likely be inconsistent, yet grammatically correct.

Using the Penn Discourse Tree Bank (PDTB) (Prasad et al., 2008) corpus, we created a collection of pairs of sentences with a known discourse relation, then corrupted them by either modifying their discourse connective or a discourse argument. For example, the discursive unit[1]:

(1) *[John did not eat breakfast this morning.]*$_{ARG1}$ *[He managed to wait until 1 pm for his lunch date.]*$_{ARG2}$ (COMPARISON:Contrast)

is composed of two sentences related by a *contrast* discourse relation. The first sentence constitutes the first argument (`Arg1`) of the discourse unit; while the second sentence is known as argument 2 (`Arg2`). Although not explicitly marked, the two arguments are connected via an implicit discourse connective (`DC`) such as *nevertheless*. In order to corrupt this instance, we can first explicitly insert its implicit discourse connective:

(2) *[John did not eat breakfast this morning.]*$_{ARG1}$ *[Nevertheless]*$_{DC}$ *[he managed to wait until 1 pm for his lunch date.]*$_{ARG2}$ (COMPARISON:Contrast)

Then, we can corrupt the resulting instance by either replacing the discourse connective with another known to signal a different discourse relation, as in:

(3) *[John did not eat breakfast this morning.]*$_{ARG1}$ *[Otherwise]*$_{DC}$ *[he managed to wait until 1 pm for his lunch date.]*$_{ARG2}$

[1] The first argument of the implicit discourse connective is marked as *ARG1*, the second argument is denoted *ARG2* and the relation is marked at the end of the sentences in parentheses.

or changing a discourse argument, as in:

(4) *[John did not eat breakfast this morning.]*$_{ARG1}$ *[Nevertheless]*$_{DC}$ *[bonsai trees are expensive.]*$_{ARG2}$ (COMPARISON:Contrast)

thus, creating two incoherent sentence pairs.

This paper first reviews related work in the area of coherence evaluation in Section 2. Section 3 describes the six corruption strategies that were experimented with to create the corpora of incoherent sentence pairs. Section 4 describes our methods to evaluate the generated corpora. Finally, Section 5 concludes this work and proposes future directions.

2. Related Work

Previous work in coherence modeling and evaluation has mostly focused on machine-generated text.

Lapata and Barzilay (2005) discuss two linguistically rich coherence models that can be used for the automatic coherence evaluation of machine-generated content. Their dataset consists of summaries that were produced by participating systems in the Document Understanding Conference[2], tagged with their respective coherence level by human annotators. In order to automatically evaluate the coherence level of the machine-generated summaries and compare the results with human judgement, they use a syntactic model that takes into account entity transitions to distinguish between coherent and incoherent text, and a semantic model that evaluates coherence by using various measures of semantic similarity between sentences. Based on their experiments, a combined approach that makes use of both syntactic and semantic models outperforms a single one.

Assuming that coherent texts exhibit certain discourse structures, Lin et al. (2011) experiment with the use of discourse relations for the automatic evaluation of text coherence. In order to have a large collection of texts for training, they create synthetic data from a collection of source documents by permuting their sentences. They design a discourse role matrix which includes occurrences of terms and their discourse roles and use it to model transitions between textual units. They find this approach effective in distinguishing between an original coherent text and a permuted version of that text lacking coherence.

Following the same approach as Lin et al. (2011), Li and Hovy (2014) build a synthetic dataset for coherence detection which consists of source documents and their permuted versions (with a different ordering of their sentences). They feed distributed representations of tokens to a recursive neural network which computes sentence representations based on the tree structure of sentences. These distributed sentence representations are later used for coherence detection.

Li and Jurafsky (2017b) develop a neural model for coherence evaluation that is trained on a collection of coherent documents and their incoherent permuted versions (similar to the dataset used by Li and Hovy (2014)). An LSTM is used to extract sentence representations of a text. These representations are then fed to another network which calculates the probability of a text's coherence. Although this model proves effective in the task of coherence evaluation, they mention negative sampling as a disadvantage of a discriminative model, as the generated negative samples cannot possibly cover all possible meanings.

Tien Nguyen and Joty (2017) use texts' entity grid representations as input to a Convolutional Neural Network (CNN) to perform various coherence-related tasks, one of which is summary coherence rating. The dataset in their work consists of documents and multiple summaries of each document which have been generated by both humans and automatic summarization systems and ranked by human experts. Their results show that using CNNs can actually lead to an improvement on the previously reported results on the same task.

As shown above, most previous work in coherence modeling has focused on sentence ordering by creating permutations of source documents with a different ordering of their sentences. This paper goes beyond this as it focuses on coherence modeling at the intra-discursive level by evaluating the coherence between sentence pairs with known discourse relations.

3. Methodology

3.1. Dataset

In order to create a corpus of incoherent pairs of sentences, we used the Penn Discourse Treebank (PDTB) (Prasad et al., 2008). The PDTB contains 40,600 annotated discourse connectives along with their discourse arguments. The PDTB follows the DLTAG framework (Marcus et al., 1993) which takes a shallow view of discourse structures where relations are defined only between adjacent sentences or close text spans. The two textual units related by a discourse relation are known as arguments (`Arg1` and `Arg2`). The PDTB annotates the beginning and end of `Arg1` and `Arg2`, a possible discourse connective (`DC`) (for example, *because*) and the discourse relation (known as sense). The PDTB contains 18,459 instances with an explicit `DC` (from an inventory of 100 `DC`s) and 16,053 instances with an implicit `DC` where the annotators inferred a `DC`.

Example 5 shows an instance of an implicit discourse relation from the PDTB.

(5) *[So much of the stuff poured into its Austin, Texas, offices that its mail rooms there simply stopped delivering it.]*$_{ARG1}$ Implicit = so *[Now, thousands of mailers, catalogs and sales pitches go straight into the trash.]*$_{ARG2}$ (CONTINGENCY:Cause:result)

In order to maintain the grammaticality of the corrupted instances as much as possible, we used only the PDTB instances containing a discourse connective marked as *implicit*. This is because, in these cases, both `Arg1` and `Arg2` refer to two individual sentences, and `Arg1` always precedes `Arg2`. In addition, since the implicit discourse connective is guaranteed to be located at the beginning of `Arg2`, when making this connective explicit, we minimize

[2] `duc.nist.gov`

the chances of creating an ungrammatical Arg2[3].

This led to 16,053 instances that were used as the positive set that we then corrupted using six different methods.

3.2. Corruption Strategies

To corrupt the coherent instances, 6 strategies were used:

1. **Random Arg2 (RA2)**: The Arg2 of an instance is swapped with another random Arg2 in the dataset, without regards to their senses.

2. **Random DC (RDC)**: The discourse connective (DC) of an instance is swapped with another random DC in the dataset, without regards to their senses.

In order to create incoherent instances that would be easier to detect, we also tried to ensure that the discourse sense of the original instances was not maintained. This led to two other strategies:

3. **Different Sense Arg2 (DSA2)**: The Arg2 of an instance is swapped with another Arg2 in the dataset, whose sense is different from the original instance's sense.

4. **Different Sense DC (DSDC)**: The DC of an instance is swapped with another DC in the dataset, whose sense is different from the original connective's sense.

Finally, we also tried to maintain the discourse relations, hoping to create corrupted instances that would be much harder to detect as incoherent. This led to:

5. **Same Sense Arg2 (SSA2)**: The Arg2 of an instance is swapped with another Arg2 in the dataset, whose sense is identical to the original instance's sense.

6. **Same DC Arg2 (SDCA2)**: The Arg2 of an instance is swapped with another Arg2 in the dataset, whose discourse connective (DC) is identical to the original instance's DC.

We applied these 6 corruption strategies to the original coherent instances of the PDTB and thus created 6 corpora.

Table 1 shows statistics of these corpora. For each corpus, the table indicates the number of instances, the maximum sentence length, denoted max L, and whether it is a set of coherent or incoherent sentence pairs.

4. Evaluation

In order to evaluate the quality of the 6 generated corpora, we proceeded with an automatic as well as a manual evaluation.

Corpus	Method	# Instances	max L	Coherent
Original	Coherent	16,053	406	Yes
RA2	Random Arg2	16,053	404	No
RDC	Random DC	16,053	410	No
DSA2	Different Sense Arg2	16,053	420	No
DSDC	Different Sense DC	16,053	415	No
SSA2	Same Sense Arg2	16,053	419	No
SDCA2	Same DC Arg2	16,053	409	No

Table 1: Statistics of the original coherent and the 6 generated incoherent corpora.

4.1. Automatic Evaluation

For the automatic evaluation, we developed a classifier to try to discriminate coherent from incoherent instances. To do this, we used the CNN architecture used by Kim (2014) to classify movie reviews as either positive or negative. This model was chosen as the two tasks are similar and Kim (2014) achieves a high accuracy (0.81) on their dataset of movie reviews.

To run the classifier, we first merged the coherent corpus with each incoherent corpus, and labeled each instance as either coherent (1) or incoherent (0). Then, we padded each instance shorter than the longest instance in the dataset (denoted by *max L* in Table 1) to ensure that all the inputs to our model had the same length. Finally, we randomly shuffled the data and kept 90% of the instances for the training set and 10% for the test set. As the datasets were balanced, we used accuracy as our evaluation metric.

Figure 1 shows the overall architecture of the model. As shown in Figure 1, the convolution layer was applied over the word vectors and supported either a single or multiple convolution filters. Maxpooling was then used on the result of the convolutional layer and dropout regularization was added. Lastly, the output layer used a softmax activation function for the final classification.

We used word2vec (Mikolov et al., 2012) as word embeddings with a dimension of 300, pre-trained on the 100 billion words from the Google News corpus. We made the embeddings non-trainable and ran the model with parameters that restricted its capacity.

The model was trained and tested on the 6 merged versions of the datasets: (RA2, RDC, DSA2, DSDC, SSA2, and SDCA2) and coherent instances. Since the discourse connective is a strong signal to a discourse relation, we expected the performance on the DSA2 and DSDC datasets to be higher than the performance on RA2 and RDC, and the lowest results to be achieved on SSA2 and SDCA2. However, after experimenting with a variety of hyperparameters (batch size, filter size, etc.), much to our surprise, none of the datasets reached an accuracy significantly higher than 53.8% (the baseline being 50%).

In order to verify the validity of our model, we used it to reproduce the binary classification task described in Denny (2015) on the dataset of movie reviews (Pang et al., 2002) which contains 5331 positive and 5331 negative instances. With the same hyperparameters as before, the model reached an accuracy of 77%, which is comparable to the 76% reported in Denny (2015). This confirmed that the

[3] For example, if placed at the beginning of Arg2, some connectives such as *because* will create an ungrammatical sentence.

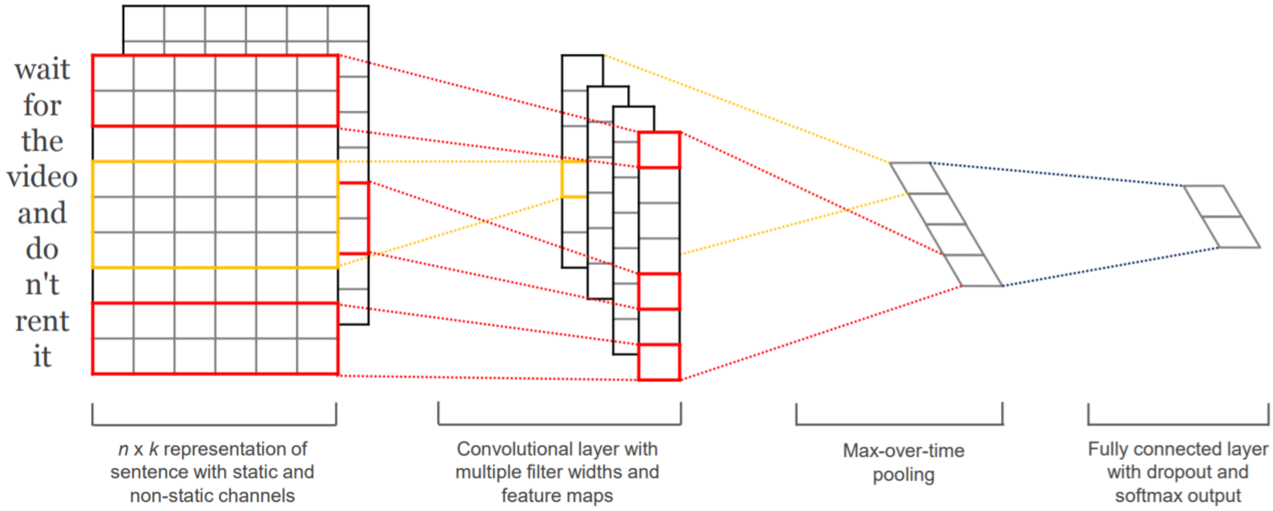
wait
for
the
video
and
do
n't
rent
it
n x *k* representation of sentence with static and non-static channels
Convolutional layer with multiple filter widths and feature maps
Max-over-time pooling
Fully connected layer with dropout and softmax output

Figure 1: Model architecture, taken from (Kim, 2014).

problem was not with the model itself, but with the generated corpora.

4.2. Manual Inspection of the Incoherent Datasets

Recall that the DSA2 corpus was created by swapping the `Arg2` of a discourse instance with another `Arg2` in the dataset, provided that the two instances had different senses. Our intuition was that this corruption strategy (along with DSDC) would have led to the most incoherent instances. We, therefore, manually inspected sample instances of the DSA2 corpus, expecting to find clear cases of incoherence. To our surprise, the instances did not seem "clearly incoherent". For example, the following instances were part of the DSA2 corpus:

(6) *[In the 1970s, several pharmaceutical and packaged-goods companies, including Colgate-Palmolive co., Eli Lilly & co., Pfizer inc. and Schering-Plough acquired cosmetics companies.]*$_{ARG1}$ *[However]*$_{DC}$ *[as that system grows, larger computers may be needed.]*$_{ARG2}$

(7) *[By starving the peasant, the communists have starved Poland.]*$_{ARG1}$ *[For example]*$_{DC}$ *[we're making a fairly obvious plea for some emotional reaction.]*$_{ARG2}$

(8) *[Some Canadian political commentators have opposed Canada's joining what they see as a U.S.-dominated organization.]*$_{ARG1}$ *[For example]*$_{DC}$ *[instead of focusing on the financial future, Mr. Dinkins has sold himself as a unifier for a city recently touched by racial violence and as a soothing antidote to 12 years of commotion generated by Mayor Koch]*$_{ARG2}$

The resulting instances are not clearly incoherent, showing that even swapping `Arg2`s with a different sense may not be sufficient.

An example from the SDCA2 corpus shows the same difficulty in judgment.

(9) *[Wall street had expected a modest rise in the company's domestic sales and earnings, and more substantial increases in overseas results.]*$_{ARG1}$ *[In addition]*$_{DC}$ *[the dollar soared against the pound, which was at \$1.5765 compared with \$1.6145 Wednesday.]*$_{ARG2}$

4.3. Manual Evaluation

In order to measure the incoherence level of the generated corpora more formally, we performed a human evaluation of samples of each corpus. Similar to the automatic evaluation, we first merged the coherent corpus with samples from different incoherent corpora. We then used the Crowdflower[4] crowdsourcing platform and asked annotators to rate each sample as either coherent or incoherent.

To ensure the quality of the annotations, we first created reference samples which were used to evaluate the annotators themselves. These reference samples consisted of instances from the corpora for which 4 English speakers agreed were either coherent or incoherent. If the crowdsourced annotators did not correctly classify over 60% of these reference samples, their annotations were discarded. We also ensured that multiple annotators would annotate each reference instance. Samples with less than 4 annotators were again discarded.

In addition, to ensure the quality of the annotations, we also used Crowdflower's confidence metric. Ranging from 0 to 1, this metric is calculated by the crowdsourcing platform and represents how many annotators classified instances the same way, weighted by the trustworthiness of each annotator, as measured by Crowdflower via metrics such as the annotators' answers in other tasks, their history on Crowdflower, and the time spent answering.

Table 2 shows the total number of samples manually evaluated for each dataset, along with the percentage of incoherent samples marked coherent by the annotators, for varying levels of confidence, denoted *C*. Note that the ground truth

[4] `www.crowdflower.com`

is 0%, as one would expect that 0% of the incoherent instances would be perceived as coherent. For the sake of comparison, we also created a corpus from the DSA2 corpus (in principle, one of the most incoherent) where the words in Arg2 were shuffled at random. This corpus is referred to as ShuffledA2 in Table 2. The expectation was that the ShuffledA2 instances would be judged as the least coherent.

Dataset	**Samples**	**Confidence Level**		
		$C > \mathbf{0}$	$C > \mathbf{0.5}$	$C > \mathbf{0.75}$
RDC	50	84.00%	85.50%	96.30%
DSDC	51	82.35%	84.00%	85.71%
RA2	51	47.06%	48.00%	55.56%
SSA2	51	54.90%	58.33%	50.00%
SDCA2	40	52.50%	53.85%	45.45%
DSA2	72	38.89%	45.16%	42.31%
ShuffledA2	51	9.80%	10.00%	6.25%

Table 2: Statistics of the evaluated samples for each dataset: percentage of incoherent samples judged coherent.

As Table 2 shows, the percentage of ShuffledA2 instances marked as coherent is indeed very low (6.25% for C>0.75). It is interesting to note that when C>0.75, in all of the incoherent corpora, except for ShuffledA2, over 40% of the instances are perceived coherent. Furthermore, Table 2 shows that in DSA2 (i.e. when swapping Arg2 with another Arg2 with a different sense) the percentage of coherent instances decreases from 55.56% in RA2 (random Arg2) to 42.31%. The same effect also holds for connectives, but to a lesser degree (96.30% to 85.71%). Finally, datasets in which the DC was changed (RDC and DSDC) seem to yield more instances perceived as coherent than when the entire Arg2 is changed (RA2, SSA2, and DSA2).

4.4. Analysis

The results of the manual evaluation revealed that annotators seemed to have a strong bias towards perceiving pairs of sentences as coherent. Several factors may have led to this phenomenon. To recognize a discourse relation, annotators need to understand each argument, how they relate to one another, and how they relate to the larger context of the whole discourse. In our experiments, annotators had difficulty understanding each of these.

First, the synthetic data created for this work was based on instances taken from the PDTB. The original instances were fairly long (with an average length of 37 words) and complex in terms of both syntactic structure and discourse domain, making the understanding of individual arguments difficult.

Also, given the specialized domain of financial and business news, annotators did not have the expertise to comprehend the relations between entities and may have relied on the inserted discourse connectives as clues to assume that the arguments were coherent.

Moreover, annotators were only given the pairs of sentences without a larger context. Without important contextual clues, annotators may not have been able to detect the incoherence and if the text allowed for a plausible interpretation, they would consider it coherent. Therefore, it is to be expected that, in the absence of contextual clues, coherence is only detected at a surface level by the annotators, resulting in inaccurate evaluations.

Finally, when annotators were unsure, the binary classification task forced them to make a choice. In hindsight, it would seem more appropriate to treat intra-discursive coherence evaluation as a regression task instead of a binary classification task. These instances can have different degrees of coherence, rather than being absolutely coherent / incoherent.

5. Conclusion and Future Work

In this paper, we highlighted the challenges of building intra-discursive incoherent instances through corruption techniques. We used the Penn Discourse Tree Bank (Prasad et al., 2008) to generate incoherent instances, by swapping either the discourse connective (DC) or Argument 2 (Arg2) of known discursive units.

We used the CNN model of Kim (2014) and Denny (2015) to classify these instances, but were unable to reach a performance greater than a random baseline. A manual evaluation through crowdsourcing revealed that the generated corpora were in fact not incoherent enough.

The annotations showed that a large percentage of the incoherent samples were actually perceived coherent by the annotators. It also provided evidence that corruption methods for generating incoherent instances based on selecting a discourse argument or discourse connective with a different sense does not seem to significantly reduce coherence.

Overall, these results show that the datasets generated were clearly not “incoherent enough”, and that effort should be spent either developing more robust ways of generating incoherent instances, or annotating “weakly corrupted” samples, such as the ones generated by our methods.

A few future directions can be proposed. First, we can adapt our method to create a corpus in which the corrupted instances are ranked based on their degree of incoherence rather than a binary classification. Also, it would be interesting to apply the same approach to shorter and syntactically simpler sentences from a simpler discourse domain. Finally, we would like to investigate the generation of synthetic instances of low coherence using Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) or sequence-to-sequence models (Sutskever et al., 2014) and explore the effectiveness of these methods for the creation of intra-discursive coherence corpora.

6. Acknowledgment

This work was financially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC). The authors would like to thank the anonymous reviewers for their feedback on a previous version of this paper.

7. References

Abdalla, M., Rudzicz, F., and Hirst, G. (2018). Rhetorical structure and Alzheimer's disease. *Aphasiology*, 32(1):41–60.

Chen, X., Qiu, X., and Huang, X. (2016). Neural sentence ordering. *CoRR*, abs/1607.06952.

Cui, B., Li, Y., Chen, M., and Zhang, Z. (2018). Deep attentive sentence ordering network. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4340–4349, Brussels, Belgium, October-November.

Denny, B. (2015). Implementing a CNN for text classification in Tensorflow. `http://www.wildml.com/2015/12`, December. Accessed: 2020-01-15.

Gong, J., Chen, X., Qiu, X., and Huang, X. (2016). End-to-end neural sentence ordering using pointer network. *CoRR*, abs/1611.04953.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in Neural Information Processing Systems 27 (NIPS)*, pages 2672–2680, Montreal, Canada, December.

Kim, Y. (2014). Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar, October.

Lapata, M. and Barzilay, R. (2005). Automatic evaluation of text coherence: Models and representations. In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI)*, volume 5, pages 1085–1090, Edinburgh, Scotland, July.

Li, J. and Hovy, E. (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, Doha, Qatar, October.

Li, J. and Jurafsky, D. (2017a). Neural net models of open-domain discourse coherence. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 198–209, Copenhagen, Denmark, September.

Li, J. and Jurafsky, D. (2017b). Neural net models of open-domain discourse coherence. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 198–209, Copenhagen, Denmark, September.

Lin, Z., Ng, H. T., and Kan, M.-Y. (2011). Automatically evaluating text coherence using discourse relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1 (ACL/HLT)*, pages 997–1006, Portland, USA, June.

Logeswaran, L., Lee, H., and Radev, D. R. (2018). Sentence ordering and coherence modeling using recurrent neural networks. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18)*, pages 5285–5292, New Orleans, USA, February.

Marcus, M. P., Santorini, B., and Marcinkiewicz, M. A. (1993). Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2012). Efficient estimation of word representations in vector space. *CoRR*, arXiv:1301.3781, January.

Pang, B., Lee, L., and Vaithyanathan, S. (2002). Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 79–86, Philadelphia, Pennsylvania, USA, July.

Prasad, R., Dinesh, N., Lee, A., Miltsakaki, E., Robaldo, L., Joshi, A. K., and Webber, B. L. (2008). The Penn Discourse Treebank 2.0. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC)*, Marrakech, Morocco, May.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27 (NIPS)*, pages 3104–3112. Montreal, Canada, December.

Tien Nguyen, D. and Joty, S. (2017). A neural local coherence model. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1320–1330, Vancouver, Canada, July.