Neural Net Models of Open-domain Discourse Coherence

Jiwei Li and Dan Jurafsky
Computer Science Department
Stanford University, Stanford, USA
jiweil,jurafsky@stanford.edu

Abstract

Discourse coherence is strongly associated with text quality, making it important to natural language generation and understanding. Yet existing models of coherence focus on measuring individual aspects of coherence (lexical overlap, rhetorical structure, entity centering) in narrow domains.

In this paper, we describe domain-independent neural models of discourse coherence that are capable of measuring multiple aspects of coherence in existing sentences and can maintain coherence while generating new sentences. We study both discriminative models that learn to distinguish coherent from incoherent discourse, and generative models that produce coherent text, including a novel neural latent-variable Markovian generative model that captures the latent discourse dependencies between sentences in a text.

Our work achieves state-of-the-art performance on multiple coherence evaluations, and marks an initial step in generating coherent texts given discourse contexts.

1 Introduction

Modeling discourse coherence (the way parts of a text are linked into a coherent whole) is essential for summarization (Barzilay and McKeown, 2005), text planning (Hovy, 1988; Marcu, 1997) question-answering (Verberne et al., 2007), and even psychiatric diagnosis (Elvevåg et al., 2007; Bedi et al., 2015).

Various frameworks exist, each tackling aspects of coherence. Lexical cohesion (Halliday and Hasan, 1976; Morris and Hirst, 1991) models chains of words and synonyms. Psychological models of discourse (Foltz et al., 1998; Foltz, 2007; McNamara et al., 2010) use LSA embeddings to generalize lexical cohesion. Relational models like RST (Mann and Thompson, 1988; Lascarides and Asher, 1991) define relations that hierarchically structure texts. The entity grid model (Barzilay and Lapata, 2008) and its extensions[1] capture the referential coherence of entities moving in and out of focus across a text. Each captures only a single aspect of coherence, and all focus on scoring existing sentences, rather than on generating coherent discourse for tasks like abstractive summarization.

Here we introduce two classes of neural models for discourse coherence. Our discriminative models induce coherence by treating human generated texts as coherent examples and texts with random sentence replacements as negative examples, feeding LSTM sentence embeddings of pairs of consecutive sentences to a classifier. These achieve state-of-the-art (96% accuracy) on the standard domain-specific sentence-pair-ordering dataset (Barzilay and Lapata, 2008), but suffer in a larger open-domain setting due to the small semantic space that negative sampling is able to cover.

Our generative models are based on augmenting encoder-decoder models with latent variables to model discourse relationships across sentences, including (1) a model that incorporates an HMM-LDA topic model into the generative model and (2) an end-to-end model that introduces a Markov-structured neural latent variable, inspired by recent work on training latent-variable recurrent nets (Bowman et al., 2015; Serban et al., 2016b). These generative models obtain the best result on a large open-domain setting, including on the difficult task of reconstructing the order of every sentence in a paragraph, and our latent variable generative model significantly improves the coherence of text generated by the model.

Our work marks an initial step in building end-to-end systems to evaluate open-domain discourse coherence, and more importantly, generating coherent texts given discourse contexts.

[1] Adding coreference (Elsner and Charniak, 2008), named entities (Eisner and Charniak, 2011), discourse relations (Lin et al., 2011) and entity graphs (Guinaudeau and Strube, 2013).

Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 198–209
Copenhagen, Denmark, September 7–11, 2017.

2 The Discriminative Model

The discriminative model treats cliques (sets of sentences surrounding a center sentence) taken from the original articles as coherent positive examples and cliques with random replacements of the center sentence as negative examples. The discriminative model can be viewed as an extended version of Li and Hovy's (2014) model but is practical at large scale[2]. We thus make this section succinct.

Notations Let C denote a sequence of coherent texts taken from original articles generated by humans. C is comprised of a sequence of sentences $C = \{s_{n-L}, ..., s_{n-1}, s_n, s_{n+1}, ..., s_{n+L}\}$ where L denotes the half size of the context window. Suppose each sentence s_n consists of a sequence of words $w_{n1}, ..., w_{nt}, ..., w_{nM}$, where M is the number of tokens in s_n. Each word w is associated with a K dimensional vector h_w and each sentence is associated with a K dimensional vector x_s.

Each C contains $2L + 1$ sentences, and is associated with a $(2L + 1) \times K$ dimensional vector obtained by concatenating the representations of its constituent sentences. The sentence representation is obtained from LSTMs. After word compositions, we use the representation output from the final time step to represent the entire sentence. Another neural network model with a *sigmoid* function on the very top layer is employed to map the concatenation of representations of its constituent sentences to a scalar, indicating the probability of the current clique being a coherent one or an incoherent one.

Weakness Two problems with the discriminative model stand out: First, it relies on negative sampling to generate negative examples. Since the sentence-level semantic space in the open-domain setting is huge, the sampled instances can only cover a tiny proportion of the possible negative candidates, and therefore don't cover the space of possible meanings. As we will show in the experiments section, the discriminative model performs competitively in specific domains, but not in the open domain setting. Secondly and more importantly, discriminative models are only able to tell whether an already-given chunk of text is coherent or not. While they can thus be used in tasks like extractive summarization for sentence re-ordering, they cannot be used for coherent text generation in tasks like dialogue generation or abstractive text summarization.

3 The Generative Model

We therefore introduce three neural generative models of discourse coherence.

3.1 Model 1: the SEQ2SEQ Model and its Variations

In a coherent context, a machine should be able to guess the next utterance given the preceding ones. A straightforward way to do that is to train a SEQ2SEQ model to predict a sentence given its contexts (Sutskever et al., 2014). Generating sentences based on neighboring sentences resembles skip-thought models (Kiros et al., 2015), which build an encoder-decoder model by predicting tokens in neighboring sentences.

As shown in Figure 1a, given two consecutive sentences $[s_i, s_{i+1}]$, one can measure the coherence by the likelihood of generating s_{i+1} given its preceding sentence s_i (denoted by *uni*). This likelihood is scaled by the number of words in s_{i+1} (denoted by N_{i+1}) to avoid favoring short sequences.

$$L(s_i, s_{i+1}) = \frac{1}{N_{i+1}} \log p(s_{i+1}|s_i) \qquad (1)$$

The probability can be directly computed using a pretrained SEQ2SEQ model (Sutskever et al., 2014) or an attention-based model (Bahdanau et al., 2015; Luong et al., 2015).

In a coherent context, a machine should not only be able to guess the next utterance given the preceding ones, but also the preceding one given the following ones. This gives rise to the coherence model (denoted by *bi*) that measures the bidirectional dependency between the two consecutive sentences:

$$\begin{aligned} L(s_i, s_{i+1}) = & \frac{1}{N_i} \log p_B(s_i|s_{i+1}) \\ & + \log \frac{1}{N_{i+1}} p_F(s_{i+1}|s_i) \end{aligned} \qquad (2)$$

We separately train two models: a forward model $p_F(s_{i+1}|s_i)$ that predicts the next sentence based on the previous one and a backward model $p_B(s_i|s_{i+1})$ that predicts the previous sentence given the next sentence. $p_B(s_i|s_{i+1})$ can be trained in a way similar to $p_F(s_{i+1}|s_i)$ with sources and targets swapped. It is worth noting that p_B and p_F are separate models and do not share parameters.

[2] Li and Hovy's (2014) recursive neural model operates on parse trees, which does not support batched computation and is therefore hard to scale up.

One problem with the described *uni* and *bi* models is that sentences with higher language model probability (e.g., sentences without rare words) also tend to have higher conditional probability given their preceding or succeeding sentences. We are interested in measuring the informational gain from the contexts rather than how fluent the current sentence is. We thus propose eliminating the influence of the language model, which yields the following coherence score:

$$\begin{aligned} L(s_i, s_{i+1}) \\ &= \frac{1}{N_i}[\log p_B(s_i|s_{i+1}) - \log p_L(s_i)] \\ &+ \frac{1}{N_{i+1}}[\log p_B(s_{i+1}|s_i) - \log p_L(s_{i+1})] \end{aligned} \tag{3}$$

where $p_L(s)$ is the language model probability for generating sentence s. We train an LSTM language model, which can be thought of as a SEQ2SEQ model with an empty source. A closer look at Eq. 3 shows that it is of the same form as the mutual information between s_{i+1} and s_i, namely $\log[p(s_{i+1}, s_i)/p(s_{i+1})p(s_i)]$.

Generation The scoring functions in Eqs. 1, 2, and 3 are discriminative, generating coherence scores for an already-given chunk of text. Eqs. 2 and 3 can not be directly used for generation purposes, since they requires the completion of s_{i+1} before the score can be computed. A normal strategy is to generate a big N-best list using Eq. 1 and then rerank the N-best list using Eq. 2 or 3 (Li et al., 2015a). The N-best list can be generated using standard beam search, or other algorithmic variations that promote diversity, coherence, etc. (Shao et al., 2017).

Weakness (1) The SEQ2SEQ model generates words sequentially based on an evolving hidden vector, which is updated by combining the current word representation with previously built hidden vectors. The generation process is thus not exposed to more global features of the discourse like topics. As the hidden vector evolves, the influence from contexts gradually diminishes, with language models quickly dominating. (2) By predicting a sentence conditioning only on its left or right neighbor, the model lacks the ability to handle the longer-term discourse dependencies across the sentences of a text.

To tackle these two issues, we need a model that is able to constantly remind the decoder about the global meaning that it should convey at each word-generation step, a global meaning which can capture the state of the discourse across the sentences of a text. We propose two models of this global meaning, a pipelined approach based on HMM-based topic models (Blei et al., 2003; Gruber et al., 2007), and an end-to-end generative model with variational latent variables.

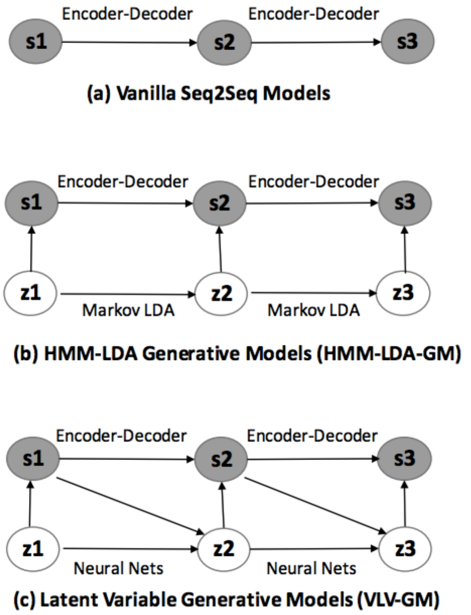

Figure 1: Overview of the proposed generative models for discourse coherence modeling.

3.2 HMM-LDA based Generative Models (HMM-LDA-GM)

In Markov topic models the topic depends on the previous topics in context (Ritter et al., 2010; Paul and Girju, 2010; Wang et al., 2011; Gruber et al., 2007; Paul, 2012). The topic for the current sentence is drawn based on the topic of the preceding sentence (or word) rather than on the global document-level topic distribution in vanilla LDA.

Our first model is a pipelined one (the *HMM-LDA-GM* in Fig. 1b), in which an HMM-LDA model provides the SEQ2SEQ model with global information for token generation, with two components:

(1) **Running HMM-LDA**: we first run a sentence-level HMM-LDA similar to Gruber et al. (2007). Our implementation forces all words in a sentence to be generated from the same topic, and this topic is sampled from a distribution based on the topic from previous sentence. Let t_n denote the distribution of topics for the current sentence, where $t_n \in \mathbb{R}^{1\times T}$. We also associate each LDA

topic with a K dimensional vector, representing the semantics embedded in this topic. The topic-representation matrix is denoted by $V \in \mathbb{R}^{T\times K}$, where T is the pre-specified number of topics in LDA. V is learned in the word predicting process when training encoder-decoder models.

(2) **Training encoder-decoder models**: For the current sentence s_n, given its topic distribution t_n, we first compute the topic representation z_n for s_n using the weighted sum of LDA topic vectors:

$$z_n = t_n \times V \tag{4}$$

z_n can be thought of as a discourse state vector that stores the information the current sentence needs to convey in the discourse, and is used to guide every step of word generation in s_n. We run the encoder-decoder model, which subsequently predicts tokens in s_n given s_{n-1}. This process is the same as the vanilla version of SEQ2SEQ models, the only difference being that z_n is incorporated into each step of decoding for hidden vector updates:

$$p(s_n|z_n, s_{n-1}) = \prod_{t=1}^{M} p(w_t|h_{t-1}, z_n) \tag{5}$$

V is updated along with parameters in the encoder-decoder model.

z_n influences each time step of decoding, and thus addresses the problem that vanilla SEQ2SEQ models gradually lose global information as the hidden representations evolve. z_n is computed based on the topic distribution t_n, which is obtained from the HMM-LDA model, thus modeling the global Markov discourse dependency between sentences of the text.[3] The model can be adapted to the bi-directional setting, in which we separately train two models to handle the forward probability $\log p(t_n|s_{n-1}, ...)$ and the backward one $\log p(t_n|s_{n+1})$. The bi-directional (*bi*) strategy described in Eq. 3 can also be incorporated to remove the influence of language models.

Weakness Topic models (either vanilla or HMM versions) focus on word co-occurrences at the document-level and are thus very lexicon-based. Furthermore, given the diversity of topics in a dataset like Wikipedia but the small number of topic clusters, the LDA model usually produces very coarse-grained topics (politics, sports, history, etc.), assigning very similar topic distributions to consecutive sentences. These topics thus capture topical coherence but are too coarse-grained to capture all the more fine-grained aspects of *discourse* coherence relationships.

3.3 Variational Latent Variable Generative Models (VLV-GM)

We therefore propose instead to train an end-to-end system, in which the meaning transitions between sentences can be naturally learned from the data. Inspired by recent work on generating sentences from a latent space (Serban et al., 2016b; Bowman et al., 2015; Chung et al., 2015), we propose the VSV-GM model in Fig. 1c. Each sentence s_n is again associated with a hidden vector representation $z_n \in \mathbb{R}^K$ which stores the global information that the current sentence needs to talk about, but instead of obtaining z_n from an upstream model like LDA, z_n is learned from the training data. z_n is a stochastic latent variable conditioned on all previous sentences and z_{n-1}:

$$\begin{aligned} &p(z_n|z_{n-1}, s_{n-1}, s_{n-2}, ...) = N(\mu^{\text{true}}_{z_n}, \Sigma^{\text{true}}_{z_n}) \\ &\mu^{\text{true}}_{z_n} = f(z_{n-1}, s_{n-1}, s_{n-2}, ...) \\ &\Sigma^{\text{true}}_{z_n} = g(z_{n-1}, s_{n-1}, s_{n-2}, ...) \end{aligned} \tag{6}$$

where $N(\mu, \Sigma)$ is a multivariate normal distribution with mean $\mu \in \mathbb{R}^K$ and covariance matrix $\Sigma \in \mathbb{R}^{K\times K}$. Σ is a diagonal matrix. As can be seen, the global information z_n for the current sentence depends on the information z_{n-1} for its previous sentence as well as the text of the context sentences. This forms a Markov chain across all sentences. f and g are neural network models that take previous sentences and z_{n-1}, and map them to a real-valued representation using hierarchical LSTMs (Li et al., 2015b)[4].

Each word w_{nt} from s_n is predicted using the concatenation of the representation previously build by the LSTMs (the same vector used in word prediction in vanilla SEQ2SEQ models) and z_n, as shown in Eq.5.

We are interested in the posterior distribution $p(z_n|s_1, s_2, ..., s_{n-1})$, namely, the information that the current sentence needs to convey given the preceding ones. Unfortunately, a highly non-linear mapping from z_n to tokens in s_n results in in-

[3] This pipelined approach is closely related to recent work that incorporates LDA topic information into generation models in an attempt to leverage context information (Ghosh et al., 2016; Xing et al., 2016; Mei et al., 2016)

[4] Sentences are first mapped to vector representations using a LSTM model. Another level of LSTM at the sentence level then composes representations of the multiple sentences to a single vector.

tractable inference of the posterior. A common solution is to use variational inference to learn another distribution, denoted by $q(z_n|s_1, s_2, ..., s_N)$, to approximate the true posterior $p(z_n|s_1, s_2, ..., s_{n-1})$. The model's latent variables are obtained by maximizing the variational lower-bound of observing the dataset:

$$\begin{aligned}&\log p(s_1, .., s_N) \leq \\ &\sum_{t=1}^{N} -D_{KL}(q(z_n|s_n, s_{n-1}, ...)||p(z_n|s_{n-1}, s_{n-2}, ...)) \\ &+ E_{q(z_n|s_n,s_{n-1},...)} \log p(s_n|z_n, s_{n-1}, s_{n-2}, ...)\end{aligned} \tag{7}$$

This objective to optimize consists of two parts; the first is the KL divergence between the approximate distribution q and the true posterior $p(s_n|z_n, s_{n-1}, s_{n-2}, ...)$, in which we want to approximate the true posterior using q. The second part $E_{q(z_n|s_n,s_{n-1},...)} \log p(s_n|z_n, s_{n-1}, s_{n-2}, ...)$, predicts tokens in s_n in the same way as in SEQ2SEQ models with the difference that it considers the global information z_n.

The approximate posterior distribution $q(z_n|s_n, s_{n-1}, ...)$ takes a form similar to $p(z_n|s_{n-1}, s_{n-2}, ...)$:

$$\begin{aligned}&q(z_n|s_n, s_{n-1}, ...) = N(\mu_{z_n}^{\text{approx}}, \Sigma_{z_n}^{\text{approx}}) \\ &\mu_{z_n}^{\text{approx}} = f_q(z_{n-1}, s_n, s_{n-1}, ...) \\ &\Sigma_{z_n}^{\text{approx}} = g_q(z_{n-1}, s_n, s_{n-1}, ...)\end{aligned} \tag{8}$$

f_q and g_q are of similar structures to f and g, using a hierarchical neural network model to map context tokens to vector representations.

Learning and Testing At training time, the approximate posterior $q(z_n|z_{n-1}, s_n, s_{n-1}, ...)$, the true distribution $p(z_n|z_{n-1}, s_{n-1}, s_{n-2}, ...)$, and the generative probability $p(s_n|z_n, s_{n-1}, s_{n-2}, ...)$ are trained jointly by maximizing the variational lower bound with respect to their parameters: a sample z_n is first drawn from the posterior distribution q, namely $N(\mu_{z_n}^{\text{approx}}, \Sigma_{z_n}^{\text{approx}})$. This sample is used to approximate the expectation $E_q \log p(s_n|z_n, s_{n-1}, s_{n-2}, ...)$. Using z_n, we can update the encoder-decoder model using SGD in a way similar to the standard SEQ2SEQ model, the only difference being that the current token to predict not only depends on the LSTM output h_t, but also z_n. Given the sampled z_n, the KL-divergence can be readily computed, and we update the model using standard gradient decent (details shown in the Appendix).

The proposed *VLV-GM* model can be adapted to the bi-directional setting and the *bi* setting similarly to the way LDA-based models are adapted.

The proposed model is closely related to many recent attempts in training variational autoencoders (VAE) (Kingma and Welling, 2013; Rezende et al., 2014), variational or latent-variable recurrent nets (Bowman et al., 2015; Chung et al., 2015; Ji et al., 2016; Bayer and Osendorfer, 2014), hierarchical latent variable encoder-decoder models (Serban et al., 2016b,a).

4 Experimental Results

In this section, we describe experimental results. We first evaluate the proposed models on discriminative tasks such as sentence-pair ordering and full paragraph ordering reconstruction. Then we look at the task of coherent text generation.

Model	Acci	Earthq	Aver
Discriminative Model	**0.930**	**0.992**	**0.956**
SEQ2SEQ (bi)	0.755	0.930	0.842
VLV-GM (bi)	0.770	0.931	0.851
Recursive	0.864	0.976	0.920
Entity Grid Model	0.904	0.872	0.888
HMM	0.822	0.938	0.880
HMM+Entity	0.842	0.911	0.876
HMM+Content	0.742	0.953	0.847
Graph	0.846	0.635	0.740
Foltz et al. (1998)-Glove	0.705	0.682	0.688
Foltz et al. (1998)-LDA	0.660	0.667	0.664

Table 1: Results from different coherence models. Results for the Recursive model is reprinted from Li and Hovy (2014), Entity Grid Model from Louis and Nenkova (2012), HMM, HMM+Entity and HMM+Content from Louis and Nenkova (2012), Graph from Guinaudeau and Strube (2013), and the final two lexical models are recomputed using Glove and LDA to replace the original LSA model of Foltz et al. (1998).

4.1 Sentence Ordering, Domain-specific Data

Dataset We first evaluate the proposed algorithms on the task of predicting the correct ordering of pairs of sentences predicated on the assumption that an article is always more coherent than a random permutation of its sentences (Barzilay and Lapata, 2008). A detailed description of this commonly used dataset and training/testing are found in the Appendix.

We report the performance of the following baselines widely used in the coherence literature.

(1) *Entity Grid Model*: The grid model presented in Barzilay and Lapata (2008). Results are directly taken from Barzilay and Lapata's (2008) paper. We also consider variations of entity grid models, such as Louis and Nenkova (2012) which models the

cluster transition probability and the *Graph Based Approach* which uses a graph to represent the entity transitions needed for local coherence computation (Guinaudeau and Strube, 2013).

(2) Li and Hovy (2014): A recursive neural model computes sentence representations based on parse trees. Negative sampling is used to construct negative incoherent examples. Results are from their papers.

(3) Foltz et al. (1998) computes the semantic relatedness of two text units as the cosine similarity between their LSA vectors. The coherence of a discourse is the average of the cosine of adjacent sentences. We used this intuition, but with more modern embedding models: (1) 300-dimensional Glove word vectors (Pennington et al., 2014), embeddings for a sentence computed by averaging the embeddings of its words (2) Sentence representations obtained from LDA (Blei et al., 2003) with 300 topics, trained on the Wikipedia dataset. Results are reported in Table 2. The extended version of the discriminative model described in this work significantly outperforms the parse-tree based recursive models presented in Li and Hovy (2014) as well as all non-neural baselines. It achieves almost perfect accuracy on the earthquake dataset and 93% on the accident dataset, marking a significant advancement in the benchmark. Generative models (both vanilla SEQ2SEQ and the proposed variational model) do not perform competitively on this dataset. We conjecture that this is due to the small size of the dataset, leading the generative model to overfit.

4.2 Evaluating Ordering on Open-domain

Since the dataset presented in Barzilay and Lapata (2008) is quite domain-specific, we propose testing coherence with a much larger, open-domain dataset: Wikipedia. We created a test set by randomly selecting 984 paragraphs from Wikipedia dump 2014, each paragraph consisting of at least 16 sentences. The training set is 30 million sentences not overlapping with the test set.

4.2.1 Binary Permutation Classification

We adopt the same strategy as in Barzilay and Lapata (2008), in which we generate pairs of sentence permutations from the original Wikipedia paragraphs. We follow the protocols described in the subsection and each pair whose original paragraph's score is higher than its permutation is treated as being correctly classified, else incorrectly classified. Models are evaluated using accuracy. We implement the Entity Grid Model (Barzilay and Lapata, 2008) using the Wikipedia training set as a baseline, the detail of which is presented in the Appendix. Other baselines consist of the Glove and LDA updates of the lexical coherence baselines (Foltz et al., 1998).

Model	Accuracy
VLV-GM (MMI)	**0.873**
VLV-GM (bi)	0.860
VLV-GM (uni)	0.839
LDA-HMM-GM (MMI)	0.847
LDA-HMM-GM (bi)	0.837
LDA-HMM-GM (uni)	0.814
SEQ2SEQ (MMI)	0.840
SEQ2SEQ (bi)	0.821
SEQ2SEQ (uni)	0.803
Discriminative Model	0.715
Entity Grid Model	0.686
Foltz et al. (1998)-Glove	0.597
Foltz et al. (1998)-LDA	0.575

Table 2: Performance on the open-domain binary classification dataset of 984 Wikipedia paragraphs.

Results Table 2 presents results on the binary classification task. Contrary to the findings on the domain specific dataset in the previous subsection, the discriminative model does not yield compelling results, performing only slightly better than the entity grid model. We believe the poor performance is due to the sentence-level negative sampling used by the discriminative model. Due to the huge semantic space in the open-domain setting, the sampled instances can only cover a tiny proportion of the possible negative candidates, and therefore don't cover the space of possible meanings. By contrast the dataset in Barzilay and Lapata (2008) is very domain-specific, and the semantic space is thus relatively small. By treating all other sentences in the document as negative, the discriminative strategy's negative samples form a much larger proportion of the semantic space, leading to good performance.

Generative models perform significantly better than all other baselines. Compared with the dataset in Barzilay and Lapata (2008), overfitting is not an issue here due to the great amount of training data. In line with our expectation, the *MMI* model outperforms the *bidirectional* model, which in turn outperforms the *unidirectional* model across all three generative model settings. We thus only report *MMI* results for experiments below. The *VLV-GM* model outperforms that the *LDA-HMM-GM* model, which is slightly better than the vanila SEQ2SEQ models.

Model	Accuracy
VLV-GM (MMI)	**0.256**
LDA-HMM-GM (MMI)	0.237
SEQ2SEQ (MMI)	0.226
Entity Grid Model	0.143
Foltz et al. (1998) (Glove)	0.084

Table 3: Performances of the proposed models on the open-domain paragraph reconstruction dataset.

Model	adver-1	adver-2	adver-3
VLV-GM (MMI)	**0.174**	**0.120**	**0.054**
LDA-HMM-GM (MMI)	0.130	0.104	0.043
SEQ2SEQ (MMI)	0.120	0.090	0.039
SEQ2SEQ (bi)	0.108	0.078	0.030
SEQ2SEQ (uni)	0.101	0.068	0.024

Table 4: Adversarial Success for different models.

4.2.2 Paragraph Reconstruction

The accuracy of our models on the binary task of detecting the original sentence ordering is very high, on both the prior small task and our large open-domain version. We therefore believe it is time for the community to move to a more difficult task for measuring coherence.

We suggest the task of *reconstructing an original paragraph from a bag of constituent sentences*, which has been previously used in coherence evaluation (Lapata, 2003). More formally, given a set of permuted sentences $s_1, s_2, ..., s_N$ (N the number of sentences in the original document), our goal is return the original (presumably most coherent) ordering of s.

Because the *discriminative model* calculates the coherence of a sentence given the known previous and following sentences, it cannot be applied to this task since we don't know the surrounding context. Hence, we only use the *generative model*. The first sentence of a paragraph is given: for each step, we compute the coherence score of placing each remaining candidate sentence to the right of the partially constructed document. We use beam search with beam size 10. We use the Entity Grid model as a baseline for both the settings.

Evaluating the absolute positions of sentences would be too harsh, penalizing orderings that maintain relative position between sentences through which local coherence can be manifested. We therefore use Kendall's τ (Lapata, 2003, 2006), a metric of rank correlation for evaluation. See the Appendix for details of Kendall's τ computation. We observe a pattern similar to the results on the binary classification task, where the *VLV-GM* model performs the best.

4.3 Adversarial evaluation on Text Generation Quality

Both the tasks above are discriminative ones. We also want to evaluate different models' ability to generate coherent text chunks. The experiment is set up as follow: each encoder-decoder model is first given a set of context sentences (3 sentences). The model then generates a succeeding sentence using beam-search given the contexts. For the *uni-directional* setting, we directly take the most probable sequence and for the *bi-directional* and *MMI*, we rerank the N-best list using the backward probability and language model probability.

We conduct experiments on multi-sentence generation, in which we repeat the generative process described above for N times, where N=1,2,3. At the end of each turn, the context is updated by adding in the newly generated sequence, and this sequence is used as the source input to the encoder-decoder model for next sequence generation. For example, when N is set to 2, given the three context sentences *context-a*, *context-b* and *context-c*, we first generate *sen-d* given the three context sentences and then generate *sen-e* given the sen-d, *context-a*, *context-b* and *context-c*.

For evaluation, standard word overlap metrics such as BLEU or ROUGE are not suited for our task, and we use adversarial evaluation Bowman et al. (2015); Anjuli and Vinyals (2016). In adversarial evaluation, we train a binary discriminant function to classify a sequence as machine generated or human generated, in an attempt to evaluate the model's sentence generation capability. The evaluator takes as input the concatenation of the contexts and the generated sentences (i.e., *context-a*, *context-b* and *context-c*, *sen-d* , *sen-e* in the example described above),[5] and outputs a scalar, indicating the probability of the current text chunk being human-generated. Training/dev/test sets are held-out sets from the one on which generative models are trained. They respectively contain 128,000/12,800/12,800 instances. Since discriminative models cannot generate sentences, and thus cannot be used for adversarial evaluation, they are skipped in this section.

We report Adversarial Success (*AdverSuc* for short), which is the fraction of instances in which a model is capable of fooling the evaluator. *Adver-*

[5] The model uses a hierarchical neural structure that first maps each sentence to a vector representation, with another level of LSTM on top of the constituent sentences, producing a single vector to represent the entire chunk of texts.

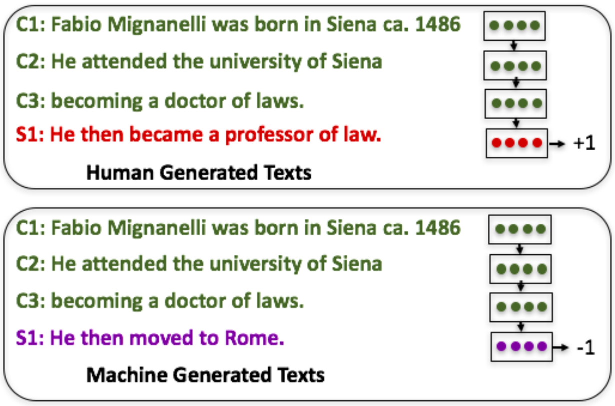
C1: Fabio Mignanelli was born in Siena ca. 1486
C2: He attended the university of Siena
C3: becoming a doctor of laws.
S1: He then became a professor of law.
+1
Human Generated Texts
C1: Fabio Mignanelli was born in Siena ca. 1486
C2: He attended the university of Siena
C3: becoming a doctor of laws.
S1: He then moved to Rome.
-1
Machine Generated Texts

Figure 2: An overview of training the adversarial evaluator using a hierarchical neural model. Green denotes input contexts. Red denotes a sentence from human-generated texts, treated as a positive example. Purple denotes a sentence from machine-decoded texts, treated as a negative example.

Suc is the difference between 1 and the accuracy achieved by the evaluator. Higher values of *AdverSuc* for a dialogue generation model are better. *AdverSuc-N* denotes the adversarial accuracy value on machine-generated texts with N turns.

Table 4 show *AdverSuc* numbers for different models. As can be seen, the latent variable model VLV-GM is able to generate chunk of texts that are most indistinguishable from coherent texts from humans. This is due to its ability to handle the dependency between neighboring sentences. Performance declines as the number of turns increases due to the accumulation of errors and current models' inability to model long-term sentence-level dependency. All models perform poorly on the *adver-3* evaluation metric, with the best adversarial success value being 0.081 (the trained evaluator is able to distinguish between human-generated and machine generated dialogues with greater than 90 percent accuracy for all models).

4.4 Qualitative Analysis

With the aim of guiding future investigations, we also briefly explore our model qualitatively, examining the coherence scores assigned to some artificial miniature discourses that exhibit various kinds of coherence.

Case 1: Lexical Coherence

Pinochet was arrested. His arrest was unexpected. **1.79**
Pinochet was arrested. His death was unexpected. **0.84**
Mary ate some apples. She likes apples. **2.03**
Mary ate some apples. She likes pears. **0.27**
Mary ate some apples. She likes Paris. **-1.35**

The examples suggest that the model handles lexical coherence, correctly favoring the 1st over the 2nd, and the 3rd over the 4th examples. Note that the coherence score for the final example is negative, which means conditioning on the first sentence actually decreases the likelihood of generating the second one.

Case 2: Temporal Order

Washington was unanimously elected president in the first two national elections. He oversaw the creation of a strong, well-financed national government. **1.48**
Washington oversaw the creation of a strong, well-financed national government. He was unanimously elected president in the first two national elections. **0.72**

Case 3: Causal Relationship

Bret enjoys video games; therefore, he sometimes is late to appointments. **0.69**
Bret sometimes is late to appointments; therefore, he enjoys video games. **-0.07**

Cases 2 and 3 suggest the model may, at least in these simple cases, be capable of addressing the much more complex task of dealing with temporal and causal relationships. Presumably this is because the model is exposed in training to the general preference of natural text for temporal order, and even for the more subtle causal links.

Case 4: Centering/Referential Coherence

Mary ate some apples. She likes apples. **3.06**
She ate some apples. Mary likes apples. **2.41**

The model seems to deal with simple cases of referential coherence.

Example3: **2.40**
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.
Example4: **1.62**
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.

In these final examples from Miltsakaki and Kukich (2004), the model successfully captures the fact that the second text is less coherent due to *rough shifts*. This suggests that the discourse embedding space may be able to capture a representation of entity focus.

Of course all of these these qualitative evaluations are only suggestive, and a deeper understanding of what the discourse embedding space is capturing will likely require more sophisticated visualizations.

5 Conclusion

We investigate the problem of open-domain discourse coherence, training discriminative models that treating natural texts as coherent and permutations as non-coherent, and Markov generative models that can predict sentences given their neighbors.

Our work shows that the traditional evaluation metric (ordering pairs of sentences in small domains) is completely solvable by our discriminative models, and we therefore suggest the community move to the harder task of open-domain full-paragraph sentence ordering.

The proposed models also offer an initial step in generating coherent texts given contexts, which has the potential to benefit a wide range of generation tasks in NLP. Our latent variable neural models, by offering a new way to learn latent discourse-level features of a text, also suggest new directions in discourse representation that may bring benefits to any discourse-aware NLP task.

Acknowledgements The authors thank Will Monroe, Sida Wang, Kelvin Guu and the other members of the Stanford NLP Group for helpful discussions and comments. Jiwei Li is supported by a Facebook Fellowship, which we gratefully acknowledge. This work is also partially supported by the NSF under award IIS-1514268, and the DARPA Communicating with Computers (CwC) program under ARO prime contract no. W911NF- 15-1-0462. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA, the NSF, or Facebook.

References

Ernst Althaus, Nikiforos Karamanis, and Alexander Koller. 2004. Computing locally coherent discourses. In *Proceedings of ACL 2004*.

Kannan Anjuli and Oriol Vinyals. 2016. Adversarial evaluation of dialogue models. *NIPS 2016 Workshop on Adversarial Training* .

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proc. of the International Conference on Learning Representations (ICLR)*.

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

Regina Barzilay and Lillian Lee. 2004. Catching the drift: Probabilistic content models, with applications to generation and summarization. In *HLT-NAACL*. pages 113–120.

Regina Barzilay and Kathleen R McKeown. 2005. Sentence fusion for multidocument news summarization. *Computational Linguistics* 31(3):297–328.

Justin Bayer and Christian Osendorfer. 2014. Learning stochastic recurrent networks. *arXiv preprint arXiv:1411.7610* .

Gillinder Bedi, Facundo Carrillo, Guillermo A Cecchi, Diego Fernández Slezak, Mariano Sigman, Natália B Mota, Sidarta Ribeiro, Daniel C Javitt, Mauro Copelli, and Cheryl M Corcoran. 2015. Automated analysis of free speech predicts psychosis onset in high-risk youths. *npj Schizophrenia* 1.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3(Jan):993–1022.

Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. 2015. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349* .

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

Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio. 2015. A recurrent latent variable model for sequential data. In *Advances in neural information processing systems*. pages 2980–2988.

Carl Doersch. 2016. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908* .

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research* 12:2121–2159.

Micha Eisner 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: short papers-Volume 2*. Association for Computational Linguistics, pages 125–129.

Micha Elsner, Joseph L Austerweil, and Eugene Charniak. 2007. A unified local and global model for discourse coherence. In *HLT-NAACL*. pages 436–443.

Micha Elsner and Eugene Charniak. 2008. Coreference-inspired coherence modeling. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human*

Language Technologies: Short Papers. Association for Computational Linguistics, pages 41–44.

Brita Elvevåg, Peter W Foltz, Daniel R Weinberger, and Terry E Goldberg. 2007. Quantifying incoherence in speech: An automated methodology and novel application to schizophrenia. *Schizophrenia research* 93(1):304–316.

Vanessa Wei Feng and Graeme Hirst. 2012. Extending the entity-based coherence model with multiple ranks. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, pages 315–324.

Peter W Foltz. 2007. Discourse coherence and lsa. *Handbook of latent semantic analysis* pages 167–184.

Peter W Foltz, Walter Kintsch, and Thomas K Landauer. 1998. The measurement of textual coherence with latent semantic analysis. *Discourse processes* 25(2-3):285–307.

Shalini Ghosh, Oriol Vinyals, Brian Strope, Scott Roy, Tom Dean, and Larry Heck. 2016. Contextual lstm (clstm) models for large scale nlp tasks. *arXiv preprint arXiv:1602.06291* .

Amit Gruber, Yair Weiss, and Michal Rosen-Zvi. 2007. Hidden topic markov models. In *AISTATS*. volume 2, pages 163–170.

Camille Guinaudeau and Michael Strube. 2013. Graph-based local coherence modeling. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*. pages 93–103.

M. A. K. Halliday and Ruqaiya Hasan. 1976. *Cohesion in English*. Longman.

Eduard H Hovy. 1988. Planning coherent multisentential text. In *Proceedings of the 26th annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics, pages 163–169.

Yangfeng Ji, Gholamreza Haffari, and Jacob Eisenstein. 2016. A latent variable recurrent neural network for discourse relation language models. *arXiv preprint arXiv:1603.01913* .

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* .

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in Neural Information Processing Systems*. pages 3276–3284.

Mirella Lapata. 2003. Probabilistic text structuring: Experiments with sentence ordering. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*. Association for Computational Linguistics, pages 545–552.

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

Alex Lascarides and Nicholas Asher. 1991. Discourse relations and defeasible knowledge. In *Proceedings of the 29th annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics, pages 55–62.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2015a. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055* .

Jiwei Li and Eduard Hovy. 2014. A model of coherence based on distributed sentence representation. In *Proceedings of Empirical Methods in Natural Language Processing*.

Jiwei Li, Minh-Thang Luong, and Dan Jurafsky. 2015b. A hierarchical neural autoencoder for paragraphs and documents. *arXiv preprint arXiv:1506.01057* .

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-Volume 1*. Association for Computational Linguistics, pages 997–1006.

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

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *EMNLP* .

William C Mann and Sandra A Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text* 8(3):243–281.

Daniel Marcu. 1997. From local to global coherence: A bottom-up approach to text planning. In *AAAI/IAAI*. Citeseer, pages 629–635.

Danielle S. McNamara, Max M. Louwerse, Philip M. McCarthy, and Arthur C. Graesser. 2010. Coh-metrix: Capturing linguistic features of cohesion. *Discourse Processes* 47(4):292–330.

Hongyuan Mei, Mohit Bansal, and Matthew R Walter. 2016. Coherent dialogue with attention-based language models. *arXiv preprint arXiv:1611.06997* .

Eleni Miltsakaki and Karen Kukich. 2004. Evaluation of text coherence for electronic essay scoring systems. *Natural Language Engineering* 10(01):25–55.

J. Morris and G. Hirst. 1991. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics* 17(1):21–48.

Michael Paul and Roxana Girju. 2010. A two-dimensional topic-aspect model for discovering multi-faceted topics. *Urbana* 51(61801):36.

Michael J Paul. 2012. Mixed membership markov models for unsupervised conversation modeling. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Association for Computational Linguistics, pages 94–104.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*. volume 14, pages 1532–1543.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 2014. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082* .

Alan Ritter, Colin Cherry, and Bill Dolan. 2010. Unsupervised modeling of twitter conversations. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, pages 172–180.

Iulian Vlad Serban, Tim Klinger, Gerald Tesauro, Kartik Talamadupula, Bowen Zhou, Yoshua Bengio, and Aaron Courville. 2016a. Multiresolution recurrent neural networks: An application to dialogue response generation. *arXiv preprint arXiv:1606.00776* .

Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron Courville, and Yoshua Bengio. 2016b. A hierarchical latent variable encoder-decoder model for generating dialogues. *arXiv preprint arXiv:1605.06069* .

Louis Shao, Stephan Gouws, Denny Britz, Anna Goldie, Brian Strope, and Ray Kurzweil. 2017. Generating long and diverse responses with neural conversation models. *arXiv preprint arXiv:1701.03185* .

Ilya Sutskever, Oriol Vinyals, and Quoc Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*. pages 3104–3112.

Stephen Tratz and Eduard Hovy. 2011. A fast, accurate, non-projective, semantically-enriched parser. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 1257–1268.

Suzan Verberne, Lou Boves, Nelleke Oostdijk, and Peter-Arno 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*. ACM, pages 735–736.

Hongning Wang, Duo Zhang, and ChengXiang Zhai. 2011. Structural topic model for latent topical structure analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, pages 1526–1535.

Chen Xing, Wei Wu, Yu Wu, Jie Liu, Yalou Huang, Ming Zhou, and Wei-Ying Ma. 2016. Topic augmented neural response generation with a joint attention mechanism. *arXiv preprint arXiv:1606.08340* .

Matthew D Zeiler. 2012. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701* .

6 Supplemental Material

Details for the domain specific dataset (Barzilay and Lapata, 2008) The corpus consists of 200 articles each from two domains: NTSB airplane accident reports (V=4758, 10.6 sentences/document) and AP earthquake reports (V=3287, 11.5 sentences/document), split into training and testing. For each document, pairs of permutations are generated[6]. Each pair contains the original document order and a random permutation of the sentences from the same document.

Training/Testing details for models on the domain specific dataset We use reduced versions of both generative and discriminative models to allow fair comparison with baselines. For the discriminative model, we generate noise negative examples from random replacements in the training set, with the only difference that random replacements only come from the same document. We use 300 dimensional embeddings borrowed from GLOVE (Pennington et al., 2014) to initialize word embeddings. Word embeddings are kept fixed during training and we update LSTM parameters using AdaGrad (Duchi et al., 2011). For the generative model, due to the small size of the dataset, we train a one layer SEQ2SEQ model with word dimensionality and number of hidden neurons set to 100. The model is trained using SGD with AdaGrad (Zeiler, 2012).

The task requires a coherence score for the whole document, which is comprised of multiple cliques. We adopt the strategy described in Li and Hovy (2014) by breaking the document into a series of cliques which is comprised of a sequence of

[6] Permutations downloaded from `people.csail.mit.edu/regina/coherence/CLsubmission/`.

consecutive sentences. The document-level coherence score is attained by averaging its constituent cliques. We say a document is more coherent if it achieves a higher average score within its constituent cliques.

Implementation of Entity Grid Model For each noun in a sentence, we extract its syntactic role (subject, object or other). We use a wikipedia dump parsed using the Fanse Parser (Tratz and Hovy, 2011). Subjects and objects are extracted based on *nsubj* and *dobj* relations in the dependency trees. (Barzilay and Lapata, 2008) define two versions of the Entity Grid Model, one using full coreference and a simpler method using only exact-string coreference; Due to the difficulty of running full coreference resolution tens of millions of Wikipedia sentences, we follow other researchers in using Barzilay and Lapata's simpler method (Feng and Hirst, 2012; Burstein et al., 2010; Barzilay and Lapata, 2008).[7]

Kendall's τ Kendall's τ is computed based on the number of inversions in the rankings as follows:

$$\tau = 1 - \frac{2\text{\# of inversions}}{N \times (N-1)} \tag{9}$$

where N denotes the number of sentences in the original document and inversions denote the number of interchanges of consecutive elements needed to reconstruct the original document. Kendall's τ can be efficiently computed by counting the number of intersections of lines when aligning the original document and the generated document. We refer the readers to Lapata (2003) for more details.

Derivation for Variation Inference For simplicity, we use μ_{post} and Σ_{approx} to denote $\mu^{\text{approx}}(z_n)$ and $\Sigma^{\text{approx}}(z_n)$, μ_{true} and Σ_{true} to denote $\mu^{\text{true}}(z_n)$ and $\Sigma^{\text{true}}(z_n)$. The KL-divergence between the approximate distribution $q(z_n|z_{n-1}, s_n, s_{n-1}, ...)$ and the true distribution $p(z_n|z_{n-1}, s_{n-1}, s_{n-2}, ...)$ in the variational inference is given by:

$$\begin{aligned}
&D_{KL}(q(z_n|z_{n-1}, s_n, s_{n-1}, ...)||p(z_n|z_{n-1}, s_{n-1}, s_{n-2}, ...) \\
=&\frac{1}{2}(\text{tr}(\Sigma_{true}^{-1}\Sigma_{approx}) - k + \log\frac{\det\Sigma_{true}}{\det\Sigma_{approx}} \\
+&(\mu_{true} - \mu_{approx})^{-1}\Sigma_{true}^{-1}(\mu_{true} - \mu_{approx}))
\end{aligned} \tag{10}$$

where k denotes the dimensionality of the vector. Since z_n has already been sampled and thus known, μ_{approx}, Σ_{approx}, μ_{true}, Σ_{true} and consequently Eq10 can be readily computed. The gradient with respect to μ_{approx}, Σ_{approx}, μ_{true}, Σ_{true} can be respectively computed, and the error is then backpropagated to the hierarchical neural models that are used to compute them. We refer the readers to Doersch (2016) for more details about how a general VAE model can be trained.

Our generate models offer a powerful way to represent the latent discourse structure in a complex embedding space, but one that is hard to visualize. To help understand what the model is doing, we examine some relevant examples, annotated with the (log-likelihood) coherence score from the *MMI* generative model, with the goal of seeing (qualitatively) the kinds of coherence the model seems to be representing. (The MMI can be viewed as the informational gain from conditioning the generation of the current sentence on its neighbors.)

[7] Our implementation of the Entity Grid Model is built upon public available code at https://github.com/karins/CoherenceFramework.