# A Spatial Model for Extracting and Visualizing Latent Discourse Structure in Text

**Shashank Srivastava**[*]
Carnegie Mellon University
Pittsburgh, PA 15213, USA
ssrivastava@cmu.edu

**Nebojsa Jojic**
Microsoft Research
Redmond, WA 98052, USA
jojic@microsoft.com

## Abstract

We present a generative probabilistic model of documents as sequences of sentences, and show that inference in it can lead to extraction of long-range latent discourse structure from a collection of documents. The approach is based on embedding sequences of sentences from longer texts into a 2- or 3-D spatial grids, in which one or two coordinates model smooth topic transitions, while the third captures the sequential nature of the modeled text. A significant advantage of our approach is that the learned models are naturally visualizable and interpretable, as semantic similarity and sequential structure are modeled along orthogonal directions in the grid. We show that the method can capture discourse structures in narrative text across multiple genres, including biographies, stories, and newswire reports. In particular, our method can capture biographical templates from Wikipedia, and is competitive with state-of-the-art generative approaches on tasks such as predicting the outcome of a story, and sentence ordering.

## 1 Introduction

The ability to identify discourse patterns and narrative themes from language is useful in a wide range of applications and data analysis. From a perspective of language understanding, learning such latent structure from large corpora can provide background information that can aid machine reading. For example, computers can use such knowledge to predict what is likely to happen next in a narrative (Mostafazadeh et al., 2016), or reason about which narratives are coherent and which do not make sense (Barzilay and Lapata, 2008). Similarly, knowledge of discourse is increasingly important for language generation models. Modern neural generation models, while good at capturing surface properties of text – by fusing elements of syntax and style – are still poor at modeling long range dependencies that go across sentences (Li and Jurafsky, 2017; Wang et al., 2017). Models of long range flow in the text can thus be useful as additional input to such methods.

Previously, the question of modeling discourse structure in language has been explored through several lenses, including from perspectives of linguistics, cognitive science and information retrieval. Prominent among linguistic approaches are Discourse Representation Theory (Asher, 1986) and Rhetorical Structure Theory (Mann and Thompson, 1988); which formalize how discourse context can constrain the semantics of a sentence, and lay out ontologies of discourse relation types between parts of a document. This line of research has been largely constrained by the unavailability of corpora of discourse relations, which are expensive to annotate. Another line of research has focused on the task of automatic *script induction*, building on earlier work in the 1970's (Schank and Abelson, 1977). More recently, methods based on neural distributed representations have been explored (Li and Hovy, 2014; Kalchbrenner and Blunsom, 2013; Le and Mikolov, 2014) to model the flow of discourse. While these methods have had varying degrees of success, they are largely opaque and hard to interpret. In this work, we seek to provide a scalable model that can extract latent sequential structures from a collection of documents, and can be naturally visualized to provide a summary of the learned semantics and discourse trajectories.

In this work, we present an approach for extract-

[*]Work done while first author was an intern at Microsoft Research

*Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Long Papers)*, pages 2268–2277
Melbourne, Australia, July 15 - 20, 2018.

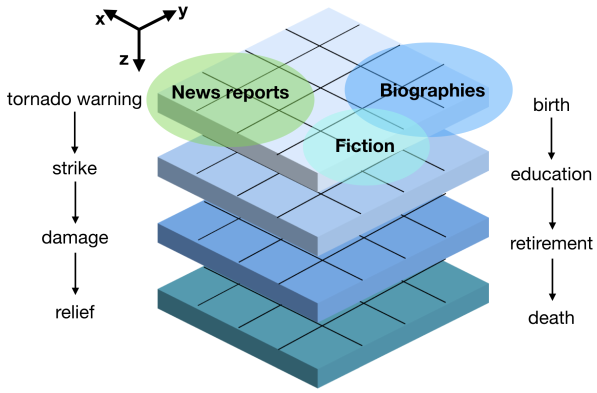


Figure 1: Modeling principle for Sequential Counting Grids. We design the method to capture semantic similarities between documents along XY planes (e.g., biographies might be more similar to literary fiction than news reports), as well extract sequential trajectories along the Z axes similar to those shown. The sequence of sentences in a document is latently aligned to positions in the grid, such that the model prefers alignments of contiguous sentences to grid cells that are spatially close.

ing and visualizing sequential structure from a collection of text documents. Our method is based on embedding sentences in a document in a 3-dimensional grid, such that contiguous sentences in the document are likely to be embedded in the same order in the grid. Further, sentences across documents that are semantically similar are also likely to be embedded in the same neighborhood in the grid. By leveraging the sequential order of sentences in a large document collection, the method can induce lexical semantics, as well as extract latent discourse trajectories in the documents. Figure 1 shows a conceptual schematic of our approach. The method can learn semantic similarity (across XY planes), as well as sequential discourse chains (along the Z-axis). The parameters and latent structure of the grid are learned by optimizing the likelihood of a collection of documents under a generative model. Our method outperforms state-of-the-art generative methods on two tasks: predicting the outcome of a story and coherence prediction; and is seen to yield a flexible range of interpretable visualizations in different domains of text. Our method is scalable, and can incorporate a broad range of features. In particular, the approach can work on simple tokenized text.

The remainder of this paper is organized as follows. In Section 2, we briefly summarize other related work. In Section 3, we describe our method in detail. We present experimental results in Section 4, and conclude with a brief discussion.

## 2 Related work

Building on linguistic theories of discourse and text coherence, several computational approaches have attempted to model discourse structure from multiple perspectives. Prominent among these are Narrative Event Chains (Chambers and Jurafsky, 2008) which learn chains of events that follow a pattern in a unsupervised framework, and the Entity grid model (Barzilay and Lapata, 2008), which represents sentences in a context in terms of discourse entities occurring in them and trains coherence classifiers over this representation. Other work extends these using better models of events and discourse entities (Lin et al., 2011; Pichotta and Mooney, 2015). Louis and Nenkova (2012) use manually provided syntactic patterns for sentence representation, and model transitions in text as Markov probabilities, which is related to our work. However, while they use simple HMMs over discrete topics, our method allows for a richer model that also captures smooth transition across them. Approaches such as Kalchbrenner and Blunsom (2013); Li et al. (2014); Li and Jurafsky (2017) model text through recurrent neural architectures, but are hard to interpret and visualize. Other approaches have explored applications related to modeling narrative discourse in context of limited tasks such as story cloze (Mostafazadeh et al., 2016) and identifying similar narratives (Chaturvedi et al., 2018).

From a large scale document-mining perspective, the question of extracting intra-document structure remains largely underexplored. While early models such as LDA completely ignore ordering and discourse elements of a documents, other methods that use distributed embeddings of documents are opaque (Le and Mikolov, 2014), even while they can in principle model sequential structure within a document. Methods such as HMM-LDA (Griffiths et al., 2005) and Topics-over-time (Wang and McCallum, 2006) address the related question of topic evolution in a stream of documents, but these approaches are too coarse to model intra-document sequential structure. In terms of our technical approach, we build on previous research on grid-based models (Jojic and Perina, 2011), which have previously been used for topic-modeling for images and text as unstructured bags-of-features.

# 3 Sequential CG model

In this section, we present our method, which we call **Sequential Counting Grids**, or **Sequential CG**. We first present our notation, model formulation and training approach. We discuss how the method is designed to incorporate smoothness and sequential structure, and how the method can be efficiently scaled to train on large document collections. In Section 3.2, we present a mixture model variant that combines Sequential CG with a unigram language model.

## 3.1 Model description

We represent a document as a sequence $\mathbf{s}$ of sentences, $\mathbf{s} = \{s_1, s_2 \dots s_D\}$, where $D$ represents the number of sentences in the document. In general, we assume each sentence is represented as a multiset of features $s_i = \{c_z\}_i$, where $c_z^i$ represents the count of the feature indexed by $z$ in the $i$th sentence in the sequence.[1]

The Sequential CG consists of a 3-D grid $G$ of size $E_x \times E_y \times E_z$, where $E_x$, $E_y$ and $E_z$ denote the extent of the grid along the X, Y and Z-axes (see Figure 1). Let us denote an index of a position in the grid by an integer-valued vector $\mathbf{i} = (i_x i_y i_z)$. The three components of the index together specify a XY location as well as a depth in the grid. The Sequential CG model is parametrized by two sets of parameters, $\pi_{\mathbf{i},z}$ and $\mathcal{P}_{\mathbf{ij}}$. Here, $\pi_{\mathbf{i},z}$ represents a multinomial distribution over the vocabulary of features $z$ for each cell in the grid $G$, i.e. $\sum_z \pi_{\mathbf{i},z} = 1 \ \forall \ \mathbf{i} \in G$. To induce smoothness across XY planes, we further define histogram distributions $h_{\mathbf{i},z}$, which average the $\pi$ distributions in a 2-D neighborhood $W_\mathbf{i}$ (of size specified by $W = [W_x, W_y]$) around the grid position $\mathbf{i}$. This notation follows Jojic and Perina (2011).

$$h_{\mathbf{i},z} = \frac{1}{W_x W_y} \sum_{\mathbf{i}' \in W_\mathbf{i}} \pi_{\mathbf{i}',z} \qquad (1)$$

The generative model assumes that individual sentences in a document are generated by $h$ distributions in the grid. Movements from one position $\mathbf{i}$ to another $\mathbf{j}$ in the grid are modeled as transition probabilities $\mathcal{P}_{\mathbf{ij}}$. The generative process consists of the following. We uniformly sample a starting location $\mathbf{i}_1$ in the grid. We sample words in the first sentence $s_1$ from $\pi_{\mathbf{i}1}$, and sample the next position $\mathbf{i}_2$ from the distribution $\mathcal{P}_{\mathbf{i}_1,:}$, and so on till we generate $s_D$. The alignments $\mathcal{I} = [\mathbf{i}_1, \mathbf{i}_2 \dots \mathbf{i}_D]$ of individual sentences in a document with positions in the grid are latent variables in our model.

Given the sequence of alignments $\mathcal{I}$ for a document, the conditional likelihood of generating $s$ is given as a product of generating individual sentences:

$$p(s|\, \mathcal{I}) = \prod_{d}^{D} p(\{c_z^d\}|\, \mathbf{i}_d) = \prod_{d=1}^{D} \prod_{z} (h_{\mathbf{i}_d,z})^{c_z^d} \qquad (2)$$

Since the alignments of sequences to their positions in the grids $\mathcal{I}$ are latent, we marginalize over these to maximize the likelihood of an observed collection of documents $\mathcal{S} := \{\mathbf{s}^t\}_{t=1}^{T}$. Here, $T$ is the total number of documents, and $t$ is an index over individual documents. Using Jensen's inequality, *any* distributions $q_\mathcal{I}^t$ over the hidden alignments $\mathcal{I}^t$ provide lower-bounds on the data log-likelihood.

$$\begin{aligned} \sum_t \log p(\mathbf{s}_t|\pi) &= \sum_t \log \Big( \sum_{\mathcal{I}} p(\mathbf{s}_t, \mathcal{I}|\pi) \Big) \\ &= \sum_t \log \Big( \sum_{\mathcal{I}} q_\mathcal{I}^t \frac{p(\mathbf{s}_t|\mathcal{I}) p(\mathcal{I}))}{q_\mathcal{I}^t} \Big) \\ &\geq - \sum_t \sum_{\mathcal{I}} q_\mathcal{I}^t \log q_\mathcal{I}^t \\ &\quad + \sum_t \sum_{\mathcal{I}} q_\mathcal{I}^t \log \big( p(s|\mathcal{I}, \pi) p(\mathcal{I}) \big) \end{aligned} \qquad (3)$$

Here, $q_\mathcal{I}^t$ denotes a variational distribution for each of the data sequences $\mathbf{s}_t$. The learning algorithm consists of an iterative generalized EM procedure (which can be interpreted as a block-coordinate ascent in the latent variables $q_\mathcal{I}^t$ and the model parameters $\pi$ and $\mathcal{P}$). We maximize the lower bound in Eqn 3 exactly by setting $q_\mathcal{I}^t$ to the posterior distribution of the data for the current values of the parameters $\pi$ (standard E step). Thus, we have

$$\begin{aligned} q_\mathcal{I}^t &\propto p(\mathbf{s}|\mathcal{I}) p(\mathcal{I}) \\ &= \Big[ \prod_{d=1}^{D} \prod_{z} (h_{\mathbf{i}_d,z})^{c_z^d(t)} \Big] \Big[ \prod_{d=2}^{D} \mathcal{P}_{\mathbf{i}_{d-1},\mathbf{i}_d} \Big] \end{aligned} \qquad (4)$$

We do not need to explicitly compute the posterior distribution $q_\mathcal{I}^t = p(\mathcal{I}|\mathbf{s})$ at this point, but only use it to compute the relevant expectation statistics in the M-step. This can be done efficiently, as we

[1] These may simply consist of tokens (words, entities and MWEs) in the sentence, but can include additional information, such as sentiment or event annotations, or other discrete sentence-level representations

see next. In the M-step, we consider $q^t_{\mathcal{I}}$ as fixed, and maximize the objective in terms of the model parameters $\pi$. Substituting this in Eqn 3, and focusing on terms that depend on the model parameters ($\pi$ and $\mathcal{P}$), we get

$$\begin{aligned}\mathcal{L}(\pi,\mathcal{P}) &\geq \sum_t \sum_{\mathcal{I}} q^t_{\mathcal{I}} \log\big(p(s|\mathcal{I},\pi)p(\mathcal{I})\big) + \mathcal{H}_q \\ &= \sum_t \sum_{\mathcal{I}} q^t_{\mathcal{I}} \Big( \sum_d \sum_z c^d_z(t) \log h_{\mathbf{i}_d,z} \\ &\qquad + \sum_d \log \mathcal{P}_{\mathbf{i}_{d-1},\mathbf{i}_d} \Big) \\ &= \sum_t \sum_{\mathcal{I}} \mathbb{E}_{q^t_{\mathcal{I}}} \Big[ \sum_d \sum_z \mathbb{I}_{\mathbf{i}^t_d=\mathbf{i}} c^d_z(t) \log h_{\mathbf{i}_d,z} \Big] \\ &\quad + \sum_t \sum_{\mathcal{I}} \mathbb{E}_{q^t_{\mathcal{I}}} \Big[ \sum_d \mathbb{I}_{\mathbf{i}^t_{d-1}=\mathbf{i},\mathbf{i}^t_d=\mathbf{j}} \log \mathcal{P}_{\mathbf{ij}} \Big]\end{aligned} \tag{5}$$

Maximizing the likelihood w.r.t. $\mathcal{P}$ leads to the following updates for the transition probabilities:[2]

$$\mathcal{P}_{\mathbf{ij}} = \frac{\sum_t \sum_d P(\mathbf{i}^t_{d-1} = \mathbf{i}, \mathbf{i}^t_d = \mathbf{j})}{\sum_t \sum_d P(\mathbf{i}^t_{d-1} = \mathbf{i})} \tag{6}$$

Here, the pairwise state-probabilities $P(\mathbf{i}^t_{d-1} = \mathbf{i}, \mathbf{i}^t_d = \mathbf{j})$ for adjacent sentences in a sequence can be efficiently calculated using the Forward-Backward algorithm. In Equation 5, rewriting the term containing $h$ in terms of $\pi$ using Eqn 1 (and ignoring constant terms $W_x W_y$), we get:

$$\begin{aligned}&\sum_t \sum_{\mathcal{I}} \mathbb{E}_{q^t_{\mathcal{I}}} \Big[ \sum_d \sum_z \mathbb{I}_{\mathbf{i}^t_d=\mathbf{i}} c^d_z(t) \log \sum_{\mathbf{i}' \in W_{\mathbf{i}}} \pi_{\mathbf{i}',z} \Big] \\ &= \sum_t \sum_{\mathcal{I}} \sum_d P(\mathbf{i}^t_d = \mathbf{i}) \sum_z c^d_z(t) \log \sum_{\mathbf{i}' \in W_{\mathbf{i}}} \pi_{\mathbf{i}',z}\end{aligned} \tag{7}$$

The presence of a summation inside of a logarithm makes maximizing this objective for $\pi$ harder. For this, we simply use Jensen's inequality introducing an additional variational distribution (for the latent grid positions within window $W_{\mathbf{i}}$ ), and maximize the lower bound. The final M-step update for $\pi$ becomes:

$$\pi_{\mathbf{i},z} \propto \Big( \sum_t \sum_d c^d_z(t) \sum_{\mathbf{k}|\mathbf{i} \in W_k} \frac{P(\mathbf{i}^t_d = \mathbf{k})}{h_{\mathbf{k},z}} \Big) \pi_{\mathbf{i},z} \tag{8}$$

As before, the state-probabilities $P(\mathbf{i}^t_d = \mathbf{i})$ can be computed using the Forward Backward algorithm.

Intuitively, the expected alignments in the E-step are distributions over sequences of positions in the grid that best explain the structure of documents for the current value of Sequential CG parameters. In the M-step, we assume these distributions embedding documents into various parts of the grid as given, and update the multinomial parameters and transition probabilities. Modeling the transitions as having a Markov property allows us to use a dynamic programming approach (Forward Backward algorithm) to exactly compute the posterior probabilities required for parameter updates. We note that at the onset of the procedure, we need to initialize $\pi$ randomly to break symmetries. Unless otherwise stated, in all experiments, we run EM to 200 iterations.

**Correlating space with sequential structure:** The use of histogram distributions $h$ to generate data forces smoothness in the model along XY planes due to adjacent cells in the grid sharing a large number of parameters that contribute to their histograms (due to overlapping windows). On the other hand, in order to induce spatial proximity in the grid to mimic the sequential flow of discourse in documents, we constrain the transition matrix $\mathcal{P}$ (which specifies transition preferences from one position in the grid to another) to a sparse banded matrix. In particular, a position $\mathbf{i} = (i_x, i_y, i_z)$ in the grid can only transition to itself, its 4 neighbors in the same XY plane, and two cells in the succeeding two layers along the Z-axis ( $(i_x, i_y, i_{z+1})$ and $(i_x, i_y, i_{z+2})$). This is enforced by fixing other elements in the transition matrix to 0, and only updating allowable transitions.

As an important note about implementation details, we observe here that the Forward-Backward procedure (which is repeatedly invoked during model training) can be naturally formulated in terms of matrix operations.[3] This allows training for the Sequential CG approach to be scalable for large document collections.

In our formulation, we have presented a Sequential CG model for a 3-D grid. This can be adapted to learn 2-D grids (trellis) by setting $E_y = 1$. In our experiments, we found 3-D grids to be better

[2] Since the optimal value for the concave problem $\sum_j y_j \log x_j$ s.t. $\sum_j x_j = 1$ occurs when $x^*_j \propto y_j$

[3] To explain, if $f^d_{1\times G}$ are forward probabilities for step $d$, and $O^{d+1}_{G\times G}$ are observation probabilities for step $d+1$, $f^{d+1} = f^d \times \mathcal{P} \times O^d$ computes forward probabilities for the next step in the sequence

in terms of task performance and visualization (for a comparable number of parameters).

### 3.2 Mixture model

The Sequential CG model described above can be combined with other generative models (e.g., language models) to get a mixture model. Here, we show how a unigram language model can be combined with Sequential CG. The rationale behind this is that since the Sequential CG is primarily designed to explain elements of documents that reflect sequential discourse structures, mixing with a context-agnostic distribution can allow it to focus specifically on elements that reflect sequential regularities. In experimental evaluation, we find that such a mixture model shows distinctly different behavior (see Section 4.1.1). Next, we briefly describe updates for this approach.

Let $\mu_z$ denote the multinomial distribution over features for the unigram model to be mixed with the CG. Let $\beta_z$ be the mixing proportion for the feature $z$, i.e. an occurrence of $z$ is presumed to come from the Sequential CG with probability $\beta_z$, and from the unigram distribution with probability $1 - \beta_z$. Further, let $\alpha_z^t$ be binary variable that denotes whether a particular instance of $z$ comes from the Sequential CG, or the unigram model. Then, Equation 2 changes to:

$$p(s|\, \mathcal{I}, \alpha) = \prod_{z,d} \Big( (h_{\mathbf{i}_d,z})^{c_z^d} \beta_z \Big)^{\alpha_z^t} \big( \mu_z^{c_z^d} (1{-}\beta_z) \big)^{1-\alpha_z^t}$$

Since we do not observe $\alpha_z^t$ (i.e., which distribution generated a particular feature in a particular document), they are additional latent variables in the model. Thus, we need to introduce a Bernoulli variational distribution $q_{\alpha_{zt}}$. Doing this modifies relevant parts (containing $q_{\alpha_{zt}}$) of Equation 5 to:

$$\begin{aligned} &\sum_t \sum_{\mathcal{I}} q_{\mathcal{I}}^t \Bigg( \sum_z q_{\alpha_{zt}} \log \big(\beta_z \prod_d h_{\mathbf{i}_d,z}^{c_z^d(t)}\big) \\ &+ (1 - q_{\alpha_{zt}}) \log \Big( (1 - \beta_z) \mu_z^{\sum_d c_z^d} \Big) \qquad (9) \\ &+ \sum_d \log \mathcal{P}_{\mathbf{i}_{d-1},\mathbf{i}_d} \Bigg) + \mathcal{H}_{q_{\alpha_{zt}}} \end{aligned}$$

This leads to the following additional updates for estimating $q_{\alpha_{zt}}$ (in the E-step)[4] and $\beta_z$ (in the M-step).

$$q_{\alpha_{zt}} = \frac{\exp\Big(\sum_{\mathbf{i}}^{\mathcal{I}} P(\mathbf{i}_d^t{=}\mathbf{i}) c_z^d(t) \log h_{\mathbf{i}_d,z}\Big)\beta_z}{\exp\Big(\sum_{\mathbf{i}}^{\mathcal{I}} P(\mathbf{i}_d^t{=}\mathbf{i}) c_z^d(t) \log h_{\mathbf{i}_d,z}\Big)\beta_z + \mu_z^{\sum_d c_z^d}(1{-}\beta_z)}$$

In the M-step, $\beta_z$ can be estimated simply as the fraction of times $z$ is generated from the Sequential CG component.

$$\beta_z = \frac{\sum_t q_{\alpha_{zt}}}{\sum_t \mathbb{I}_z}$$

## 4 Evaluation

In this section, we analyze the performance of our approach on text collections from several domains (including short stories, newswire text and biographies). We first qualitatively evaluate our generative method on a dataset of biographical extracts from Wikipedia, which visually illustrates biographical trajectories learned by the model, operationalizing our model concept from Figure 1 in real data (see Figure 2). Next, we evaluate our method on two standard tasks requiring document understanding: story cloze evaluation and sentence ordering. Since our method is completely unsupervised and is not tailored to specific tasks, competitive performance on these tasks would indicate that the method learns helpful regularities in text structure, useful for general-purpose language understanding.

### 4.1 Visualizing Wikipedia biographies

We now qualitatively explore models learned by our method on a dataset of biographies from Wikipedia.[5] For this, we use the data previously collected and processed by Bamman and Smith (2014). In all, the original dataset consists of extracts from biographies of about 240,000 individuals. For ease of training, we trained our method on a subset of the 50,0000 shortest documents from this set. The original paper uses the numerical order of dates mentioned in the biographies to extract biographical templates, but we do not use this information. Figure 2 visualizes a Sequential CG model learned on this dataset for on a grid of dimensions $E = 8 \times 8 \times 5$, and a histogram window $W$ of dimensions $3 \times 3$ . In general, we found that using larger grids leads to smoother transitions and learning more intricate patterns including hierarchies of trajectories, but here we show a model with a

[4] Since the optimal value for the concave problem $\sum_j x_j \log \frac{y_j}{x_j}$ s.t. $\sum_j x_j = 1$ occurs when $x_j^* \propto y_j$

[5] For all our experimental evaluation, we tokenize and lemmatize text using the Stanford CoreNLP pipeline, but retain entity-names and contiguous text-spans representing MWEs as single units

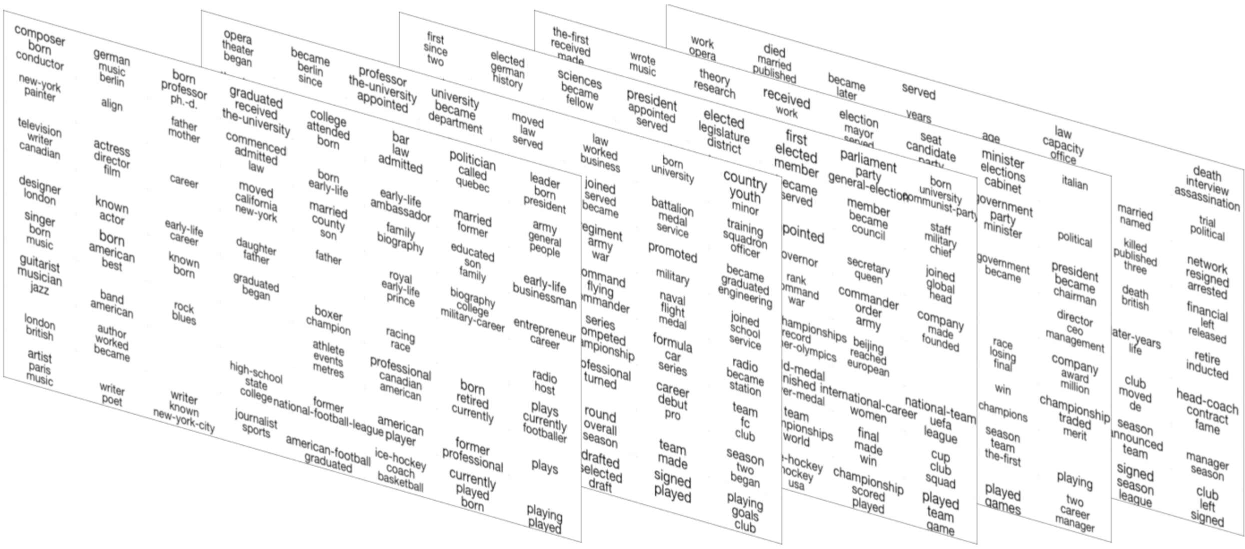


Figure 2: Visualization of a Sequential-CG model with grid size of $8 \times 8 \times 5$, trained on 50,000 documents from the Wikipedia biographies dataset. Cells in the grid show words with highest probabilities (empty cells may indicate that no word has a substantially higher probability than others).

smaller grid for ease of visualization. Here, the words in each cell in the grid denote the highest probability assignments in that cell. Larger fonts within a cell indicate higher probabilities.

We observe that the method successfully extracts various biographical trajectories, as well as capture a notion of similarity between them. To explain, we observe that the lower-right part of the learned grid largely models documents about sportpersons (with discernable regions focusing on sports like soccer, American football and ice-hockey). On the other hand, the left-half of the grid is dominated by biographies of people from the arts and humanities (inlcuding artists, writers, musicians, etc.). The top-center of the grid focuses on academicians and scientists, while the top-right represents biographies of political and military leaders. We note smooth transitions between different regions, which is precisely what we would expect from the use of the smoothing filter that incorporates parameter sharing across cells in the method. Further, as we go across the layers in the figure, we note the biographical trajectories learned by the model across the entire grid. For example, from the grid, the life trajectory of a football player can be visualized as being drafted, signing and playing for a team, and eventually becoming a head-coach or a hall-of-fame inductee.

### 4.1.1 Effects of mixing

The Sequential-CG method can be combined with other generative models in a mixture model, following the approach previously described in Section 3.2. A major reason to do this might be to allow the base model to handle general content, while allowing the Sequential-CG method to focus on modeling context-sensitive words only. Here, we empirically characterize the mixing behavior for different categories of words.

Figure 3 shows the mixing proportion of different words when the Sequential-CG model is combined with a unigram model. In the figure, the X-axis corresponds to words in the dataset with decreasing frequency of occurrence, whereas the Y-axis denotes the mixing proportions $\beta_z$ learned by the mixture model. We note that the mixture model learns to explain frequent as well as the long-tail of rare words using the simple unigram model (as seen from low mixing proportion of Sequential-CG method). These regimes correspond to (1) stopwords and very common nouns, and (2) rare words respectively. In turn, this allows the Sequential-CG component to preserve more probability mass to explain the intermediate content words. Thus, the Sequential-CG component only needs to model words that reflect useful statistical sequential patterns, without expending modeling effort on background content (common words) or noise (rare words). For the long tail of infrequent words, we observe that Sequential CG is much more likely to generate verbs and adjectives, rather than nouns. This is as we would expect, since verbs and adjectives often denote events and sentiments, which can

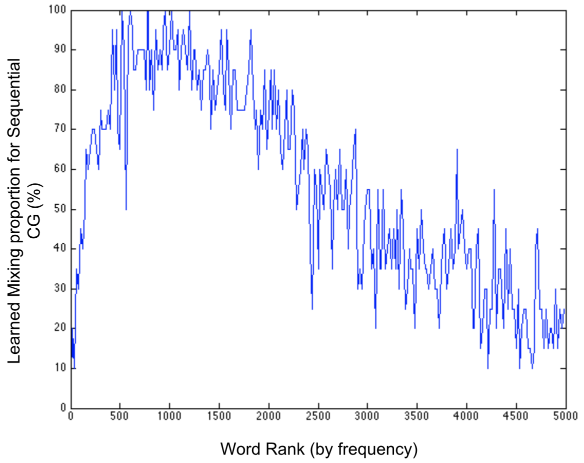


Figure 3: Learned mixing proportion ($\beta_z$) in the mixture model of Section 3.2 for words of different frequencies. $\beta_z$ denotes the probability of a word being generated from the Sequential CG model (rather than from the Unigram model). The Sequential CG learns to model content words (with intermediate ranks), and conserves modeling effort by avoiding modeling both very common words (that occur across contexts), as well as rare words.

| | Accuracy |
|---|---|
| **Our Method variants** | |
| Sequential CG + Unigram Mixture | 0.602 |
| Sequential CG + Brown clustering | 0.593 |
| Sequential CG + Sentiment | 0.581 |
| Sequential CG | 0.589 |
| Sequential CG (unnormalized) | 0.531 |
| DSSM | 0.585 |
| GenSim | 0.539 |
| Skip-thoughts | 0.552 |
| Narrative-Chain(Stories) | 0.494 |
| N-grams | 0.494 |

Table 1: Performance of our approach on story-cloze task from Mostafazadeh et al. (2016) compared with other unsupervised approaches (accuracy numbers as reported in Mostafazadeh et al. (2016)).

be important elements in discourse trajectories.

### 4.2 Story-cloze

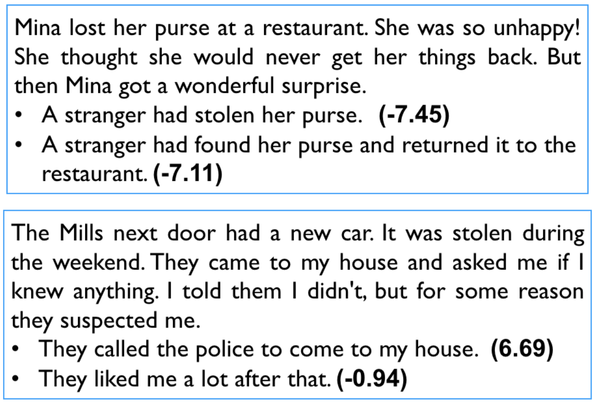
We next evaluate our method on the story-cloze task presented by Mostafazadeh et al. (2016), which tests common-sense understanding in context of children stories. The task consists of identifying the correct ending to a four-sentence long story (called *context* in the original paper) and two possible ending options. The dataset for the task consists of a collection of around 45K unlabeled 5-sentence long stories as well as 3742 5-sentence stories with two provided endings, with one labeled as the correct ending. For this task, we train our method on grids of dimension $15 \times 15 \times 6$ ($E$), and histogram windows $W$ of size $5 \times 5$ on the unlabeled collection of stories. At test time, for each story, we are provided two versions (a story-version $v$ consists of the provided context $c$, followed by a possible ending $e_1$, i.e. $v = [c, e]$ ). For prediction, we need to define a goodness score $S_v$ for a story-version.

In the simplest case, this score can simply be the log-likelihood $\log p_{SCG}(v)$ of the story-version, according to the Sequential-CG model. However, this is problematic since this is biased towards choosing shorter endings. To alleviate this, we define the goodness score by discounting the log-likelihood by the probability of the ending $e$ itself, under a simple unigram model.

$$S_v = \log p_{SCG}(c, e) - \log p_{uni}(e)$$

The predicted ending is the story-version with a higher score. Table 1 shows the performance of variants of our approach for the task. Our baselines include previous approaches for the same task: *DSSM* is a deep-learning based approach, which maps the context and ending to the same space, and is the best-performing method in Mostafazadeh et al. (2016). *GenSim* and *N-gram* return the ending that is more similar to the context based on *word2vec* embeddings (Mikolov et al., 2013) and n-grams, respectively. *Narrative-Chains* computes the probability of each alternative based on event-chains, following the approach of Chambers and Jurafsky (2008).

We note that our method improves on the previous best unsupervised methods for the task. This is quite surprising, since our Sequential-CG model in this case is trained on bag-of-lemma representations, and only needs sentence segmentation, tokenization and lemmatization for preprocessing. On the other hand, approaches such as *Narrative-Chains* require parsing and event-recognition, while approaches such as *GenSim* require learning word embeddings on large text corpora for training. Further, we note that predicting the ending without normalizing for the probability of the words in the ending results in significantly weaker performance, as expected. We train another



Mina lost her purse at a restaurant. She was so unhappy! She thought she would never get her things back. But then Mina got a wonderful surprise.

- A stranger had stolen her purse. **(-7.45)**
- A stranger had found her purse and returned it to the restaurant. **(-7.11)**

The Mills next door had a new car. It was stolen during the weekend. They came to my house and asked me if I knew anything. I told them I didn't, but for some reason they suspected me.

- They called the police to come to my house. **(6.69)**
- They liked me a lot after that. **(-0.94)**

Figure 4: Illustrative story-cloze examples where the model correctly identifies the appropriate ending (model score in parentheses).

| | *Accidents* | *Earthquakes* |
|---|---|---|
| Sequential CG | 0.813 | 0.946 |
| VLV-GM (2017) | 0.770 | 0.931 |
| HMM (2012) | 0.822 | 0.938 |
| HMM+Entity (2012) | 0.842 | 0.911 |
| HMM+Content (2012) | 0.742 | 0.953 |
| **Discriminative approaches** | | |
| DM (2017) | 0.930 | 0.992 |
| Recursive (2014) | 0.864 | 0.976 |
| Entity-Grid (2008) | 0.904 | 0.872 |
| Graph (2013) | 0.846 | 0.635 |

Table 2: Performance of our approach on sentence ordering dataset from Barzilay and Lapata (2008).

variant of Sequential-CG with the sentence-level sentiment annotation (from Stanford CoreNLP) also added as a feature. This does not improve performance, consistent with findings in Mostafazadeh et al. (2016). We also experiment with a variant where we perform Brown clustering (Brown et al., 1992) of words in the unlabeled stories ($K = 500$ clusters), and include cluster-annotations as features for training the method. Doing this explicitly incorporates lexical similarity into the model, leading to a small improvement in performance. Finally, a mixture model consisting of the Sequential-CG and a unigram language model leads to a further improvement in performance. The performance of our unsupervised approach on this task indicates that it can learn discourse structures that are helpful for general language understanding.

The story-cloze task has recently also been addressed as a shared task at EACL (Mostafazadeh et al., 2017) with a significantly expanded dataset, and achieving much higher performance. However, we note that the proposed best-performing approaches (Chaturvedi et al., 2017; Schwartz et al., 2017) for this task are all supervised, and hence not included here for comparison.

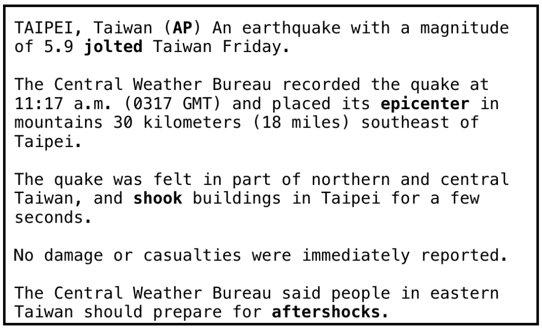
Figure 4 shows examples where the model correctly identifies the ending. These show a mix of behavior such as sentiment coherence (identifying dissonance between 'wonderful surprise' and 'stolen') and modeling causation (police being called after being suspected).

### 4.3 Sentence Ordering

We next evaluate our method on the sentence ordering task, which requires distinguishing an original document from a version consisting of permutations of sentences of the original (Barzilay and Lapata, 2008; Louis and Nenkova, 2012). For this, we use two datasets of documents and their permutations from Barzilay and Lapata (2008), which are used as standard evaluation for coherence prediction tasks. These consist of (i) reports of accidents from the National Transportation Safety Bureau (we refer to this data as *accidents*), and (ii) newswire reports about earthquake events from the Associated press (we refer to this as *earthquakes*). Each dataset consists of 100 training documents, and about 100 documents for testing. Also provided are about 20 generated permutations for each document (resulting in 1986 test pairs for *accidents*, and 1955 test pairs for earthquakes). Documents in *accidents* consist of between 6 and 19 sentences each, with a median of 11 sentences. Documents in *earthquakes* consist of between 4 and 30 sentences each, with a median of 10 sentences.

Since the datasets for these tasks only have a relatively small number of training documents (100 each), we use Sequential-CG with smaller grids ($3 \times 3 \times 15$), and don't train a mixture model (which needs to learn a parameter $\beta_z$ for each word in the vocabulary). Further, we train for a much smaller number of iterations to prevent overfitting ($K = 3$, chosen through cross-validation on the training set). During testing, since provided article pairs are simply permutations of each other and identical in content, we do not need to normalize as needed in Section 4.2. The score of a provided article is simply calculated as its log-likelihood. The article with higher likelihood is predicted to be the original.

Table 2 shows performance of the method compared with other approaches for coherence prediction. We note that Sequential-CG performs com-



TAIPEI, Taiwan (**AP**) An earthquake with a magnitude of 5.9 **jolted** Taiwan Friday.

The Central Weather Bureau recorded the quake at 11:17 a.m. (0317 GMT) and placed its **epicenter** in mountains 30 kilometers (18 miles) southeast of Taipei.

The quake was felt in part of northern and central Taiwan, and **shook** buildings in Taipei for a few seconds.

No damage or casualties were immediately reported.

The Central Weather Bureau said people in eastern Taiwan should prepare for **aftershocks.**

Figure 5: Example of newswire report about an earthquake event. Bold fonts represent words that align particularly well with the learned model at corresponding points in the narrative.

petitively with the state-of-the-art for generative approaches for the task, while needing no other annotation. In comparison, the HMM based approaches use significant annotation and syntactic features. Sequential-CG also outperforms several discriminative approaches for the task. In Figure 5 we illustrate the learned discourse trajectories in terms of the most salient features in each sentence. Words in bold are those identified by the model to be most context-appropriate at the corresponding point in the narrative. This is done by ranking words by the ratio between their probabilities ($\pi_{:,z}$) in the grid weighted by alignment locations of the document ($q^t_{\mathcal{I}}$), and unigram probabilities.

## 5 Conclusion

We have presented a simple model for extracting and visualizing latent discourse structure from unlabeled documents. The approach is coarse, and does not have explicit models for important elements such as entities and events in a discourse. However, the method outperforms some previous approaches on document understanding tasks, even while ignoring syntactic structure within sentences. The ability to visualize learning is a key component of our method, which can find significant applications in data mining and data-discovery in large text collections. More generally, similar approaches can explore a wider range of scenarios involving sequences of text. While here our focus was on learning discourse structures at the document level, similar methods can also be used at other scales, such as for syntactic or morphological analysis.

## References

Nicholas Asher. 1986. Belief in discourse representation theory. *Journal of Philosophical Logic*, 15(2):127–189.

David Bamman and Noah Smith. 2014. Unsupervised discovery of biographical structure from text. *Transactions of the Association for Computational Linguistics*, 2:363–376.

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

Peter F Brown, Peter V Desouza, Robert L Mercer, Vincent J Della Pietra, and Jenifer C Lai. 1992. Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–479.

Nathanael Chambers and Daniel Jurafsky. 2008. Unsupervised learning of narrative event chains. In *ACL*, pages 789–797. The Association for Computer Linguistics.

Snigdha Chaturvedi, Haoruo Peng, and Dan Roth. 2017. Story comprehension for predicting what happens next. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1603–1614.

Snigdha Chaturvedi, Shashank Srivastava, and Dan Roth. 2018. 'Where have I heard this story before?' : Identifying narrative similarity in movie remakes. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Thomas L Griffiths, Mark Steyvers, David M Blei, and Joshua B Tenenbaum. 2005. Integrating topics and syntax. In *Advances in neural information processing systems*, pages 537–544.

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

Nebojsa Jojic and Alessandro Perina. 2011. Multidimensional counting grids: Inferring word order from disordered bags of words. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, pages 547–556. AUAI Press.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent convolutional neural networks for discourse compositionality. *ACL 2013*, page 119.

Quoc V. Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pages 1188–1196.

Jiwei Li and Eduard Hovy. 2014. A model of coherence based on distributed sentence representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2039–2048.

Jiwei Li and Dan Jurafsky. 2017. Neural net models of open-domain discourse coherence. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 198–209.

Jiwei Li, Rumeng Li, and Eduard H Hovy. 2014. Recursive deep models for discourse parsing. In *EMNLP*, pages 2061–2069.

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

Annie Louis and Ani Nenkova. 2012. A coherence model based on syntactic patterns. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1157–1168. Association for Computational Linguistics.

William C Mann and Sandra A Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text-Interdisciplinary Journal for the Study of Discourse*, 8(3):243–281.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of NAACL-HLT*, pages 839–849.

Nasrin Mostafazadeh, Michael Roth, Annie Louis, Nathanael Chambers, and James Allen. 2017. LSDSem 2017 shared task: The story cloze test. In *Proceedings of the 2nd Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics*, pages 46–51, Valencia, Spain.

Karl Pichotta and Raymond J Mooney. 2015. Learning statistical scripts with LSTM recurrent neural networks. In *AAAI*.

Roger C Schank and Robert P Abelson. 1977. Scripts, plans, goals and understanding: an inquiry into human knowledge structures. *Erlbaum*.

Roy Schwartz, Maarten Sap, Ioannis Konstas, Leila Zilles, Yejin Choi, and Noah A Smith. 2017. The effect of different writing tasks on linguistic style: A case study of the roc story cloze task. *arXiv preprint arXiv:1702.01841*.

Di Wang, Nebojsa Jojic, Chris Brockett, and Eric Nyberg. 2017. Steering output style and topic in neural response generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2140–2150, Copenhagen, Denmark. Association for Computational Linguistics.

Xuerui Wang and Andrew McCallum. 2006. Topics over time: A non-markov continuous-time model of topical trends. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, pages 424–433, New York, NY, USA. ACM.