A Unified, Global and Local, Hierarchical Generative Document Ordering Model

Joe Austerweil*

May 4, 2007

1 Introduction

When reading a thesis, do you expect the first sentence to motivate the research or to involve a complex statistical process? When writing a news article, should the first sentence be an expert's commentary on the situation or a general introduction to the event? Although intuitively the answer to these two questions is obvious, most current language models do not distinguish between different orderings of sentences.

Recently, natural language researchers have developed local and global models of document coherence. Document coherence is "a property of well-written texts that makes them easier to read and understand than a sequence of randomly strung sentences." (Lapata and Barzilay 2005) The goal is to capture the flow of topic over the sentences of a document.

Local models capture coherence through the transition of either semantic or syntactic properties from sentence to sentence. To quantitatively estimate their success, researchers test the models on the document discrimination task. It evaluates how often the model can discriminate the document's sentence order from random permutations of the sentences. On the other hand, global models explicitly model the flow of topic through the document. The sentence ordering task empirically tests the success of these models. In this task, the model must impose an ordering on a "bag of sentences" and see how different this ordering is from the original document order. Although both models are successful within their own task, their performance on the other's task is dramatically lower.

Intuitively, unifying the two models is a natural solution. We propose a combined document ordering model that accounts for both local and global features. Not only will the combined model solve this problem, but it improves its performance on both tasks! In the next sections, we motivate

*The research underlying this paper was joint work involving the author, Micha Elsner, and Eugene Charniak.

natural language generation systems (2), local and global document modeling frameworks (3 and 4) and discuss a previous attempt to unify the features (5) Unfortunately, the previous model is not generative and learns local and global features independently. This motivates our own unified jointly learned generative model (6). Finally, we test the models on the two tasks (7).

2 Natural Language Generation Systems

Over the last decade, the natural language generation literature has taken a dramatic shift from domain specific systems to domain general modeling. Before this shift, generation systems relied on expert knowledge representations in order to structure and create text easily understandable by human readers. For example, Kathleen McKeown's `TEXT` generation system for army devices is based on document structure schemas and a hardcoded relational system between different types of vehicle and missiles at different levels of specificity (McKeown 1985). As it is neither practical nor scalable to create a knowledge representation for every possible discourse, researchers have moved to techniques for learning the structure of domains automatically from corpora. These corpus based techniques borrow sophisticated machine learning tools and apply them to natural language generation.

Generally, there are three major components of a natural language generation system: document planning, microplanning, and realization Document planning determines the content and outline of text to be generated. Microplanning and realization focus on converting the structure created during planning into an actual readable document. Although intuitively these components are not independent, in order to simplify the problem we (and most researchers) treat these components as independent modules. As human readers read documents to find content in particular places of a text, the document planning step is extremely important. To quote Sripada et. al. (2001) :

> Content determination is extremely important to end users; in most applications users probably prefer a text which poorly expresses appropriate content to a text which nicely expresses inappropriate content. From a theoretical perspective content determination should probably be based on deep reasoning about the system's communicative goal, the user's intentions, and the current context, but this requires an enormous amount of knowledge and reasoning, and is difficult to do robustly in real applications.

Accordingly, we focus on the document planning problem.

There are two main components to a document planning system: content determination and document structuring. Content determination selects what information should be presented in the document. Until recently,

content determination was directly encoded by a domain-expert and not learned from statistical analysis of corpora. It was thought impossible "to specify general rules for content determination" (Reiter and Dale 2000). Although this is still true in the sense that new content selection systems must be retrained for each domain-specific corpus, these systems do not need to be respecified by a domain expert. By defining content determination as a classification task (each potential item is either present or not in the generated text), we can use previously developed machine learning techniques to solve the content determination problem. This is the approach used in the last few years by Duboue and McKeown (2003) and Barzilay and Lapata (2005) . Now that we have the content to expound in the text, how should we order this content without domain-specific rules?

Sentence ordering models are not only used as a module in natural language generation systems. Miltsakaki and Kukich (2004) use rough heuristics to predict students essay grades for ETS standardized tests. In principle, a true model of document coherence (with a sentence ordering model as a component) could provide a more principled method to classify student essays. Another major use for sentence ordering models is in multi-document summarization. Kathleen McKeown's natural language processing group at Columbia combine multiple different news sources into one coherent news article with aspects from the different sources (Evans et. al. 2004). In a news article domain, sentence ordering is intimately connected to summarization as news article readers have strong constraints on the order of presentation of information (generally, most informative to least). Again this is the technique used by Barzilay and Lee (2004)'s hidden Markov model of a document.

There are two major computational approaches to order documents without specifying domain-specific information. First, local coherence models score coherence as the product of a function of groups of adjacent sentences over all groups of sentences in the document. Second, global coherence models track the change of topic throughout the document and judge its coherence based on the flow of sentence topic. Regardless of the type of model, they both define the optimal ordering of sentences to be the ordering of sentences with the largest coherence score.

3 Local Coherence Models

Local coherence models focus on the relation between a small number of the sentences adjacent to each sentence in a document (D). More formally,

$$P(D) = P(S_1, \ldots, S_n) \equiv \prod_{i=1}^{n} P_{\text{coherence}}(S_i | S_{i-1}, \ldots, S_{i-h})$$

where $P_{\text{coherence}}(S_i|S_{i-1}, \ldots, S_{i-h})$ is a coherence function whose argument is the i-th sentence given the last h sentences (interpreted as a probability).[1] Generally, there are two main types of local coherence models, semantic and syntactic One method for defining the local coherence between adjacent sentences is semantic similarity. In these models, we assume (most likely incorrectly) that the meaning of adjacent sentences are more similar than the meaning of sentences later on in the text. Another type of local coherence model, syntactic models, are inspired by Centering Theory, which predicts that the realized entities and pronouns are subject to strict syntactic constraints based on the discourse entities of the last few sentences. Grosz et al. (1995) have argued that the coherence of an entire text depends mostly on the local coherence between adjacent sentences. Although in practice local coherence is important, we show that a fully coherent document depends on the interconnection between local coherence and global structure. In the next section, we discuss two local semantic similarity models, word overlap and latent semantic analysis.

3.1 Word Overlap and Latent Semantic Analysis

We model the coherence of a text in local semantic models to be:

$$\text{coherence}(D) = \frac{1}{n-h} \prod_{i=1}^{n-h} sim(S_i, \ldots, S_{i-h})$$

Given the general local coherence framework above, we define a simple coherence metric by representing a sentence as a vector of words and computing the word overlap between each adjacent sentence (Markov horizon $h = 1$). Word overlap provides some useful insights. For example, a word introduced in a sentence is more likely to be further explained in the next sentence than much later in the document. Unfortunately, it provides no method of determining the order between the pair of sentences (it is symmetric), which is a prevalent problem for all local coherence models. Additionally, it is unable to provide insight about semantically related and coreferent words. For example, if a car is mentioned in a previous sentence, it is more likely that the word "wheel" will appear than if falafel was last mentioned. This insight led Foltz et. al. (1998) to create the latent semantic analysis (LSA) framework for identifying semantic relations between words.

Latent semantic analysis transforms the document matrix formed for word overlap into a lower dimensional space. The hope is that the lower dimensional space represents different groups of related words, which are actually interchangeable We define our local coherence metric as a distance metric similar to word overlap, but instead of using the words as our vectors, we use vectors corresponding to the words mapped into the lower dimensional space. This way LSA could theoretically distinguish between the two

[1] This assumes a generative framework. If our model is conditional, we can include later sentences in our context (e.g. $P_{\text{coherence}}(S_i|S_{i-1}, \ldots, S_{i-h}, S_{i+1}, \ldots, S_{i+h})$).

1: The commercial pilot, sole occupant of the airplane, was not injured .
2: The airplane was owned and operated by a private owner.
3: Visual meteorological conditions prevailed for the personal cross country flight for which a VFR flight plan was filed.
4: The flight originated at Nuevo Laredo, Mexico, at approximately 1300.

entity	1	2	3	4
PLAN	-	-	O	-
AIRPLANE	X	O	-	-
CONDITION	-	-	S	-
FLIGHT	-	-	X	S
PILOT	O	-	-	-
PROPER	-	-	-	X
OWNER	-	X	-	-
OCCUPANT	X	-	-	-

Figure 1: A sample four sentences and its corresponding entity grid representation

sentence pairs mentioned above. Unfortunately, in practice, the clusters of words LSA finds is quite noisy. Additionally, the interchangeability assumption is troubling. In the previous example, car and wheel are not fully interchangeable Wheels are inferable from the mention of a car; however, a car is not inferable from the mention of wheels. This asymmetry is not captured by LSA. Although there are more troubling aspects to modeling document coherence with LSA (most notably its inability to track entities through the document), we move on to the entity grid model.

3.2 Naive Generative Entity Grid Model

The generative formulation of the entity grid model presented by Lapata and Barzilay (2005) attempts to model two concepts formalized in Centering Theory. Centering Theory explores how the focus on different entities changes over the sentences of a document and how these entities are pronomialized. First, entities occur in particular syntactic patterns over the sentences of the document. More specifically, there are soft constraints on the change of focus from sentence to sentence. Second, roughly there are two main types of entities in a discourse: salient and non-salient. The entities that are salient are more likely to appear in important syntactic roles and in more sentences throughout the document. For example, in a document about Microsoft, we would expect Bill Gates and Microsoft to be salient and falafel to be non-salient. With these two modeling assumptions, we form a simple yet powerful generative model of local document coherence.

In their generative formulation of the entity grid model, Lapata and Barzilay represents a sentence (S_i in a document (D) as a vector of the

syntactic roles ($\vec{r_i}$ are the roles that all entities take in sentence i) of all the entities ($\vec{e}$) of the document in this particular sentence. As shown in figure 1[2], we represent the simple 4 sentence document as a matrix ($E = R$) of the roles of each entity ($e_{i,j} = r_{i,j}$) over all the sentences of the document. There are four different roles an entity ($r_{i,j} \in R = \{S, O, X, -\}$)[3] can have throughout the document: subject (S), object (O), occurring but neither the subject nor object(X), or not occurring in the sentence($-$). Now we formally define the coherence of a n sentence document with m entities to be:

$$P_{\text{coherence}}(D) = P(S_1, \ldots, S_n) = P(e_{1,1} = r_{1,1}, \ldots, e_{1,m} = r_{1,m}, \ldots, e_{n,1} = r_{n,1}, \ldots, e_{n,m} = r_{n,m}) = P(E = R)$$

In order to simplify the derivation to a form we can actually compute, Lapata and Barzilay make two assumptions. First, all entities are generated independently of the other entities (column independence). Second, the role an entity takes in a sentence is independent of the other roles it takes given the roles it has taken in the last h sentences (row Markov independence). This allows us to simplify our calculation of the probability of a document to a double product over each role each entity takes conditioned over the roles within its Markov horizon:

$$P(S_1, \ldots, S_n) =$$
$$P(e_{1,1} = r_{1,1}, \ldots, e_{1,m} = r_{1,m}, \ldots, e_{n,1} = r_{n,1}, \ldots, e_{n,m} = r_{n,m}) =$$
$$\prod_{j=1}^{m} \prod_{i=1}^{n} P(r_{i,j} | r_{1,j}, \ldots, r_{i-1,j}) \approx \prod_{j=1}^{m} \prod_{i=1}^{n} P(r_{i,j} | r_{i-h,j}, \ldots, r_{i-1,j})$$

Unfortunately, the simple maximum likelihood solution fo̊these estimators, $\hat{P}(r_{i,j} | r_{i-h,j}, \ldots, r_{i-1,j}) = \frac{C(r_{i-h,j}, \ldots, r_{i,j})}{C(r_{i-h,j}, \ldots, r_{i-1,j})}$ (where $C(r_{i-h,j}, \ldots, r_{i,j})$ is the number of times any the sequence of roles $(r_{i-h,j}, \ldots, r_{i,j})$ appears in the training corpus), is not very informative. Given the above discussion, the most obvious problem is that we still do not distinguish between salient and non-salient entities. Since salient entities should be frequently mentioned in important syntactic roles, their distribution over roles should be widely different than non-salient entities for a few reasons. First, most entities do not occur in most sentences and thus an overwhelming amount of probability mass is given to the $-$ role. This matches our intuition and also Zipf's law, which dictates a majority of the words in a corpus occur only once. Second, Centering Theory predicts that salient entities reoccur according to different rules than non-salient entities. With this theoretical impetus, we add the number of times an entity occurs in a document to the information

[2]Taken from Elsner et. al. (2007)

[3]Like Elsner et. al. (2007) we determine the roles heuristically using syntax trees produced by the parser of Charniak and Johnson(2005) .

we condition on.[4] When we add the number of occurrences of an entity to our conditioning information, we arrive at another related problem. Even salient entities do not occur in the majority of sentences of a document. This leads to our MLE estimate of $\forall H : \hat{P}(r_{i,j} \in S, O, X|H) < 0.5$, where H is any possible conditioning information or equivalently $\forall H : \hat{P}(r_{i,j} = -|H) > 0.5$. Taking a generative lens, the most likely generated document from our model is one without any entities! Not only is this not intuitive (legitimate documents with no entities?), it leads to poor performance on the binary classification task. In order to properly address this problem, we need to move to a more complicated estimator and relax our independence assumptions.

The first step to improving the generative entity grid is to include psuedo-count smoothing over the different types. This changes our MLE estimator of the transition probabilities to

> $\hat{P}(r_{i,j}|r_{i-h,j}, \ldots, r_{i-1,j}, k; C_k) = \frac{C(r_{i-h,j},\ldots,r_{i,j})+C_k}{C(r_{i-h,j},\ldots,r_{i-1,j})+|R|C_k}$, where k is the number of occurrences of entity $e_{i,j}$, and C_k is its associated smoothing constant.

Although this amounts to a substantial improvement, the improvement is grossly different for the optimal values of the smoothing constants and the ones obtained by estimation from a development corpus. After trying a variety of techniques for deriving principled estimates of the optimal values for C_k, we discovered that none of these techniques yields satisfactory performance during testing.[5] As the value of smoothing constant increases, the probability of the training corpus decreases. However, it improves the performance of the model during testing. By increasing the value of our smoothing constant, we are increasing the uniformity of our probability distribution, which "steals" probability mass from the $-$ to $-$ transition, the most likely transition. By shifting probability mass to transitions with entity occurrences, we change our decision boundary during evaluation. Rather then focusing on when entities do no occur, we increase the importance of entity occurrences (which are more informative). From these results, we infer a need to improve our local coherence model beyond simple smoothing.

[4]Unfortunately, this adulterates our model. It is now a degenerate generative model. Our two attempts at purifying the salience portion of our model failed empirically. Both conditioning on the amount of times the entity has occurred "so far" and a simple boolean flag (salient means occurs more than twice, although this too has its theoretical issues) failed. So we are stuck with conditioning on the total number of occurrences of each entity in the document.

[5]No technique yielded satisfying performance while testing on the development corpus. We did not waste the test corpus on our inferior model, but rather used ten-fold cross-validation with the training corpus and then for each training corpus in the current validation broke data into a training set and a held-out set. We used this held-out set to derive the values of the smoothing constants by maximizing the probability of the held-out data under the model varying the value of the constants.

3.3 Relaxed Generative Entity Grid Model

The results from the previous section all point to a major flaw of the naive generative entity grid model: the most likely transition of the model is $-$ to $-$. Although this yields satisfying results when the generative model is "plugged into" a conditional model (like the SVM as in Barzilay and Lapata(2005)), it is no longer generative. This makes it difficult to achieve our ultimate goal of incorporating the local model into a global coherence model.[6] Thus, we propose a relaxed generative entity grid model. It generates the syntactic roles of each sentence and decides whether or not each entity fills the role.

The relaxed entity grid model has its benefits and disadvantages. On one hand, it is a degenerate and inconsistent estimator (see Elsner, Austerweil, and Charniak (2007)), two unwanted properties. Inconsistency imples that given an infinite amount of training data, our estimator does not converge to the proper distribution. On the other hand, it yields improved performance and a model that can be principally incorporated into a global model. We show in general, local and global models have disparate problems and their combined model improves performance on both of our tasks.

To "steal" probability mass from the blank transition to more informative entity transitions, we relax the naive model's row independence assumption (entity independence). First, we separate our entities into two sets: those that previously occurred, K_i (known entities at sentence i), and those that have not, N_i (new entities). Since our local model has little information to base which syntactic role a new entity should fill, we ignore entities in N_i for now (but not in the combined model). Thus, our new model only discerns between the different syntactic roles old entities fill. The generative process modeling a sentence can be thought of as first generating the syntactic roles of the sentence. Next we pick the syntactic role, r, each known entity fills given we are filling r, given the number of times it occurs in the document and the Markov horizon of all known entities.

Although this eliminates the problem of the most likely document having no entities, it does mean multiple entities can fill the same syntactic role and the same entity can fill multiple syntactic roles (degeneracy). However, this is an improvement over an entity-less most likely document. Additionally we assume the distribution of syntactic roles over sentences (how many subjects, objects, etc. in sentence i) does not change from sentence to sentence. Formally, the probability of a sentence under our new model is: $P(S_i|S_{i-1},\ldots,S_1) \approx$

$$P(r_{i,1},\ldots,r_{i,|R|}) \prod_{e_j \in K_i} \prod_{r \in R_i} P(e_j = r | r, \vec{E}_{i-1},\ldots,\vec{E}_{i-h}, k_e), \text{ where}$$

[6]The full benefits of combining generative models vs. conditional models is beyond the scope of this thesis. However, it should be clear how to easily incorporate generative models together.

R_i are the syntactic roles to be filled for sentence i, $\vec{E}_{i-1}$ is the vector of roles filled by entities in sentence $i-1$, and k_e is the number of occurrences for entity e_j.

Here, we face a similar situation in the natural language processing literature; our context is too specific and thus, our distribution is sparse. Therefore, we form a new estimator by throwing out the histories of the known entities not in question and normalizing via brute force, which is unfortunately inconsistent (Elsner et. al. 2007). This results in the following equation as the basis of our local model:

$$P(e_j = r|r, \vec{E}_{i-1}, \ldots, \vec{E}_{i-h}, k_e) \approx \frac{P(e_j=r|r,e_{i-1,j},\ldots,e_{i-h,j},k_e)}{\sum_{e_l \in K_i} P(e_l=r|r,e_{i-1,l},\ldots,e_{i-h,l},k_{e_l})}$$

We show in the results section that this model is intuitively more satisfying and leads to better performance in both ordering tasks. Although it does improve performance, it still is not completely satisfying. It remains an important door for further refinement and exploration.

Before moving to a global coherence model, we discuss the types of discriminations that local coherence models succeed and fail to make. On one hand, local coherence models are great at determining random permutations of the sentences of large documents from the original ordering. If we abstract away the exact syntactic roles to either occurrence or not, we observe the general behavior shown in figure 2. We name one bursty column a "Markov cluster." Random orderings frequently break up Markov clusters of sentences, which the model uses to distinguish coherent from incoherent documents with. This allows our local models to perform very successfully on the binary classification task.

On the other hand, there are many permutations, which a human reader can distinguish between, but our model can not (or has little information to). Consider the ordering of document that is the complete reverse ordering (so the first sentence is the last sentence, second sentence is the second to last, and so on). According to local coherence models, the two documents are nearly equivalent because we still have the same Markov clusters of sentences (and the effects of asymmetry are minimal). Another difficult transformation would be random permutations of paragraphs (but maintain the ordering of sentences within paragraphs). To impose an order on a bag of sentences, we need stronger constraints for how the sentences within and the clusters themselves should be arranged. We need a way to judge coherence that relies on something besides the Markov clusters. One method is to track the global change of sentence topic throughout the document. This is the underlying theme behind Barzilay and Lee's (2004) HMM global coherence model that we discuss next .

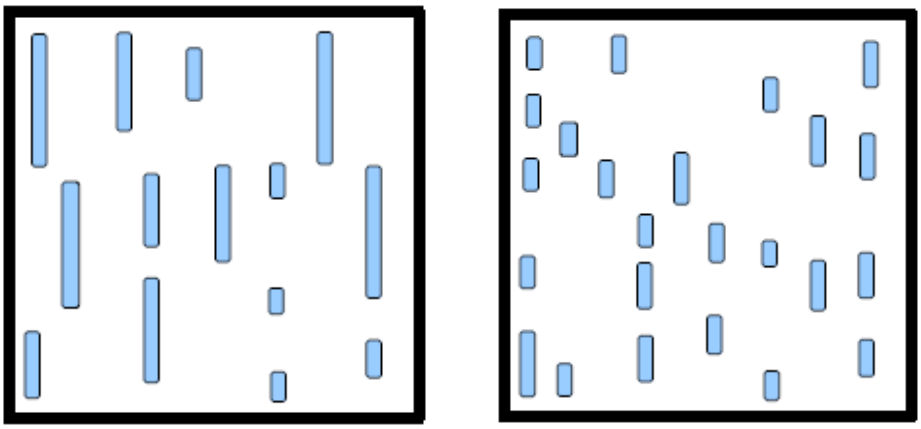

Figure 2: On the left is the abstracted matrix for a coherent document. A bar represents the occurrence of the entity in the current sentence. Note the bursty chains of co-occuring entities. On the right is the abstracted matrix for a randomly permuted document. The bursty chains of co-occuring entities are violated.

4 Global Coherence Model

Over the last two decades, the seminal natural language generation systems focused on global text generation schemas. The schemas, like parsing grammars, form a hierarchical structure over the document, which defines the local and global properties of a document in a recursive manner. Although these systems are successful in extremely constrained domains , automatically learning these schemas from corpora proves difficult. This is an exciting area of future research. The building blocks towards automatically learning rich global topic grammars are an interesting area for later pursuit.

4.1 Topic-based Hidden Markov Model

One exciting recent development in automatic global coherence modeling is Barzilay and Lee's (2004) topic based model. Unlike other recent developments in topic based language modeling (see Hofmann 1999 and more recent Griffiths et al 2005), Barzilay and Lee do not treat topic as an auxiliary variable. Instead, they assume it is the underlying cause for the document. They treat the words of each sentence of a document as the emissions from a hidden Markov model. The hidden Markov model transitions between hidden semantic states and each state is a language model for the current sentence. Each hidden state is the topic of each sentence. Since many HMM learning and prediction algorithms are well studied and developed, the new modeling method comes with sophisticated techniques for parameter esti-

mation from corpora.

Although similar to a straight-forward application of an old modeling technique applied to a new domain, they adapt the technique and preprocess the corpus. Instead of starting their topics from a blank slate, they preprocess their topics via complete link clustering (where clusters are sets of single sentences). To cluster, they form simple bigram features, create k clusters, and merge together all clusters with less than T sentences (where k and T are parameters to the algorithm set to 100 and 4 respectively). They assume the hidden states generated the clusters formed by complete-link clustering. Naturally, they formulate their emission probability functions to be simple bigram language models with pseudocount smoothing (although in general, any language model could be applied). Additionally, they define the state transition probabilities to be the ratio of the number of documents we observe a sentence from cluster j right after a sentence from cluster i over the number of documents with at least one sentence from cluster i. Again, pseudocount smoothing is applied to the transition probabilities to improve estimates from sparse data. Given these formalizations, we learn parameter values using EM reclustering at each step. However, in our own implementation, we found it unnecessary to recluster at each step if a more principled basis is used (yet still domain-general) for forming the initial clusters.

This model opened the door for domain-general global coherence modeling. For example, it learns from the airplane training corpus that news articles frequently begin with a sentence describing the date, time, and nature of the accident (it appropriately groups these sentences as a cluster). Although this technique is an important first step towards a domain-general global coherence model, there are two major flaws to address. We assume the underlying structure generating documents is a hidden Markov model; however, this contradicts our intuition of a hierarchical document structure. Although for small documents this seems reasonable, it does not scale to longer documents (sentence topics are actually conditionally independent given a structured outline) and its solution could lead to improved results. It would be intriguing to apply syntactic grammar induction techniques to the document modeling; however, our initial formulations proved this to be an elusive and difficult project (as most semantic modeling projects are). Perhaps we can assume an underlying hidden cause for the hidden states.

Important to our combined model, we ignore the local constraints between entities. This makes it difficult for the model to develop useful sentence topic states that do not contain too specific information. For example, consider an article describing a battle between Captain Jack Sparrow and Commodore Norrington in a theoretical Pirate Conquests corpus. Should the first topic state for the news article include that it is describing a fight between a pirate and naval commander or that the entities are Sparrow and

Norrington in particular? [7] Although this may seem contrived, it is a serious problem because the global model is forced to choose between being overtrained (and sparsity issues) or ignoring how entities change overthe document. Qualitatively, this leads to the topic model having rough ideas over what part of the document sentences should be, but trouble deciding how to group the sentences together (assuming the model choose to ignore local entity transitions). Quantitatively, it is reflected in its relatively poor performance on the binary classification task (see table 2 in section 7.2).

5 Motivation for Combined Model

The motivation for a combined model is the hope that we can create one document coherence model, which performs well on both tasks with only one training session. Our discussion of local coherence modeling showed that local coherence models are good at forming groups of sentences; however, they cannot order the sentences within each group and the groups themselves. Although this is a problem for imposing a coherent sentence order, it can easily classify between the original and random orderings of a document because random orderings do not contain Markov clusters. On the other hand, the HMM model we just discussed has difficulty tracking specific entities through a document. However, it imposes loose, reasonably coherent constraints on sentence orderings. If these two models have complementary flaws and benefits, can we combine them in a single unified model that has all the benefits without the flaws and performs better than each alone on both tasks? This is the basic motivation behind our proposed model and Soricut and Marcu (2006)'s mixture model approach.

5.1 A Mixture Model Approach

A mixture model is a convenient method for integrating multiple disparate models into one model. Each of the models comprises one feature or component of the mixture model. The overall decision of the mixture model is a weighted sum of its different features. Intuitively, each feature's weight corresponds to how diagnostic the feature is. Additionally, it assumes that the coherence of a document is determined by an independent (given the feature weights) weighted average of the coherence ratings according to each model. Since the feature weights are constants, the mixture model is driven either by local or global dependencies and cannot learn the interdependence between local and global constraints. This is one major problem with a mixture model approach to unified document coherence models.

Soricut and Marcu (2006)'s sentence ordering model combines local and global models by a log linear mixture of four different models:

[7] Jack Sparrow and Norrington are property of the Walt Disney Corp.

$P(D) = \frac{\exp[\sum_{m=1}^{4} \lambda_m f_m(D)]}{\sum_{D'} \exp[\sum_{m=1}^{4} \lambda_m f_m(D')]}$, where D is a potential ordering of the document, λ_m is the mixture weight of model m, and $f_m(D')$ is the coherence of ordering D' by model m.

Although an explicit calculation of $P(D)$ demands computing an expensive normalization factor, the normalization constant does not need to be calculated for our two tasks. The binary classification task is the ratio of two document probabilities. The constant appears in both the numerator and denominator of the ratio and so, it can be ignored. Similarly, imposing a sentence ordering is the argmax of document probability over all possible sentence orders. The normalization factor is constant over orderings and so, it also can be ignored. The four different models used in their mixture are: the naive entity-grid model (although slightly adapted and non-generative), the HM content model with bi-gram language models, and two local word-co-occurrence models based on the classic machine-translation IBM models. Given uniform weighting between the four models, the mixture performed worse than its best component alone; however, when the mixture weights are optimized, the mixture outperforms any of its components on the sentence ordering task. (Soricut and Marcu 2006)

By unifying global and local coherence models, we form one model, which performs better than any of the individual models on a particular task. Using the mixture modeling framework, we form a unified model that successfully captures document coherence. Unfortunately, there are two major problems with this technique: independence and modularity. The independence problem was alluded to earlier. Given the mixture weights, all of the component models are independent from each other. The mixture model must decide to always include both models (in a constant weighted average) or choose one of the models to always base its decision on. This independence is a problem because the global and local constraints are dependent on each other and thus cannot be decomposed into two independent parts. One reason (besides appealing to intuition) for a jointly dependent model is it allows some words, such as salient entities, to effect both the local and global constaints, where as other words, such as adverbs, to effect only local constaints. Another major problem of the mixture model is modularity. We motivated sentence ordering document coherence models as a module in a multi-document summarization or natural language generation system. Generative models provide a principled manner for integrating models together (simply add the link(s) to the graphical model and derive new estimators). It is unclear how to directly integrate a mixture model system into a multi-document summarization or coreference system. Although the sentence ordering task is inherently interesting, it is important to principally use them in other natural language processing tasks. Additionally, we can incorporate other generative models and look for areas to improve our model in principled ways. (see Griffiths et al. 2005 for a similar discussion)

6 Generative Jointly Learned Unified Global and Local Model

After the discussion of the mixture model coherence approach, the need to capture interdependencies between the local and global models is clear. One principled method that captures component interdependencies is the generative framework. Additionally, the framework lends itself to modularity - another of our goals. Before we delve into our sentence ordering model, we discuss Dirichlet and Pitman-Yor processes and a Bayesian interpretation of the HMM. Afterwards, we have the tools to form our unified global and local generative model of document coherence.

6.1 Chinese Restaurant, Dirichlet and Pitman-Yor Processes

Over the last decade, models containing the Chinese Restaurant, Dirichlet, and Pitman-Yor Processes have proliferated the statistical machine learning community. These statistical processes have been applied to: enhance pre-existing models (e.g. the infinite Hidden Markov Model of Beal et al. 2002), create new models with improved performance (e.g. latent Dirichlet allocation of Blei et al. 2003), and understand the principle behind extremely successful, but poorly understood techniques, (e.g. Kneser-Ney smoothing as an approximation to a generative language model by Goldwater et al 2006 and Teh 2006). One main benefit for modeling with these processes is generativity. Thus, we can form models that are intuitive, complex, make principle assumptions explicit, capture the interdependencies of codependent features, and modular. In her thesis, Goldwater (2006) explores the importance of making the correct prior assumptions. Interestingly, one assumption underlying all of these processes is *a priori* we expect our distribution be power-law. One example of the benefit of proper prior assumptions is Goldwater et al. (2006) and Teh (2006)'s explanation of Kneser-Ney smoothing with a power-law prior.

Throughout the twentieth century, researchers have shown many social phenomena follow a power-law distribution. For example, in 1896, Pareto described the distribution of income over people as a distribution where a few people have most of the money and the vast majority have relatively small amounts money.[8] This "bursty" behavior (mostly small, but occasionally extremely large) is characteristic of a power-law distribution. Formally, a power law distribution is any that fits the following form (Mitzenmacher 2003):

$P(X \geq x) \sim cx^{-\alpha}$, where $\alpha > 0$ and $c > 0$ is a normalizing

[8]Researchers frequently assume a Gaussian underlying distribution. One phenomenon reasonably modeled by a Gaussian distribution is the height of people in the world. Unfortunately, not all phenomena are Gaussian.

constant.

If $0 \leq \alpha \leq 2$, the power-law distribution has infinite variance, which allows the distribution to have interesting bursty behavior typical of social phenomena. Additionally, it violates the central assumptions behind the basic central limit theorem. The normalized sum of random variables with infinite variance does not converge to a Gaussian distribution. In our income example, it does not make sense to model the underlying distribution as Gaussian because most people do not have an income near the mean of the distribution. More importantly for natural language processing, Zipf's law [9] states that the frequency of words in a language follow a power-law distribution. Thus, the majority of words occur only a few times in the document; however, a few words such as "the" occur thousands of times more. It has been empirically shown that this property holds across different languages, where each language has a different α value. (Manning and Schütze 1999, Zipf 1935).

One way to generate a power-law distribution is preferential attachment. Under preferential attachment, the probability of an event occurring is proportional to the number of times it occurred previously. Consider a view where a table represents an event and a customer seated at a table represents one occurrence of the event represented by the table. *A priori*, we do not know the number of tables, but we assume a parameter, θ, controls the probability of creating a new table. Additionally, we assume preferential attachment, which dictates that the probability we sit at a table is proportional to the number of customers at the table. If we have seated the first $z_{-i} = \{z_1, \ldots, z_{i-1}\}$ customers, the probability that the next customer, z_i, sits at table k is:

$$P(z_i = k | z_{-i}; \theta) = \begin{cases} \frac{n_k^{(z_{-i})}}{i-1+\theta} & k <= Z \\ \frac{\theta}{i-1+\theta} & k = Z+1 \end{cases}$$

where $n_k^{(z_{-i})}$ is the number of times k occurs in z_{-i}, Z is the current number of tables, and θ is a parameter.

This is one view of the Chinese Restaurant Process (CRP).

Changing θ controls how likely we are to create a new table at each time step.[10] As $\theta \to \infty$, we create a new table each customer and as $\theta \to 0$ all customers sit at one table. Another interpretation is that θ provides a number of pseudo-customers to each of the countably infinite tables (similar to add pseudo-count smoothing). This interpretation is particularly insightful when we add labels to the tables. We label a table when it is created by

[9] There is debate about whether or not Zipf actually made this discovery. See Mitzenmacher (2003) for an interesting historical discussion.

[10] Although the probability of the first customer sitting at a new table is 1 regardless of the value of θ

generating its label from some parent distribution. A customer can sit at a table if the table's label matches some information of the customer. For example, consider a hierarchical model where the base distribution ($G_0(\cdot)$) is uniform over all unigram words. We label a new table in the child (CRP) distribution by drawing a sample from the parent distribution. Our customers in the child distribution are unigram words and can only sit at a table whose label matches its word (if no such table exists, we create a new one for it). This gives us the following distribution for the next word:

$P(z_i = w|z_1, \ldots, z_{i-1}; \theta, G_0(\cdot)) = \sum_{k=1}^{Z} I(t_k = w)\frac{n_k^{z_{-i}}}{i-1+\theta} + \frac{\theta}{i-1+\theta}G_0(w)$, where $I(\cdot)$ is the indicator function and t_k is the label of table k.[11]

We can use any context as our table labels and any base distribution, which provides us with a powerful modeling framework. Teh (2006) describes a language model where the parent distribution of the n-gram restaurant is the (n-1)-gram restaurant. This defines a hierarchical recursive structure with a base case uniform distribution as the parent of the uni-gram restaurant (shown in the equation above). Intuitively, as θ at each level increases, we approach the base distribution, and as it decreases, we approach the MLE of each word. Thus, θ controls the smoothing between the parent and child distributions and optimizing θ provides an optimal smoothed distribution. It can be shown that this process produces a power-law distribution where θ controls the value of the exponent of the power-law ($\alpha = \frac{1}{1-\theta}$). Thus, we have defined the probability of a seating arrangement to be:

$P(z) = \prod_{i=1}^{n} P(z_i|z_1, \ldots, z_{i-1})$
$= (\prod_{i=1}^{n-1} \frac{1}{i+\theta})(\theta^{Z-1})(\prod_{k=1}^{Z} (n_k^{(z)} - 1)!)$
$= \frac{\Gamma(1+\theta)}{\Gamma(n+\theta)}(\theta^{Z-1})(\prod_{k=1}^{Z} (n_k^{(z)} - 1)!)$
(note: This distribution is *exchangeable*, which allows us to use Gibbs sampling to estimate the parameters later. This is not equivalent to independent and identically distributed.)

The Dirichlet Process (DP) is one form of the CRP when θ is restricted to positive non-zero values. (Goldwater 2006) The Pitman-Yor process is an generalization of the CRP with an extra "discount" parameter δ , which is a constant discount of the probability mass assigned to each table. Similarly, the probability of a customer choosing a table given the previous seatings is:

$$P(z_i = k|z_{-i}) = \begin{cases} \frac{n_k^{(z_{-i})} - \delta}{i-1+\theta} & k <= Z \\ \frac{Z\delta+\theta}{i-1+\theta} & k = Z+1 \end{cases}$$

[11] Note the table labels are not necessarily unique.

After creating a new table, the discount parameter saves a constant amount of probability mass to creating another new table. The Pitman-Yor process is a CRP when $\delta = 0$, follows a power-law distribution when $0 < \delta < 1$, and is exchangeable (Goldwater et. al. 2006). We use it later as our simple unigram language model based on the Bayesian form of Kneser-Ney smoothing (equivalent to the above hierarchical uni-gram/uniform model using the Pitman-Yor process instead of the CRP). (Teh 2006)

6.2 The Infinite Hidden Markov Model

Beal et al. (2002) define the infinite hidden Markov model as a hidden Markov model with an implicit countably infinite number of possible states. It is a non-parametric model. This means the number of states grow naturally with the data; not there are no parameters in the probability distributions. Since a DP controls the transition probabilities we can "implicitly integrate out" the infinite hidden states during inference. Since we want to develop recurring trajectories, our transition probabilities are defined via two coupled DPs.[12] The bottom DP (with parameter α_{bot}) represents the decision between performing a pre-existing transition or creating a new state transition. We can think of the pre-existing transitions as the tables of this DP and a new state transition as creating a new table. When creating a new table, we partition the probability mass assigned to each state according to the top DP. This top DP (with parameter α_{top}) represents a bias towards popular states when creating new transitions (preferential attachment). The customers at each of its tables represent the number of times we have ever gone to each state. Additionally, creating a new table in the top DP creates a new hidden state. Thus, our transition probability matrix is dynamic. Define $q_1, \ldots, q_t$ to represent our hidden state sequence up to time t, n_{ij} to be the number of times we have transitioned from state i to state j, and n_i^0 to be the number of times we have ever been in state i. Our probability transition matrix is defined:

$$P(q_{t+1} = j | q_t = i; n, \alpha_{\text{bot}}, \alpha_{\text{top}}) =$$
$$\begin{cases} \frac{1}{Z_{\text{bot}}} n_{ij} & \text{we pick a pre-existing transition} \\ \frac{1}{Z_{\text{bot}} Z_{\text{top}}} \alpha_{\text{bot}} n_j^0 & \text{we pick a new transition to a pre-existing state} \\ \frac{1}{Z_{\text{bot}} Z_{\text{top}}} \alpha_{\text{bot}} \alpha_{\text{top}} & \text{create a new state and transition to it} \end{cases}$$
where $Z_{\text{bot}} = \sum_j n_{ij} + \theta_{\text{bot}}$ and $Z_{\text{top}} = \sum_j n_j^0 + \theta_{\text{top}}$

One of the nice properties of modeling transitions via hierarchical DP is that DP generate power-law distributions. Our resulting HMM favor certain transitions in a power-law fashion. In our case, the hidden states correspond

[12] Unlike Beal et al (2002), we do not reserve mass for a self-transitions. The DPs are not hierarchical in the standard Bayesian modeling sense. See Teh et al. (2006)

to semantic states (like Barzilay and Lee (2004)). Thus it is reasonable to assume certain sentence topic transitions are favored over others (preferential attachment). Beal et. al. (2002) describe a similar formulation for emission probabilities; however, we insert our local entity grid as our emission generator and thus their emission discussion is not relevant.

6.3 Combined Model - A Hierarchical Generative Document Story

Our model combines the (global) Barzilay HMM model with the (local) relaxed entity grid model into one fully integrated unified model. One benefit of generative models is modularity and now we reap the benefits! As discussed before, we can create a generative hidden Markov model whose hidden states represent the semantic states of the sentence. Thus, one portion of our model is governed by the transition of sentence topic through the document. If our emission generator is a bigram add pseudocount language model, we have a generative Bayesian formulation of Barzilay and Lee (2004).

Instead, shown in figure 3's graphical model, we split our emission function into two separate generative processes. First, denoted E_i on figure 3, we generate all the known entities in the sentence the relaxed entity grid. Since hidden topic states gain meaning through lexicalization, we add the word token to the conditioning information of our entity grid model. This is a sparse distribution. Accordingly, so we interpolate over the unlexicaled entity grid model.[13] Second, we generate the new entities, denoted N_i, and non-entity words, denoted W_i, from a unigram Pitman-Yor language model (see Teh 2006 for a thorough explanation of hierarchical Pitman-Yor language models). Figure 3 shows a time-slice of our generative process given the previous entities and the last sentence. Although generating the entire sentence is separated into different generative processes, these generative processes are not independent. The graphical model defines a structure of independence and conditional independence. See Jordan and Weiss 2002 for an excellent tutorial on inference and learning in graphical models (Bayesian networks).

The graphical model defines a system of equations, which comprise our model. In addition to defining a probability on documents (which we interpret as a metric of document coherence), we optimize the parameters of the inner processes by Gibbs sampling. Finally, we can optimize the values of outer interpolation hyperparameters by Metropolis-Hastings Sampling. Using the above graphical model (figure 3) as a guide, the following set of generative processes define our model:

[13] It is possible to interpolate again over the entity grid with one less piece of Markov history and so on until we are left with no context information. In practice, we found this to have little effect.

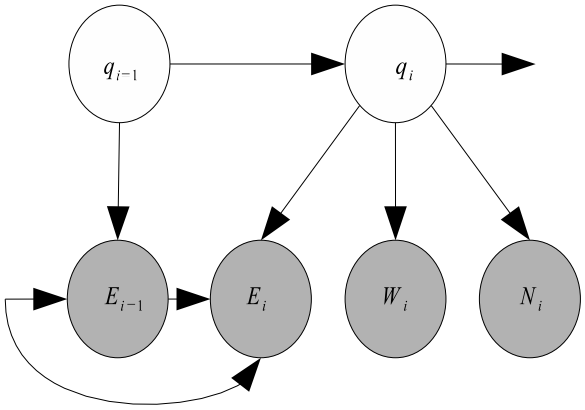

Figure 3: A graphical model representation of the generative process behind one sentence

$$
\begin{aligned}
&q_i|q_{i-1}, \alpha_{\text{bot}}, \alpha_{\text{top}} \sim \text{iHMM}(\cdot) \\
&\forall r \in \vec{R}_i \forall e_{i,j} \in \vec{K}_i : \\
&e_{i,j}|r, \vec{E}_{i-1}, \ldots, \vec{E}_{i-h}, lex(e_j), k_j, q_i, \theta_{\text{EG}} \sim \text{RelaxedEGrid}(\cdot; \theta_{\text{EG}}) \\
&\vec{W}_i, \vec{N}_i|q_i\vec{\theta}_{\text{LM}} \sim \text{PY}(\cdot|q_i; \vec{\theta}_{\text{LM}})
\end{aligned}
$$

The relaxed entity grid model is slightly more complicated than we presented above. In order to make the state information useful, we add the lexical word of the entity to the conditioning information ($lex(e_j)$). Whenever lexical information is introduced, sparsity problems occur. Thus, we backoff to the entity grid without conditioning on the current sentence state and lexical item. The resulting smoothed equation is:

$$
\begin{aligned}
&P(e_{i,j}|r_i, e_{i-1,j}, \ldots, e_{i-h,j}, lex(e_j), k_j, q_i; \theta_{\text{EG}}) = \\
&\frac{C(e_{i,j}, \ldots, e_{i-h,j}, lex(e_j), k_j, q_i) + \theta_{\text{EG}} P(e_{i,j}|e_{i-1,j}, \ldots, e_{i-h,j}, k_j)}{C(e_{i,j}, \ldots, e_{i-h,j}, lex(e_j), k_j, q_i) + \theta_{\text{EG}}}
\end{aligned}
$$

where $P(e_{i,j}|e_{i-1,j}, \ldots, e_{i-h,j}, k_j)$ is the backed-off relaxed entity grid model discussed previously. This is similar to a two level hierarchical DP model. The parameter, θ_{EG}, determines how often we create a new table. In other words, it controls the influence of the parent distribution or the relaxed entity grid model. This is equivalent to distributing pseudocounts by their back-off probability.[14] Finally, the Pitman-Yor language model generates the non-entity words and new entities.

Now that we can calculate the probability of a document, how do we optimize the parameters on our training corpus? Fortunately, the Dirichlet

[14]This may seem complicated (and it is) but after staring at the equations for long enough, it should make sense. The benefit of this approach is that we obtain principled optimized patterns relatively easily.

and Pitman-Yor processes are exchangeable, which allows us to use Gibbs sampling to optimize and train the inner interpolation processes. To learn parameter values from the training set, we first produce and seat all the events from each document in the training set into our processes. There are three different types of events, which correspond to the three generative processes: new entities to the new entity language model, non-entity words from the non-entity language model, and known entities from the entity grid model. Each event is added to its corresponding bottom level process and propagated through its parents to the hidden states. All events for a sentence are linked into one event in the underlying hidden state transitions. Its emission probability is the product of the probabilities of generating each individual event in the sentence from its appropriate generative process (e.g. known entities from the relaxed entity grid).

After all events are seated, we have an (informative) starting distribution to initialize Gibbs sampling. Fortunately, the parent processes define how their child processes create new tables (the backed-off distribution) and thus Gibbs sampling amounts to reseating the child processes. As all of these hierarchical processes have a parent distribution, we define an uninformative prior distribution over the parameters of the root processes. We refine the value of the hyperparameter by performing Metropolis-Hastings sampling. This amounts to randomly accepting randomly proposed hyperparameter values at a rate proportional to the ratio of the likelihood of all events under the old and proposed values. Ultimately, we train our model by the following steps:

1. Create all sentence events, $e_1, \ldots, e_T$, from the training set.
2. In order[15], seat all events into the appropriate bottom-level processes and update their appropriate parent processes
3. For training iteration 1 to ...
   (a) For $i = 1$ to T
      i. Remove sentence e_i from the iHMM processes
      ii. For all observed events, o_{cur} within e_i
         A. Remove o_{cur} from its bottom process and all parent processes
      iii. Seat e_i into the iHMM given the iHMM with all other events
      iv. For all observed events, o_{cur} within e_i
         A. Seat o_{cur} into its bottom process according to it and all parent processes without it
         B. Update appropriate parent processes to include o_{cur}

[15] Order in all of these steps does not matter as long as all events are visited infinitely often

(b) For all root processes,

i. Generate new proposed root theta, θ', from an uninformative distribution. $\theta' \sim Gamma(\alpha, \beta)$[16]

ii. Calculate the likelihood of all events under the model with the current theta value, $L(\theta)$, and proposed new theta value $L(\theta')$

iii. Sample $x \sim U(0, 1)$

iv. $\theta \leftarrow \theta'$ if $\frac{L(\theta')}{L(\theta)} \leq x$

All steps within (a) compose one full Gibbs sampling step over the training set. The steps within (b) compose one full Metropolis-Hastings step. After a number of training iterations, we are guaranteed to converge on the proper distributions in all of our processes. In practice, our model converges in under ten iterations through the training set. (Elsner et. al. 2007)

6.4 Using Simulated Annealing to Impose Order

From the above discussion, we can train and learn parameters for our model and calculate the probability a sentence ordering. Unfortunately, given "a bag of sentences" (an unordered document), it is not trivial to find the optimal ordering of the sentences given a metric that calculates the value of a particular ordering. In fact, Althaus et. al. (2004) have shown that the ordering problem is comparable to the Travelling Salesman Problem, is NP-complete, and "any polynomial algorithm will compute arbitrarily bad solutions on unfortunate inputs." Thus, previous researchers have used different approaches to arrive at their solution: A* (Soricut and Marcu 2006), reduction to TSP (Althaus et al 2004), and Viterbi-style approximation (Barzilay and Lee 2004). Unfortunately, these methods use the fact that the models are Markovian. Due to the known entities sets ($\vec{K}_i$), our model is not Markovian. Instead, we use simulated annealing to approximate the optimal ordering, which in practice rarely yielded estimated search errors. (Elsner et. al. 2007) We follow Soricut and Marcu (2006) in defining an estimated search error as when $P(\text{solution}) < P(\text{original document})$ (under the model).

Simulated annealing is an approximation method of optimization for an arbitrarily complex function. Given the current state, a neighborhood function, and a temperature schedule, simulated annealing randomly chooses between greedy hill-climbing and a less optimal state as its next current state. It picks less optimal next states randomly with probability proportional to the ratio of the function's values of the two states and the current temperature of the system. In theory the method is guaranteed to converge to the global minimum, although for most problems, this would take an

[16] Changing α and β has little effect. In our experiments we use $\alpha = \beta = 1$

exponential time temperature schedule (and thus we gain nothing by using simulated annealing). (Russell and Norvig 2003) For the sentence ordering problem, our two transition functions, $\gamma_1(\cdot)$ and $\gamma_2(\cdot)$, transform the current optimal ordering by either swapping two random sentences or moving a random sentence to a random location in the document (respectively). Unfortunately, this approach is not scalable to large documents because the amount of time needed to search the space to find optimal solution increases dramatically. Additionally,the transition functions may no longer be appropriate for documents with complex paragraph structure. In practice, we used the following steps to perform simulated annealing and find an approximate optimal solution:

1. Initialize arbitrary ordering $\vec{\sigma}_{\text{cur}} = (\sigma_1, \ldots, \sigma_n)$
2. For $T = 1$ to .1 by the temperature schedule $T \leftarrow 0.99995T$
   (a) $x \sim U(0, 1)$
   (b) if $x \leq .75$, $\gamma = \gamma_1(\cdot)$
   (c) otherwise $\gamma = \gamma_2(\cdot)$
   (d) $\vec{\sigma}_{\text{temp}} = \gamma(\vec{\sigma}\text{cur})$
   (e) $x \sim U(0, 1)$
   (f) if $T\frac{P(\vec{\sigma}_{\text{temp}})}{P(\vec{\sigma}\text{cur})} \geq x$, $\vec{\sigma}_{\text{cur}} = \vec{\sigma}_{\text{temp}}$
3. Approximate optimal ordering is $\vec{\sigma}_{\text{cur}}$

7 Experiments

There are two standard evaluation tasks in the sentence ordering literature: binary classification and sentence ordering. The binary classification problem explores the model's success at choosing between the correct ordering of sentences of a document from a random ordering of the same sentences. Lapata and Barzilay (2005) show that human coherence judgments correlate strongly with the original ordering of the document. In a human judgment experiment, inter-subject agreement is substantially higher than model-human agreement (0.768 vs. 0.322 at best). (Lapata and Barzilay 2005) Performance is quantified by the percentage of times the model can distinguish correctly between the actual and random orderings. As the number of sentences in a document grows, the number of possible permutations of the sentences grow exponentially.[17] Accordingly, the number of possible permutations that preserve the Markov clusters described in the local coherence section shrinks dramatically.

[17] Thus, we only test on a random subset of all the possible orderings of each document

Remember from the local coherence section, the local coherence models calculate the probability of a sentence ordering from the syntactic role transitions of entities. Thus, the model bases its judgement according to whether or not Markov clusters occur. So, we expect local coherence models to perform well on this task. Since the global models are not acutely tuned to the Markov clusters, we do not expect excellent performance. Barzilay and Lee (2004) do not report binary classification scores, but we evaluated their model using our own test permutations. Finally, the combined models should perform at least as well as the local coherence models. Unfortunately, Soricut and Marcu(2006) do not report binary classification scores; however, they do mention successfully reimplementing Barzilay and Lapata (2005)'s entity grid model. If they optimize their mixture weights on binary classification, they should do at least as well as Barzilay and Lapata (2005).[18] It is uncertain that the mixture weights optimized on the sentence ordering task would perform well on the binary classification task. As shown in the last section, our combined model is not trained with a particular task in mind.

The other standard task is imposing a sentence ordering on an unordered bag of sentences. This task is more difficult than the binary classification problem, particularly for the local coherence (specifically entity transition based) models. The local coherence models can group sentences together, but is indifferent to different orderings of sentences within the clusters and the orderings of the clusters themselves within the document. Thus, we would be surprised to see local models performing better than random. On the other hand, global coherence models are specifically built with this purpose in mind. Thus, we should expect their performance to be much better. Finally, it is our hope that the unification of local and global coherence models yields improvement on the sentence ordering task.

The Kendall's τ metric evaluates proposed sentence orderings. Lapata (2006) has shown that the Kendall's τ metric reflects human coherence judgements for different orderings of the same sentence. Intuitively, the metric is proportional to the number pairwise swaps between the proposed ordering of the document and the correct ordering. It ranges from -1 to 1 where 1 signifies the correct ordering and -1 signifies the complete opposite ordering. Additionally, $\tau = 0$ signifies random performance. It is evaluated:

$\tau(\sigma) = 1 - \frac{S(\sigma)}{N(N-1)}$
where $S(\sigma)$ is the minimum number of pairwise swaps between the correct ordering and σ and N is the number of sentences in the article.

[18] set all mixture weights to zero except for the entity grid which is set to one

Model	τ	Discr. (%)
[Barzilay and Lapata2005]	-	90
[Barzilay and Lee2004]	.44	74
[Soricut and Marcu2006]	**.50**	-
Our Unified Model [Elsner et al.2007]	**.50**	**94**

Table 1: Results for our two tasks on the AIRPLANE test corpus

7.1 Airplane Corpus

In the last few years, the majority of its experiments focus on the AIRPLANE corpus . It is composed of 200 articles (using the same split as Barzilay and Lapata (2005) into 100 training and test articles) from the National Transportation Safety Board describing airplane crashes. Fortunately for the sentence ordering problem, the articles are generally short without being trivial. The average sentence length is 11.5 sentences. We reiterate similar complaints to Elsner et al (2007) regarding the strange introduction sentence(s) of the documents in the corpus. Forty-six percent of the training documents begin with the same two sentence preamble: "This is preliminary information, subject to change, and may contain errors. Any errors in this report will be corrected when the final report has been completed." This makes ordering the introduction of these documents trivial! Additionally, many of the other documents start abruptly with technical jargon and unintroduced definite noun phrase. It would be difficult for human readers to order these articles. Despite these problems, one advantage of the corpus is its small document length and formulaic style, which affords reoccuring lexical and semantic patterns. We propose a movement to the Wall Street Journal corpus used in the parsing literature. As the Wall Street Journal contains longer articles, we need to develop new techniques that scale to impose an ordering on a bag of sentences.

7.2 Results

Table 1 shows the results of the two models on the two different tasks on the AIRPLANE test corpus. We see that our model performs at the state of the art in both tasks. On one hand, the results of imposing a sentence-ordering task are inconclusive betwen our model and Soricut and Marcu(2006)'s mixture model. However, this does not invalidate the usefulness of our unified model. First, our model is generative and thus can be easily incorporated as a module of another model. Since one goal of these models is to aid multi-document summarization models, this is an important advantage. Additionally, our model uses a strict subset of the document information the mixture model uses. For example, our language models are unigram and we do not incorporate any of the IBM models. Thus, it is reasonable to expect that in-

Model	τ	Discr. (%)
Naive Entity Grid	.17	81
Relaxed Entity Grid	.02	87
Unified w/ Naive EGrid	.39	85
Unified w/ Relaxed EGrid	**.54**	**96**

Table 2: Results for 10-fold cross-validation on the `AIRPLANE` training corpus for different versions of our model

troducing our model into their mixture model would yield further improved results. However, this is speculative; we did not attempt to incorporate our model into their mixture model.

Table 2 illustrates the quantitative motivation for moving from the naive to relaxed entity grid. Considering the two isolated local coherence models, the Relaxed Entity Grid model provides an improvement over the Naive Entity Grid model for 10-fold cross-validation on the `AIRPLANE` training corpus. Surprisingly, the isolated Naive Entity Grid model performs much better on the τ task. We believe this is due to the unrealistic preamble in a large portion of the training documents. The Naive Entity Grid model can use information from the preamble to base its judgment; however, under the Relaxed Entity Grid model, most entities in the preamble are considered new and thus ignored. Fortunately, the results of the unified models match our predictions. The unification provides increased performance on both tasks for both local models. Additionally, the gain of unifying the relaxed entity grid and hidden Markov models is greater than any of our other unified models. Although we are only beginning testing on the Wall Street Journal corpus, we suspect this gain is domain-general. At this time, we do not have results for the Wall Street Journal corpus, but we expect the same trend as the `AIRPLANE` corpus.

8 Conclusion

Initially, we motivated a document coherence model that accounts for both local and global constraints. In particular, we motivated an interest for sentence ordering models that distinguish between different permutations of the sentences in a document. Although most common language models discriminate between different orderings of words, these models do not account for different sentence orders. Human readers have strong constraints for sentence order (e.g., news articles) and thus sentence order is essential in natural language generation tasks.

The first sentence ordering models focus on local constraints. Specifically, they track the change of word usage from sentence to sentence (or a Markov horizon of sentences). Although successful at distinguishing be-

tween random and the original ordering of a document, local models do not impose strong enough constraints to compute a coherent ordering of sentences. Accordingly, global models attempt to capture the flow of sentence topic through the document. They are successful at imposing coherent sentence orderings. Unfortunately, these models have difficulty tracking local constraints between entities. Hence, they lose obvious connections between related sentences. It seems natural to combine these models into a unified framework.

With this goal in mind, Soricut and Marcu (2006)'s mixture model formulated document coherence as a mixture of local and global constraints. Although successful at improving the isolated models, it does not capture the interdependence between local and global models. Additionally, it cannot be principally incorporated into the tasks they were motivated for. On the other hand, the two main advantages of Bayesian generative modeling address these two points: principled model interdependence and modularity. Using the infinite hidden Markov model, Dirichlet Processes, and the relaxed entity grid model, we reformulated the local and global models into one unified generative process. This unified model performs at the state of the art on both the sentence ordering and discrimination tasks. It is easy to incorporate as a module in larger natural language systems. Additionally, there are obvious directions for improving the models. Therefore, we provide a compelling, intuitive, combined framework for modeling document coherence.

9 Acknowledgments

I would like to begin by thanking my mother and father, Theresa and Arthur Austerweil. None of this work could have been possible without their endless support. Additionally, I am thankful for the fun and love my brother and sister-in-law have given me. I am indebted to Elaine Labrocca for fostering my initial research interest in high school. I would like to acknowledge Eugene Charniak for being a fantastic advisor and preacher of generative modeling. Additionally, his comments on my thesis helped improve my technical writing immensely. Fortunately, Micha Elsner and I happened to come across the same research idea independently at the same time. Collaborating with him (from pair programming to working on the white board together) has been a great experience and he is a great friend. During the beginning stages of this research work, I am thankful for the insightful comments from Regina Barzilay and Mirella Lapata that jump started our work. Throughout the year, my friends at the Brown Laboratory for Linguistic Information Processing have provided great comments and a fun research group environment. Special thanks to Sharon Goldwater for putting up with my many questions about Chinese Restaurant Processes. Additionally, I am thankful

to Tom Griffiths, my future graduate advisor at UC Berkeley, who has enriched my modeling intuitions and has been incredibly supportive. I would like to thank Mark Johnson for useful comments on my thesis. Financially, I am grateful to the Karen T. Romer Foundation for supporting me with a summer fellowship. Finally, I am fortunate to have many great friends up to and including college. The amount of love they provide is incredible and I am indebted to them.

References

[Althaus et al.2004] E. Althaus, N. Karamanis, and A. Koller. 2004. Computing locally coherent discourses. In *In the Proceedings of the Association for Computational Linguistics*.

[Barzilay and Lapata2005] Regina Barzilay and Mirella Lapata. 2005. Modeling local coherence: An entity-based approach. In *Proceedings of the ACL*, pages 141–148.

[Barzilay and Lee2004] Regina Barzilay and Lillian Lee. 2004. Catching the drift: Probabilistic content models, with applications to generation and summarization. In *Proceedings of the North American Association for Computational Linguistics*, pages 113–120.

[Beal et al.2002] M. J. Beal, Z. Ghahramani, and C. Rasmussen. 2002. The infinite hidden markov model. 13:577–584.

[Blei et al.2003] D. M. Blei, A. Y. Ng, and M. Jordan. 2003. Latent dirichlet allocation. 3:993–1022.

[Charniak and Johnson2005] Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine n-best parsing and maxent discriminative reranking. In *In the Proceedings of the Association for Computational Linguistics*, pages 173–180.

[Duboue and McKeown2003] P. A. Duboue and K. R. McKeown. 2003. Statistical acquisition of content selection rules for natural language generation. In *Proc. Empirical Methods in Natural Language Processing*. The Association for Computational Linguistics.

[Elsner et al.2007] M. Elsner, J. Austerweil, and E. Charniak. 2007. A unified local and global model for discourse coherence. In *Proceedings of the North American Association for Computational Linguistics*.

[Evans et al.2004] David Kirk Evans, Judith L. Klavans, and Kathleen R. McKeown. 2004. Columbia newsblaster: Multilingual news summarization on the web. In *HLT-NAACL*.

[Foltz et al.1998] P. W. Foltz, W. Kintsch, and T. K. Landauer. 1998. Textual coherence using latent semantic analysis. *Discourse Processes*, 25(2&3):285–307.

[Goldwater et al.2006] S. Goldwater, T. Griffiths, and M. Johnson. 2006. Interpolating between types and tokens by estimating power-law generators. 18.

[Goldwater2006] Sharon Goldwater. 2006. *Nonparametric Bayesian Models of Lexical Acquisition*. Ph.D. thesis, Cognitive and Linguistic Sciences Department, Brown University.

[Griffiths et al.2005] T. Griffiths, M. Steyvers, D. Blei, and J. Tenenbaum. 2005. Integrating topics and syntax. 17.

[Grosz et al.1995] Barbara J. Grosz, Aravind K. Joshi, and Scott Weinstein. 1995. Centering: A framework for modeling the local coherence of discourse. 21(2):203–225.

[Hofmann1999] T. Hofmann. 1999. Probabilistic latent semantic indexing. In *Proceedings of the Twenty-Second Annual International SIGIR Conference*.

[Lapata and Barzilay2005] Mirella Lapata and Regina Barzilay. 2005. Automatic evaluation of text coherence: Models and representations. In *IJCAI*, pages 1085–1090.

[Lapata2006] M. Lapata. 2006. Automatic evaluation of information ordering: Kendalls tau. 32(4).

[Manning and Schütze1999] Chris Manning and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, Massachusetts.

[McKeown1985] Kathleen R. McKeown. 1985. *Text Generation*. Cambridge University Press, New York, New York.

[Miltsakaki and Kukich2004] E. Miltsakaki and K. Kukich. 2004. Evaluation of text coherence for electronic essay scoring systems. 10(1):25–55.

[Mitzenmacher2003] Michael Mitzenmacher. 2003. A brief history of generative models for power law and lognormal distributions. 1(2):226–251.

[Reiter and Dale2000] E. Reiter and R. Dale. 2000. *Building Natural Language Generation Systems*. Cambridge University Press, Cambridge.

[Russell and Norvig2003] Stuart Russell and Peter Norvig. 2003. *Artificial Intelligence: A Modern Approach*. Prentice Hall, New Jersey.

[Soricut and Marcu2006] Radu Soricut and Daniel Marcu. 2006. Discourse generation using utility-trained coherence models. In *Proceedings of the Association for Computational Linguistics*.

[Sripada et al.2001] Somayajulu G. Sripada, Ehud Reiter, Jim Hunter, and Jin Yu. 2001. A two-stage model for content determination. In *ACL-EWNLG*.

[Teh et al.2006] Yee Whye Teh, Michael I. Jordan, Matthew J. Beal, and David M. Blei. 2006. Hierarchical dirichlet processes. 101:1566–1581.

[Teh2006] Y. W. Teh. 2006. A bayesian interpretation of interpolated knesner-ney. Technical Report TRA2/06, National University of Singapore.

[Zipf1935] George Kingsley Zipf. 1935. *The Psycho-Biology of Language: An Introduction to Dynamic Philology*. The Riverside Press, Boston.