Mixed Membership Markov Models for Unsupervised Conversation Modeling

Michael J. Paul
Center for Language and Speech Processing
Johns Hopkins University
Baltimore, MD 21218, USA
mpaul@cs.jhu.edu

Abstract

Recent work has explored the use of hidden Markov models for unsupervised discourse and conversation modeling, where each segment or block of text such as a message in a conversation is associated with a hidden state in a sequence. We extend this approach to allow each block of text to be a mixture of multiple classes. Under our model, the probability of a class in a text block is a log-linear function of the classes in the previous block. We show that this model performs well at predictive tasks on two conversation data sets, improving thread reconstruction accuracy by up to 15 percentage points over a standard HMM. Additionally, we show quantitatively that the induced word clusters correspond to speech acts more closely than baseline models.

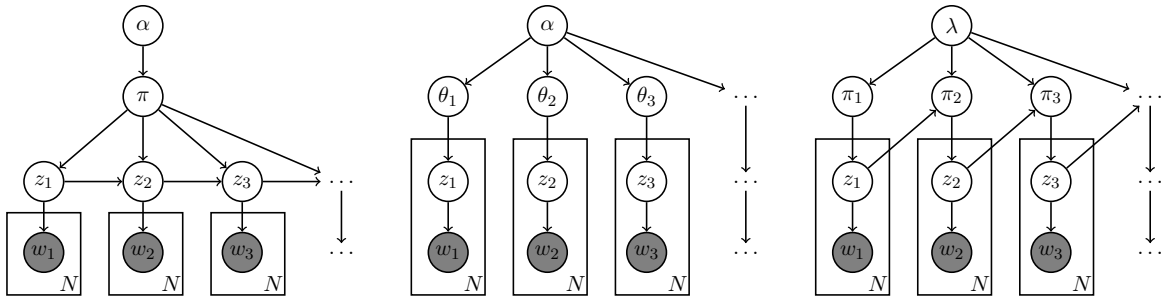
1 Introduction

The proliferation of social media in recent years has lead to an increased use of informal Web data in the language processing community. With this rising interest in social domains, it is natural to consider models which explicitly incorporate the conversational patterns of social text. Compared to the naive approach of treating conversations as flat documents, models which include conversation structure have been shown to improve tasks such as forum search (Elsas and Carbonell, 2009; Seo et al., 2009), question answering and expert finding (Xu et al., 2008; Wang et al., 2011a), and interpersonal relationship identification (Diehl et al., 2007).

While conversational features may be important, Web-derived corpora are not always annotated with this information, and the nature of conversations on the Web can vary wildly across domains and venues. Addressing these concerns, there has been recent work with *unsupervised* models of Web conversations based on hidden Markov models (Ritter et al., 2010), where each state corresponds to a conversational class or "act." Unlike more traditional uses of HMMs in which a single token is emitted per time step, HMM emissions in conversations correspond to entire blocks of text, such that an entire message is generated at each step. Because each time step is associated with a block of variables, we refer to this type of HMM as a *block HMM* (Fig. 1a).

While block HMMs offer a concise model of inter-message structure, they have the limitation that each text block (message) belongs to exactly one class. Many modern generative models of text, in contrast, allow documents to contain many latent classes. For example, topic models such as Latent Dirichlet Allocation (LDA) (Blei et al., 2003) assume each document has its own distribution over multiple classes (often called "topics"). For many predictive tasks, topic models outperform single-class generative models such as Naive Bayes. These properties could similarly be desirable in conversation modeling. An email might contain a request, a question, and an answer to a previous question – three distinct dialog acts within a single message. This motivates the desire to allow a message to be a *mixture* of classes.

In this paper, we introduce a new type of model which combines the functionality of topic models, which posit latent class assignments to each individual token, with Markovian sequence models, which



(a) Block HMM (b) LDA (c) M^4

Figure 1: The graphical models for the block HMM (left) where each block of tokens depends on exactly one latent class, LDA (center) where each token individually depends on a latent class, and M^4 (right) where the class distributions are dependent across blocks. Some parameters are omitted for simplicity. This figure depicts the Bayesian variant of the block HMM (Ritter et al., 2010) where the transition distributions π depend on a Dirichlet(α) prior.

govern the transitions between text blocks in a sequence. We generalize the block HMM approach so that there is no longer a one-to-one correspondence between states in the Markov chain and latent discourse classes. Instead, we allow a state in the HMM to correspond to a mixture of many classes: we refer to this family of models as **mixed membership Markov models (M^4)**. Instead of defining explicit transition probabilities from one class to another as in a traditional HMM, we define the distribution over classes as a function of the entire histogram of class assignments of the previous text segment. We define our model using the same number of parameters as a standard HMM (§2), and we present a straightforward approximate inference algorithm (§3).

While we introduce a general model, we will focus on the task of unsupervised conversation modeling. Specifically, we build off the Bayesian block HMMs used by Ritter et al. (2010) for modeling Twitter conversations, which will be our primary baseline. After discussing related work (§4), we present experimental results on a set of Twitter conversations as well as a set of threads from CNET discussion forums (§5). We show that M^4 increases thread reconstruction accuracy by up to 15% compared to the HMM of Ritter et al. (2010), and we reduce variation of information against speech act annotations by an average of 18% from HMM and LDA baselines. To the best of our knowledge, this work is the first attempt to quantitatively compare unsupervised models against gold standard speech act annotations.

2 M^4: Mixed Membership Markov Models

In this section, we extend the block HMM by introducing mixed membership Markov models (M^4).

Under the block HMM, as utilized by Ritter et al. (2010), messages in a conversation flow according to a Markov process, where the words of messages are generated according to language models associated with a state in a hidden Markov model. The intuition is that HMM states should correspond to some notion of a conversation "act" such as QUESTION or ANSWER. The intuition is the same under M^4, but now each token in a message is given its own class assignment, according to a class distribution for that particular message. A message's class distribution depends on the class assignments of the previous message, yielding a model that retains sequential dependencies between messages, while allowing for finer grained class allocation than the block HMM. Modeling messages (or more generally, text blocks) as a mixture of multiple classes rather than a single class gives rise to the "mixed membership" property.

In the subsections below, we formalize and analyze this new model.

2.1 Structure Assumptions

We first define the discourse structure and terminology we will be assuming. The discourse structure is a directed graph, where nodes correspond to segments of a document (which we will refer to as "blocks" of text), and the edges define the dependencies between them.

Thus, a text *block* is a set of tokens, while a *document* consists of the discourse graph and all blocks associated with it. In the context of modeling conversation threads, which will be the focus of our experiments later, we will assume a block corresponds to a single message in a thread. The parent of a message m is the message to which it is a response; if a message is not in response to anything in particular, then it has no parent. Any replies to the message m are the children of m. The thread as a whole is called a document.

The discourse graph should be acyclic. A directed acyclic graph (DAG) offers a flexible representation of discourse (Rosé et al., 1995), but for simplicity, we will restrict this and assume that each subgraph is a tree; i.e. no message has multiple parents. The graph as a whole may be a forest: for example, someone could write a new message in a conversation that is not directly in reply to any previous message, so this message would not have any parents, and would form the root of a new tree in the forest.

2.2 Generative Story

Extending the block HMM, latent classes in M^4 are now associated with each individual token, rather than one class for an entire block. The key difference between the generative process behind M^4 and the block HMM is that the transition distributions are defined with a log-linear model, which uses class assignments in a block as features to define the distribution over classes for the children of that block. Put another way, a state in M^4 corresponds to a class histogram, and transitions between states are functions of the log-linear parameters.

Given a block b, we will use the notation $\mathbf{b}$ to denote the block's feature vector, which consists of the histogram of latent class assignments for the tokens of b.[1] There are K classes. Additionally, we assume each feature vector has an extra cell containing an indicator denoting whether the block has no parent – this allows us to learn transitions from a "start" state. We also include a bias feature that is always 1, to learn a default weight for each class. There are thus $K + 2$ features which are used to predict the probability of each of the K classes. The features are weighted by transition parameters, denoted λ. The random variable z denotes a latent class, and ϕ_z is a discrete distribution over word types – that is, each class is associated with a unigram language model. The transition distribution over classes is denoted π, which is given in terms of λ and the feature vector of the parent block.

Under this model, a corpus $\mathcal{D}$ is generated by:

1. For each (j, k) in the transition matrix $\Lambda_{K \times K+2}$:
   (a) Draw transition weight $\lambda_{jk} \sim \mathcal{N}(0, \sigma^2)$.
2. For each class j:
   (a) Draw word distribution $\phi_j \sim \text{Dirichlet}(\omega)$.
3. For each block b of each document d in $\mathcal{D}$:
   (a) Set class probability $\pi_{bj} = \frac{\exp(\lambda_j^{\mathrm{T}} \mathbf{a})}{\sum_{j'} \exp(\lambda_{j'}^{\mathrm{T}} \mathbf{a})}$ for all classes j, where $\mathbf{a}$ is the feature vector for block a, the parent of b.
   (b) For each token n in block b:
      i. Sample class $z_{(b,n)} \sim \pi_b$.
      ii. Sample word $w_{(b,n)} \sim \phi_z$.

For each block of text in a document (e.g. each message in a conversation), the distribution over classes π is computed as a function of the feature vector of the block's parent and the transition parameters (feature weights) Λ. Each λ_{jk} has an intuitive interpretation: a positive value means that the occurrence of class k in a parent block increases the probability that j will appear in the next block, while a negative value reduces this probability.

The observed words of each block are generated by repeatedly sampling classes from the block's distribution π, and for each sampled class z, a single word is sampled from the class-specific distribution over words ϕ_z. In contrast, under the block HMM, a class z is sampled once from the transition distribution, and words are repeatedly sampled from ϕ_z.

We place a symmetric Dirichlet prior on each ϕ with concentration parameter ω, which smoothes the word distributions, and we place a 0-mean Gaussian prior on each λ parameter, which acts as a regularizer. The graphical diagram is shown in Figure 1 along with the block HMM and LDA. This figure

[1] One could also use other functions of the class histograms rather than the raw counts themselves. For example, we experimented with binary indicator features (i.e. "does class k appear anywhere in block b?"), but this performed consistently worse in early experiments, and we do not consider this further.

shows how M^4 combines the sequential dependencies of the block HMM with the token-specific class assignments of LDA.

2.3 Discussion

Like the block HMM, M^4 is a type of HMM. A latent sequence under M^4 forms a Markov chain in which a state corresponds to a histogram of classes. (For simplicity, we are ignoring the extra features of the start state indicator and bias in this discussion.) If we assume *a priori* that the length of a block is unbounded, then this state space is $\mathbb{N}^K$ where $0 \in \mathbb{N}$. The probability of transitioning from a state $\mathbf{b}$ to another state $\tilde{\mathbf{b}} \in \mathbb{N}^K$ is:

$$P(\mathbf{b} \rightarrow \tilde{\mathbf{b}}) \propto \zeta_N \text{Multinomial}(\tilde{\mathbf{b}}|\pi(\mathbf{b}), N) \quad (1)$$

where $N = \sum_k \tilde{\mathbf{b}}_k$, ζ_N is the probability that a block has N tokens,[2] and $\pi(\mathbf{b})$ is the transition distribution given a vector $\mathbf{b}$. This follows from the generative story defined above, with an additional step of generating the number of tokens N from the distribution ζ.

We currently define a block b's distribution π_b in terms of the discrete feature vector $\mathbf{a}$ given by its parent a. We could have instead made π_b a function of the parent's distribution π_a – this would lead to a model that assumes a dynamical system over a continuous space rather than a Markov chain. However, as a generative story we believe it makes more sense for a block's distribution to depend on the actual class values which are emitted by the parent. Similar arguments are made by Blei and Mcauliffe (2007) when designing supervised topic models.

Under a block HMM with one class per block, there are K states corresponding to the K classes, requiring $K \times K$ parameters to define the transition matrix. Under M^4, there is a countably infinite number of states, but the transitions are still defined by $K \times K$ parameters (ignoring extra features). M^4 thus utilizes a larger state space without increasing the number of free parameters.

3 Inference and Parameter Estimation

We must infer the values of the hidden variables $\mathbf{z}$ as well as the parameters for the word distributions Φ and transition weights Λ. Standard HMM dynamic programming algorithms cannot straightforwardly be used for M^4 because of the unboundedly large state space. We instead turn to Markov chain Monte Carlo (MCMC) methods as a tool for approximate inference. We derive a stochastic EM algorithm in which we alternate between sampling class assignments for the word tokens and optimizing the transition parameters, outlined in the following two subsections.

3.1 Latent Class Sampling

To explore the posterior distribution over latent classes, we use a collapsed Gibbs sampler such that we marginalize out each word multinomial ϕ and only need to sample the token assignments $\mathbf{z}$ conditioned on each other. Given the current state of the sampler, we sample a token's class according to:

$$P(z_{(b,n)} = k|\mathbf{z}_{-(b,n)}, \mathbf{w}, \lambda, \omega) \propto \quad (2)$$

$$\frac{\exp(\lambda_k^{\mathrm{T}}\mathbf{a})}{\sum_{k'}\exp(\lambda_{k'}^{\mathrm{T}}\mathbf{a})} \frac{n_k^w + \omega}{n_k + W\omega} \prod_{c \in \mathcal{C}} \prod_j \left(\frac{\exp(\lambda_j^{\mathrm{T}}\mathbf{b})}{\sum_{j'}\exp(\lambda_{j'}^{\mathrm{T}}\mathbf{b})} \right)^{n_c^j}$$

The notation n_k^w indicates the number of tokens with word type w that have been assigned to topic k. W is the vocabulary size. a is the parent block of b, and $\mathcal{C}$ is the set of b's children. $\mathbf{b}$ is the feature vector corresponding to block b (i.e. the class histogram plus the bias feature), where the histogram includes the incremented count of the candidate class k.

This sampling distribution is very similar to that of LDA (Griffiths and Steyvers, 2004), but the distribution over "topics" is now a function of the previous block, which gives the leftmost term. The rightmost term is a result of the dependency of the child blocks ($\mathcal{C}$) on the class assignments of b.

Due to the rightmost term, the complexity of computing the sampling distribution is quadratic in the number of classes, rather than the linear complexity of a single-class HMM. Our assumption is that the number of sequence-dependent classes (e.g. speech acts or discourse states) will be reasonably small. If it is desired to have a large number of latent topics as is common in LDA, this model could be combined with a standard topic model without sequential dependencies, as explored by Ritter et al. (2010).

[2] The distribution over the number of tokens can be arbitrary, as this is observed and does not affect inference. In topic models, this is sometimes assumed to be Poisson (Blei et al., 2003).

3.2 Transition Parameter Optimization

Differentiating the corpus likelihood with respect to Λ yields the standard equation for log-linear models:

$$\frac{\partial \ell}{\partial \lambda_{zk}} = \sum_b \mathbf{a}_k \left(n_b^z - n_b \frac{\exp(\lambda_z^{\mathrm{T}} \mathbf{a})}{\sum_{z'} \exp(\lambda_{z'}^{\mathrm{T}} \mathbf{a})} \right) - \frac{\lambda_{zk}}{\sigma^2} \tag{3}$$

where a is the parent of block b, $\mathbf{a}$ is the feature vector associated with a, n_b^z is the number of times class z occurs in block b and n_b is the total number of tokens in block b.

Standard optimization methods can be used to learn these parameters. In our experiments, we find that we obtain good results by simply performing a single iteration of gradient ascent after each sampling iteration t,[3] with the following update:

$$\lambda_{zk}^{(t+1)} = \lambda_{zk}^{(t)} + \eta(t) \frac{\partial \ell}{\partial \lambda_{zk}} \tag{4}$$

where η is a step size function.

4 Related Work

Hidden Markov models have a recent history as simple models of document structure. Stolcke et al. (2000) used HMMs as a general model of discourse with an application to speech acts (or dialog acts) in conversations. Barzilay and Lee (2004) applied HMMs as an *unsupervised* model of discourse. This work used HMMs to model the progression of sentences in articles, and was shown to be useful for ordering sentences and generating summaries of news articles. More recently, Wang et al. (2011b) experimented with similar tasks using a related HMM-based model called the Structural Topic Model.

Unsupervised HMMs were applied to conversational data by Ritter et al. (2010) who experimented with Twitter conversations. The authors also experimented with incorporating a topic model on top of the HMM to distinguish speech acts from topical clusters, with mixed results. Joty et al. (2011) extended this work by enriching the emission distributions and using additional features such as speaker and position information. An approach to unsupervised discourse modeling that does not use HMMs is the latent permutation model of Chen et al. (2009). This model assumes each segment (e.g. paragraph) in a document is associated with a latent class or topic, and the ordering of topics within a document is modeled as a deviation from some canonical ordering.

Extensions to the block HMM have incorporated mixed membership properties within blocks, notably the Markov Clustering Topic Model (Hospedales et al., 2009), which allows each HMM state to be associated with its own distribution over topics in a topic model. Like the block HMM, this still assumes a relatively small number of HMM states, but with an extra layer of latent variables before the observations are emitted. This is more restrictive than the unbounded state space of M^4.

Decoupling HMM states from latent classes was considered by Beal et al. (1997) with the Factorial HMM, which uses factorized state representations. The Factorial HMM is most often used to model independent Markov chains, whereas M^4 has a dense graphical model topology: the probability of each of the latent classes depends on the counts of all of the classes in the previous block. The trick in M^4 is to define the transition matrix via a function of a limited number of parameters, allowing tractable inference in a model with arbitrarily many states.

In topic models, log-linear formulations of latent class distributions[4] are utilized in correlated topic models (Blei and Lafferty, 2007) as a means of incorporating covariance structure among topic probabilities. Applying log-linear regression to potentially many features was combined with LDA by Mimno and McCallum (2008), who model the Dirichlet prior over topics as a function of document features. In M^4, such features would correspond to the class histograms of previous blocks, introducing additional dependencies between documents.

One topic model that imposes sequential dependencies between documents is Sequential LDA (Du et al., 2010), which models a document as a sequence of segments (such as paragraphs) governed by a Pitman-Yor process, in which the latent topic distribution of one segment serves as the base distribution for the next segment. This is in the spirit

[3] Incremental updates are justified under the generalized EM algorithm (Dempster et al., 1977). Each gradient step with respect to λ corresponds to a generalized M-step, while each sampling iteration corresponds to a stochastic E-step.

[4] This formulation corresponds to the *natural* parameterization of the multinomial distribution.

of our work, where the latent classes in a segment depend on the class distribution of the previous segment. By using the Pitman-Yor process, however, this work assumes topics are positively correlated, i.e. the occurrence of a topic in one segment makes it likely to appear in the next. In contrast, we wish to learn arbitrary transitions, both positive and negative, between the latent classes.

5 Experiments with Conversation Data

We experiment with two corpora of text-based asynchronous conversations on the Web. One of these is annotated with speech act labels, against which we compare our unsupervised clusters. We measure the predictive capabilities of the model via perplexity experiments and the task of thread reconstruction.

5.1 Data Sets

First, we use a corpus of discussion threads from CNET forums (Kim et al., 2010), which are mostly technical discussion and support. This corpus includes 321 threads and a total of 1309 messages, with an average message length of 78 tokens after preprocessing.[5] Second, we use the Twitter data set created by Ritter et al. (2010). We consider 36K conversation threads for a total of 100K messages with average length 13.4 tokens.

Both data sets are already annotated with the reply structure, so the discourse graph is given. We preprocess the data by treating contiguous blocks of punctuation as tokens, and we remove infrequent words. The Twitter corpus has some additional preprocessing, such as converting URLs to a single word type.

5.2 Baseline Models

Our work is motivated by the Bayesian HMM approach of Ritter et al. (2010) – the model we refer to as the block HMM (BHMM) – and we consider this our primary baseline. (See also (Goldwater and Griffiths, 2007) for more details on Bayesian HMMs with Dirichlet priors.) We also compare against LDA, which makes latent assignments at the token-level, but blocks of text are independent of each other. In other words, BHMM models sequential dependencies but allows only single-class membership, whereas LDA uses no sequence information but has a mixed membership property. M^4 combines these two properties.

We use standard Gibbs samplers for both baseline models, and we optimize the Dirichlet hyperparameters (for the transition and topic distributions) using Minka's fixed-point iterations (2003).

5.3 Incorporating Background Distributions

In our experiments, we find that the intrusion of common stop words can make the results difficult to interpret, but we do not want to perform simple stop word removal because common function words often play important roles in the latent classes (i.e. speech acts) of the conversation data we consider here. We instead handle this by extending our model to include a "background" distribution over words which is independent of the latent classes in a document; this was also done by Wang et al. (2011b).

The idea is to introduce a binary switching variable x into the model which determines whether a word is generated from the general background distribution or from the distribution specific to a latent class z. Loosely, if the marginal probability of a word was given by $p(w) = \sum_z p(w|z)p(z)$, the introduction of a background distribution gives the marginal probability $p(w) = p(x = 0)p(w|B) + p(x = 1)\sum_z p(w|z)$. This is common practice and we will not go into detail; see (Chemudugunta et al., 2006) for a general example on sampling switching variables. We augment all three models with a background distribution in exactly the same way, so that the comparison is fair. We use a Beta$(10.0, 10.0)$ prior over the switching distribution.

5.4 Experimental Setup

All of our results are averaged across four randomly initialized chains which are run for 5000 iterations, with five samples collected during the final 500 iterations. We take small gradient steps of decreasing size with $\eta(t) = 0.1/(1000 + t)$.

We set $\sigma^2 = 10.0$ as the variance of the λ weights. We use optimized asymmetric priors as described in §5.2, and we use a symmetric Dirichlet for the word distributions, following Wallach et al. (2009). We sample the scaling hyperparameter ω via

[5] Three messages in this corpus have multiple parents. For the sake of conciseness, we simply remove these threads rather than introducing a method to model multiple parents.

	5	10	15	20	25
CNET					
Unigram	63.07	63.07	63.07	63.07	63.07
LDA	57.16	54.35	52.88	51.63	50.50
BHMM	61.26	61.06	60.92	60.86	60.85
M^4	60.38	59.58	59.26	59.21	59.25
Twitter					
Unigram	93.00	93.00	93.00	93.00	93.00
LDA	83.70	78.40	74.01	70.91	70.16
BHMM	90.51	89.94	89.68	89.59	89.38
M^4	88.44	86.17	85.50	85.55	86.31

Table 1: Average perplexity of held-out data for various numbers of latent classes.

Metropolis-Hastings proposals: we add Gaussian-distributed noise to the log of the current ω, then exponentiate this to yield the proposed $\omega^{(\text{new})}$. This log-space proposal ensures that ω is always positive.

When computing the transition distributions for M^4, we normalize the class histograms so that the counts to sum to 1. This helps with numeric stability because the input vectors stay within a small bounded range.[6]

5.5 Experimental Results

5.5.1 Perplexity

We begin with standard measures of the perplexity of held-out data. For these experiments, we train on 75% of the data, and test on the remaining 25%. We run the sampler for 500 iterations using the word distributions and transition parameters learned during training; we compute the average perplexity from the final ten sampling iterations.

Results for different numbers of classes are shown in Table 1. These results demonstrate the advantage of models with the mixed membership property. Although LDA outperforms both sequence models, this is be expected. Each block's topic distribution is stochastically generated with LDA, whereas in the two sequence models, the distribution over classes is simply a deterministic function of the previous block. This allows LDA to infer parameters that fit the data more tightly. Comparing only the two sequence models, we find that M^4 does significantly better than BHMM in all cases with $p < 0.05$.

[6]Implementations of both M^4 and the block HMM will be available at `http://cs.jhu.edu/~mpaul`

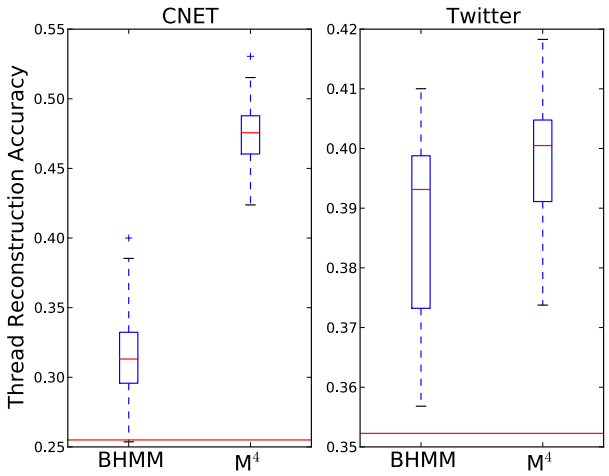


Figure 2: Accuracy at the task of thread reconstruction. The horizontal bar indicates a random baseline.

If capturing sequence information is not important, then LDA may provide a better fit to a corpus than sequence models. In the next two subsections, we will consider tasks where the sequential structure is important, thus LDA is not an appropriate choice.

5.5.2 Thread Reconstruction

A natural predictive task of the sequence models is to reconstruct the discourse graph of a document where the structure is unknown. In the conversation domain, this corresponds to the task of *thread reconstruction* (Yeh and Harnly, 2006; Wang et al., 2011c). Given only a flat structure, can we recover the reply structure of messages in the conversation?

Previous work with BHMM found the optimal structure by computing the likelihood of all permutations of a thread or sequence (Ritter et al., 2010; Wang et al., 2011b). We take a more practical approach and find the optimal structure as part of our inference procedure. We do this by treating the parent of each block as a hidden variable to be inferred. The parent of block b is the random variable r_b, and we alternate between sampling values of the latent classes $\mathbf{z}$ and the parents $\mathbf{r}$. The sampling distributions are annealed, as a search technique to find the best configuration of assignments (Finkel et al., 2005). At temperature τ, we sample a block's parent according to:

$$P(r_b = a|\mathbf{z}, \lambda) \propto \prod_j \left(\frac{\exp(\lambda_j^{\mathrm{T}} \mathbf{a})}{\sum_{j'} \exp(\lambda_{j'}^{\mathrm{T}} \mathbf{a})} \right)^{n_b^j / \tau} \quad (5)$$

For each conversation thread, any message is a candidate for the parent of block b (except b itself) including the dummy "start" block.

As before, we train on 75% of the data, and run this experiment on the remaining 25%. We run the sampler for 500 iterations, cooling τ by 1% after each iteration, where $\tau^{(0)} = 1$. We measure accuracy as the percentage of blocks whose assignment for r_b matches the true parent. For each fold, we run this estimation procedure from five random initializations and average the results. Like Ritter et al. (2010), we do not enforce temporal constraints in the thread structure for this experiment. We are purely evaluating the predictive abilities of the model rather than its performance in a full-fledged reconstruction setup, which would require richer features beyond the scope of this paper.

Figure 2 shows results comparing M^4 against BHMM. Because all blocks are independent under LDA, it cannot be used in this experiment; using LDA would amount to a random baseline.

We plot the distribution of results from various samples and various numbers of classes in $\{5, \ldots, 25\}$. Most of the variance is across folds and samples; we find that there is not a strong trend in accuracy as a function of the number of classes. This suggests that most of the sequence predictions are carried by a small subset of the classes.

On average, M^4 outperforms BHMM by more than 15 points on the CNET corpus. M^4 is also better on the Twitter corpus, but the difference is not so stark. This seems to confirm our intuition that the advantage of M^4 over BHMM is greater when the blocks are longer; tweets may be short enough that the single-class assumption is not as limiting.

5.5.3 Speech Act Discovery

Thus far, we have investigated the predictive power of the model, but we would also like to determine if the inferred clusters correspond to human-interpretible classes. In the case of conversation data, our hope is that some of the latent classes represent *speech acts* or dialog acts (Searle, 1975). While there is a body of work in supervised speech act classification (Cohen et al., 2004; Bangalore et al., 2006; Surendran and Levow, 2006; Qadir and Riloff, 2011), the variety of conversation domains on the Web motivates the use of unsupervised approaches.

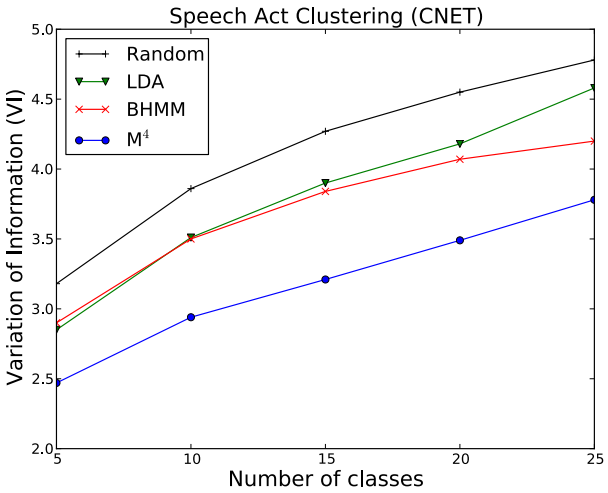


Figure 3: The variation of information between the human-created speech act annotations of the CNET corpus and the latent class assignments by various models.

The CNET corpus is annotated with twelve speech act classes: QUESTION and ANSWER, which are both broken down into multiple sub-classes, as well as RESOLUTION, REPRODUCTION, and OTHER (Kim et al., 2010). We would like to quantitatively measure how closely the latent states induced by our model match these annotations.[7]

We can measure this with *variation of information* (Meila, 2003), which has been used in recent years for unsupervised evaluation, e.g. in part-of-speech clustering (Goldwater and Griffiths, 2007). Given two sets of variable assignments $\mathbf{z}$ and $\mathbf{z}'$, the variation of information is defined as $H(Z|Z') + H(Z'|Z)$. In other words, given one clustering, how much uncertainty do we have about the other? Results are shown in Figure 3: a lower value corresponds to higher similarity.

On the CNET corpus, M^4 outperforms both baselines in all cases by a very significant margin. Qualitatively, we see clusters and transition parameters that make sense. For example, the class with top words $\{i, my, have, computer, am, ?, tried, help\}$ is most likely to begin a thread (with $\lambda = +1.94$) and appears to describe questions or requests for

[7] Some messages have multiple labels. Since messages are not annotated at finer granularities, we handle this by simply duplicating such messages, once per label, and measuring clustering performance on this expanded set of labeled data which now has one label per token.

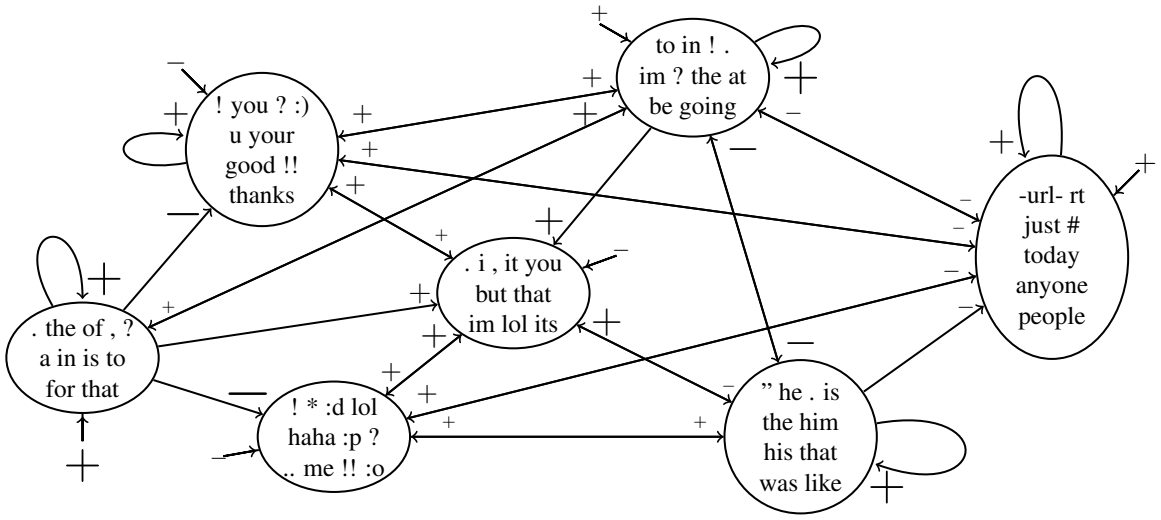


Figure 4: Example output from a model trained on the Twitter corpus with 15 classes (7 shown). Each node corresponds to a class learned by the model, and the most probable words are shown for each class. The symbols $+$ and $-$ on the directed edges denote the sign of the λ associated with transitioning from one class to another, and the size of the symbols is scaled by the magnitude of λ. Non-edge arrows going into a node represent the weight of starting a conversation with that class. Low-magnitude weights are not shown, and some edges are omitted to avoid clutter.

help. The class is not likely to be followed by itself ($\lambda = -0.32$) but is likely to be followed by the class with words $\{you, your, /, com, ., http, windows\}$ (with $\lambda = +1.38$).

The Twitter corpus does not have speech act annotations, so we offer example output in Figure 4. We again see patterns that we might expect to find in social media conversations, and some classes appear to correspond to speech acts such a declarations, personal questions, and replies. For example, the class in the center of the figure has words like *you* and *but* which suggests it is used in reply to other messages, and indeed we see that it has a positive weight of following almost every class, but a negative weight for actually starting a thread. Conversely, the class containing URLs (which corresponds to the act of sharing news or media) is likely to begin a thread, but is not likely to follow other classes except itself.

How well unsupervised models can truly capture speech acts is an open question. Much as LDA "topics" do not always correspond to what humans would judge to be semantic classes (Chang et al., 2009), the conversation classes inferred by unsupervised sequence models are similarly unlikely to be a perfect fit to human-assigned classes. Nevertheless, these results suggest M^4 is a step forward. Our model provides a framework for defining inter-message transitions as functions of multiple classes, which will be a desirable property for many corpora.

6 Conclusion

We have presented mixed membership Markov models (M^4), which extend the simple HMM approach to discourse modeling by positing class assignments at the level of individual tokens. This allows blocks of text to belong to potentially multiple classes, a property that relates M^4 to topic models. This type of model can be viewed as an HMM with an expanded state space, but because the transition probabilities are a function of a small number of parameters, the output remains human-interpretible.

M^4 can be taken as a general family of models and can be readily extended. In this work, we focused on introducing a model of *inter*-message structure, but certainly more sophisticated models of *intra*-message structure beyond unigram language models could be incorporated into M^4. Standard topic model extensions such as n-gram models (Wallach, 2006) can straightforwardly be applied here, and indeed we already applied such an extension by incorporating background distributions in §5.3. For conversational data, it could make sense to segment

messages (e.g. into sentences) and constraint each segment to belong to one class or speech act; modifications along these lines have been applied to topic models as well (Gruber et al., 2007). While we have focused on conversation modeling, M^4 is a general probabilistic model that could be applied to other discourse applications, for example modeling sentences or paragraphs in articles rather than messages in conversations; it could also be applied to data beyond text.

Compared to a Bayesian block HMM, M^4 performs much better at a variety of tasks. A drawback is that the time complexity of inference as presented here is quadratic in the number of classes rather than linear. Improving this may be the subject of future research. Another potential avenue of future work is to model transitions such that a Dirichlet prior for the class distribution of a block, rather than the class distribution itself, depends on the previous class assignments. This would yield a model that more closely resembles LDA, but with topic priors that encode sequence information.

Acknowledgements

Thanks to Matt Gormley, Mark Dredze, Jason Eisner, the members of my lab and the anonymous reviewers for helpful feedback and discussions. This material is based upon work supported by a National Science Foundation Graduate Research Fellowship under Grant No. DGE-0707427 and a Dean's Fellowship from the Johns Hopkins University Whiting School of Engineering.

References

Srinivas Bangalore, Giuseppe Di Fabbrizio, and Amanda Stent. 2006. Learning the structure of task-driven human-human dialogs. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, ACL-44, pages 201–208.

Regina Barzilay and Lillian Lee. 2004. Catching the drift: Probabilistic content models, with applications to generation and summarization. In *HLT-NAACL 2004: Main Proceedings*, pages 113–120, Boston, Massachusetts, USA, May 2 - May 7. Association for Computational Linguistics.

M. J. Beal, Z. Ghahramani, and C. E. Rasmussen. 1997. Factorial hidden markov models. In *Machine Learning*, volume 29, pages 29–245.

D. Blei and J. Lafferty. 2007. A correlated topic model of science. *Annals of Applied Statistics*, 1(1):17–35.

David M. Blei and Jon D. Mcauliffe. 2007. Supervised topic models. In *Advances in Neural Information Processing Systems 21*.

David Blei, Andrew Ng, and Michael Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3.

Jonathan Chang, Jordan Boyd-Graber, Sean Gerrish, Chong Wang, and David Blei. 2009. Reading Tea Leaves: How Humans Interpret Topic Models. In *Neural Information Processing Systems (NIPS)*.

Chaitanya Chemudugunta, Padhraic Smyth, and Mark Steyvers. 2006. Modeling general and specific aspects of documents with a probabilistic topic model. In *NIPS*, pages 241–248.

Harr Chen, S. R. K. Branavan, Regina Barzilay, and David R. Karger. 2009. Global models of document structure using latent permutations. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 371–379.

William W. Cohen, Vitor R. Carvalho, and Tom M. Mitchell. 2004. Learning to classify email into "speech acts". In *Proceedings of EMNLP 2004*, pages 309–316, Barcelona, Spain, July. Association for Computational Linguistics.

A. P. Dempster, N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38.

Christopher P. Diehl, Galileo Namata, and Lise Getoor. 2007. Relationship identication for social network discovery. In *AAAI'07*.

Lan Du, Wray Buntine, and Huidong Jin. 2010. Sequential latent dirichlet allocation: Discover underlying topic structures within a document. *2010 IEEE International Conference on Data Mining*, pages 148–157.

Jonathan L. Elsas and Jaime Carbonell. 2009. It pays to be picky: An evaluation of thread retrieval in online forums. In *32nd Annual International ACM SIGIR Conference on Research and Development on Information Retrieval(SIGIR 2009)*.

Jenny Rose Finkel, Trond Grenager, and Christopher D. Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *ACL*.

Sharon Goldwater and Tom Griffiths. 2007. A fully bayesian approach to unsupervised part-of-speech tagging. In *Proceedings of the 45th Annual Meeting of*

the Association of Computational Linguistics, pages 744–751, Prague, Czech Republic, June. Association for Computational Linguistics.

Tom Griffiths and Mark Steyvers. 2004. Finding scientific topics. In *Proceedings of the National Academy of Sciences of the United States of America*.

Amit Gruber, Michal Rosen-Zvi, and Yair Weiss. 2007. Hidden topic markov models. In *Artificial Intelligence and Statistics (AISTATS)*, San Juan, Puerto Rico.

Timothy Hospedales, Shaogang Gong, and Tao Xiang. 2009. A markov clustering topic model for mining behaviour in video. In *International Conference on Computer Vision (ICCV)*.

Shafiq R. Joty, Giuseppe Carenini, and Chin-Yew Lin. 2011. Unsupervised modeling of dialog acts in asynchronous conversations. In *IJCAI*, pages 1807–1813.

Su Nam Kim, Li Wang, and Timothy Baldwin. 2010. Tagging and linking web forum posts. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, CoNLL '10, pages 192–202.

Marina Meila. 2003. Comparing clusterings by the variation of information. *Learning Theory and Kernel Machines*, pages 173–187.

D. Mimno and A. McCallum. 2008. Topic models conditioned on arbitrary features with dirichlet-multinomial regression. In *UAI*.

Tom Minka. 2003. Estimating a dirichlet distribution.

Ashequl Qadir and Ellen Riloff. 2011. Classifying sentences as speech acts in message board posts. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 748–758, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.

Alan Ritter, Colin Cherry, and Bill Dolan. 2010. Unsupervised modeling of twitter conversations. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 172–180.

Carolyn Penstein Rosé, Barbara Di Eugenio, Lori S. Levin, and Carol Van Ess-Dykema. 1995. Discourse processing of dialogues with multiple threads. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, ACL '95, pages 31–38.

John Searle, 1975. *A taxonomy of illocutionary acts*. University of Minnesota Press, Minneapolis.

Jangwon Seo, W. Bruce Croft, and David A. Smith. 2009. Online community search using thread structure. In *ACM Conference on Information and Knowledge Management (CIKM 2009)*, pages 1907–1910.

Andreas Stolcke, Noah Coccaro, Rebecca Bates, Paul Taylor, Carol Van Ess-Dykema, Klaus Ries, Elizabeth Shriberg, Daniel Jurafsky, Rachel Martin, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26(3):339–373, September.

Dinoj Surendran and Gina-Anne Levow. 2006. Dialog act tagging with support vector machines and hidden markov models. In *Interspeech*.

Hanna M. Wallach, David Mimno, and Andrew McCallum. 2009. Rethinking LDA: Why priors matter. In *NIPS*.

H.M. Wallach. 2006. Topic modeling: beyond bag-of-words. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 977–984.

Hongning Wang, Chi Wang, ChengXiang Zhai, and Jiawei Han. 2011a. Learning online discussion structures by conditional random fields. In *34th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'11)*, pages 435–444.

Hongning Wang, Duo Zhang, and ChengXiang Zhai. 2011b. Structural topic model for latent topical structure analysis. In *ACL*, pages 1526–1535. The Association for Computer Linguistics.

Li Wang, Marco Lui, Su Nam Kim, Joakim Nivre, and Timothy Baldwin. 2011c. Predicting thread discourse structure over technical web forums. In *Proceedings of EMNLP 2011*, pages 13–25.

Gu Xu, Hang Li, and Wei-Ying Ma. 2008. Fora: Leveraging the power of internet communities for question answering. In *1st International Workshop on Question Answering on the Web (QAWeb08)*.

Jen-Yuan Yeh and Aaron Harnly. 2006. Email thread reassembly using similarity matching. In *Proceedings of the 3rd Conference on Email and Anti-Spam (CEAS 2006)*, pages 64–71.