A Comparison of Latent Variable Models For Conversation Analysis

Sourish Chaudhuri and Bhiksha Raj
Language Technologies Institute,
School of Computer Science,
Carnegie Mellon University,
Pittsburgh, PA - 15213.
{sourishc, bhiksha} @ cs.cmu.edu

Abstract

With the evolution of online communication methods, conversations are increasingly handled via email, internet forums and other such methods. In this paper, we attempt to model lexical information in a context sensitive manner, encoding our belief that the use of language depends on the participants in the conversation. We model the discourse as a combination of the speaker, the addressee and other participants in the conversation as well as a context specific language model. In order to do this, we introduce a novel method based on an HMM with an exponential state space to capture speaker-addressee context. We also study the performance of topic modeling frameworks in conversational settings. We evaluate the models on the tasks of identifying the set of people present in any conversation, as well as identifying the speaker for every utterance in the conversation, and they show significant improvement over the baseline models.

1 Introduction

In this paper, we experiment with different methods of automatically analyzing discourse. We present and validate hypotheses on how conversations can be better analyzed using information about the speakers, as well as other participants in the conversation. We present a novel method of modeling discourse using an exponential state Hidden Markov Model where states are based on speakers and addressees. We also cast the problem into the popular topic modeling frameworks, and compare the various approaches.

Consider a small group of people that a person knows well. Given a transcript of a discussion on a topic of mutual interest, that person would likely be able to identify who is likely to have said what, based on his knowledge of the speakers and their inclinations on various topics. We would like to be able to encode similar intelligence into a system that could automatically learn about speakers based on transcripts of prior conversations, and use that information to analyze new conversations.

The scenario we consider in this work is as follows: we have a known set of characters, any subset of whom could be present in a conversation. Given the transcript of a conversation only, without speaker annotations, we would like to : 1. Predict the *set* of participants in the conversation from the characteristics of the entire conversation, and 2. Identify the *individual* speakers at each conversation turn.

In order to do this, we model each utterance in a conversation as dependent on the speaker, the addressee and the other people present. As we shall describe, our models encode the belief that people speak/behave differently depending on other participants in the conversation. This has a two-fold benefit: first, it can help us discover social (or even, professional) relationship structures; second, it can help us understand how to respond to different people, and incorporate that information into automated conversational agents which can then behave in a more context sensitive manner. The ability to automatically model discourse as context specific in this manner is also useful for other tasks such as directed advertising and duplicity detection.

In Section 2, we describe relevant related work.

Proceedings of the SIGDIAL 2011: the 12th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 30–38,
Portland, Oregon, June 17-18, 2011.

Section 3 describes the dataset for our experiments, Section 4 describes the problem, our use of topic models, and the novel HMM based method, while Section 5 summarizes the results and we conclude in Section 6.

2 Related Work

The task of automatically segmenting speech and then identifying speakers from audio (Reynolds and Torres-Carrasquillo, 2005) is referred to as diarization and has been well-studied (Tranter and Reynolds, 2006). More recently, approaches have been developed to fuse information from both the audio and video modalities (Noulas et al., 2011) to improve diarization systems when video information is available. In this paper, we attempt to understand just how much information is available in the text alone. Systems that can work with text only can be used to improve audio-based systems which can provide speech recognition output to a text-based system. They can also be used to work with closed caption streams, or on human-generated transcriptions of meeting recordings.

Research on identifying speakers from text or lexical information is limited in comparison to work with audio data. However, efforts have been made to use discourse level information to automatically identify speakers to calibrate idiolectal differences between speakers (Doddington, 2001). (Canseco et al., 2005,) investigated the use of lexical features to automatically diarize (but not actually identify) transcripts to determine if a current speaker continued or a previous speaker spoke or the next speaker spoke. Lei and Mirghafori (2 007) attempted to incorporate idiolect based speaker information by using word conditioning of phone N-grams to recognize speakers in dialogs with 2 speakers.

In our work, the models we use to identify speakers are powerful enough to predict the addressee as well. In this context, we note that several attempts have been made recently to automatically identify addressees in dialog settings. These approaches have used information about the context and content of the utterance, using dialog acts and information about the speaker's gaze to aid classifier performance (Jovanovic et al., 2006). Den Akker and Traum (2009) proposed rule-based methods for addressee classification. Unlike in these works, we attempt to jointly model both the speaker and the addressee as one of our proposed approaches. This is similar to the approach employed by (Otsuka et al., 2005,), who proposed a Dynamic Bayesian Network model to understand multiparty conversation structure using non-verbal cues only– eye gaze, facial expression, gesticulations and posture.

3 Data

The data for our experiments consists of fan-sourced transcripts of the episodes of the sitcom F.R.I.E.N.D.S. The structure of the data is as follows: we have a set of conversations as training data. Each conversation contains a sequence of turns, with each turn annotated with its speaker. We do not have any information about the addressee from the dataset. We do, however, have implicit information of the set of speakers within a conversation segment (we make the assumption here that if a character doesn't speak in a segment, he is not present). Annotator notes appear periodically to indicate that the scene changed or that new characters entered the scene or that some characters left the scene. We treat these annotator notes as conversation boundaries and the segment of turns between two such boundaries constitutes one conversation instance.

The set of characters used for our experiments is finite. The 6 primary characters in the sitcom (Chandler, Joey, Monica, Phoebe, Rachel and Ross) are retained. In addition to these 6 primary characters, there are a number of supporting characters who appear occasionally. We use *Other* to denote all other characters, as the amount of data for a number of the supporting characters is quite small and would not result in learning useful patterns regarding their behavior. As a result, we treat all of these characters as one character that can be thought of as a universal supporting character. Hence, we have a total of 7 possible characters. Any subset of these 7 characters could be part of a conversation. Below is an example of a pair of conversations from our dataset:

[EVENT]
Paul: thank you! thank you so much!
Monica: stop!
Paul: no, i'm telling you last night was like umm, all my birthdays, both graduations, plus the barn

raising scene in witness.

Monica: we'll talk later.

Paul: yeah. thank you.

[EVENT]

Joey: that wasn't a real date?! what the hell do you do on a real date?

Monica: shut up, and put my table back.

All: okayyy!

[EVENT]

The *event* markers are tags inserted at preprocessing time, to denote transcriber annotations such as characters entering or leaving scenes. The sequence of turns between two *event* markers are treated as one conversation. Also, note the character Paul in the first conversation in the example above – when training the system, the content of Paul's utterances are used to train the model for *Other*, since Paul is not one of the primary characters that we track. At test time, the input looks similar to the above, except that the turns are not annotated by speaker.

The transcripts used in our experiments are segmented by speaker turns, so that consecutive turns are uttered by different speakers. The entire set of 230 episodes was split randomly into training, development and test splits. Sequential information for the individual conversations were not used. Each episode was further divided into conversations based on the scene boundaries denoted by the transcribers. For training, overall, we used 195 episodes from F.R.I.E.N.D.S, with a total of 9,171 conversations and a total 52,516 turns. The average length in number of turns for each conversation was 5.73. The test set consisted of a total of 20 episodes with 855 conversations and 4,981 turns. The average length of a conversation in the test set was 5.83. The remaining 15 episodes were used as development data to tune hyperparameters – this set consisted of 529 conversations and 2,984 turns in total. The distribution of the number of utterances by speakers across the training, test and development set are shown in Figure 1. As one can observe, the distribution is not particularly skewed for any of the speakers across the splits of the dataset.

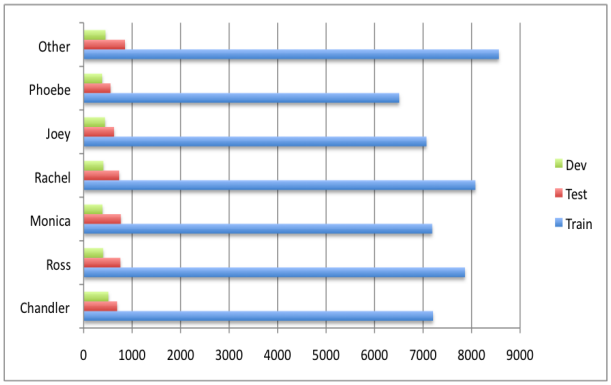

Figure 1: Distribution of #utterances for each speaker in the dataset.

4 Conversation Models

Previous work in analyzing participants in a conversation have used meeting data, with a fixed number of participants. In our task, the total number of possible participants is finite, but we do not have information on how many of them are present at any particular instant. Thus, our model first attempts to detect the participants in a segment of conversation, and then attempts to attribute speaker turns to individuals.

Our model for discourse structure is based on two premises. First, we believe that what a person says will depend on who he or she is speaking to. Intuitively, consider a person trying to make the same point to his boss and (at a different time and place) to his friend. It is likely that he will be more formal with his boss than his friend. Second, if the speaker addresses someone specifically in a group of people, knowing who he addressed would likely help us predict better who would speak next. We assume that the first hypothesis above also holds for groups of people in conversations, where the topics and their distribution in discussions (and words that affect the tone of the discussion) depend on the participants.

As described earlier, we evaluate our models on two tasks. First, we would like to identify the set of characters present in any conversation. Given segments of conversation, we attempt to understand the distribution of topics for specific subsets of characters present in that segment. To do this, we cast this problem into a topic modeling framework – we experiment with the Author-Topic model (Rosen-

Zvi et al., 2004), described in Section 4.1, for this task. We use the Author Topic Model to link the co-occurrence information of characters with the words in the conversation.

Second, we attempt to attribute speakers to utterances, described in Section 4.2. We introduce a novel approach using an HMM with an exponential state space to model speakers and addressees, described in Section 4.2.1. We also use the Author Topic Model and the Author-Recipient Topic Model (McCallum et al., 2007), described in Section 4.2.2 for this task. The key difference between the HMM-based model and the topic model based approaches is that the former explicitly takes sequence information into account.

4.1 Identifying Character Subset Presence

The premise behind attempting to model subsets of characters is that the nature of the conversation depends on the group of people participating. For instance, it seems intuitively likely that the content of a conversation between two friends would be different if they were the only ones present than it would be if their families were also present. To extend this hypothesis to a general scenario, the content of each speaker's turn depends not only on the speaker, but also on the person being spoken to as well as the other people present. To model this, we require a model that captures the distribution of the text for entire conversation, for each possible subset of characters. In this section, we describe the training of a generic model for conversations, and use it to produce features for a discriminative classifier.

Let there be N characters who could participate in a conversation. We assume a general scenario, where any subset of these characters may be present. Thus, there are $2^N - 1$ character subsets that are possible. We can model this as a multi-class classification problem (we will refer to this as *subset modeling*, henceforth).

The generative model for this task is as follows: Each conversation segment is associated with a set of utterances, and a set of characters. For each such set of characters, we associate a distribution over topics. For each word that is present in the segment, we select a topic from the subset-specific topic distribution, and then we select the word from that topic. Figure 2 shows the graphical model for this in plate notation.

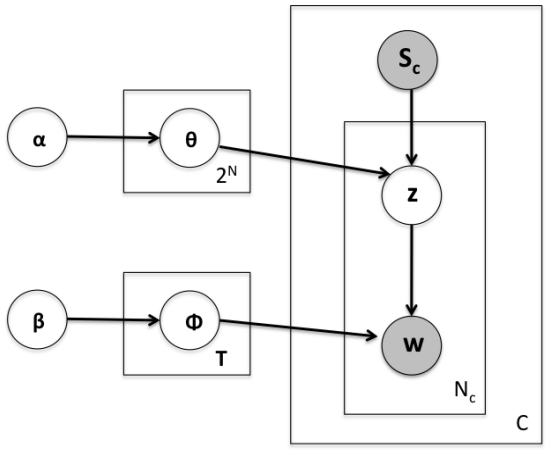

Figure 2: Graphical representation of the subset model in plate notation

In the plate notation, the observed variables are shaded and the latent variables are unshaded. Plates encapsulate a set of variables which are repeatedly sampled a fixed number of times, and the number at the bottom right indicates this fixed number.

S_c represents a subset of the characters who were present in the conversation segment. We have C such conversations, and each conversation contains N_c words. z represents the latent topic variable, and θ represents the multinomial topic distribution for each subset of characters (there are 2^N such subsets). The multinomial distribution of topics has a prior distribution characterized by α. Similarly, every topic (there are a set of T topics) has a multinomial distribution ϕ over the words in the vocabulary, and ϕ has a prior distribution characterized by β.

For every conversation in the training corpus, the set of characters present is known. The content of the conversation is treated as a bag of words. From the topic distribution for the subset of characters present, we sample a topic. Based on the word distributions for this topic, we sample a word. This process is repeated N_c times corresponding to the number of words in the conversation. The entire process of generating a conversation is repeated C times, corresponding to the number of conversations in the training corpus.

Depending on the value of N, the number of pos-

sible classes may be very high. Training a large number of models may lead to a data scarcity, especially given the high dimensionality of language data. We therefore slightly modify the model, so that instead of topic distributions for each possible subset, we have a topic distribution for each character, and the distribution of topics in the conversation is a mixture of the topic distributions for each character. This leads us to a graphical model that has been well-studied in the past – the Author-Topic model (ATM, henceforth) and is shown in Figure 3.

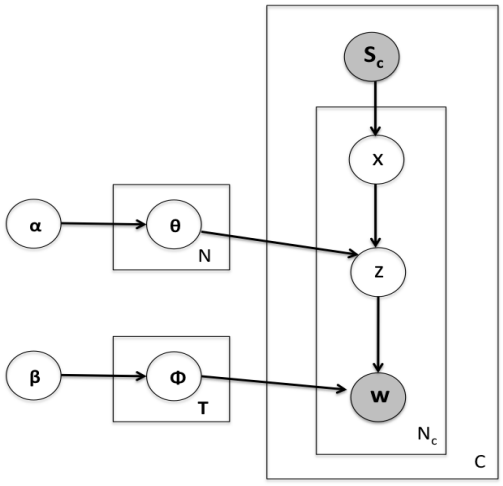

Figure 3: Graphical representation of the simplified subset model in plate notation

Thus, given the set of characters present, we sample one of them (x) from a uniform distribution. Then we generate a topic by sampling from the distribution of topics for that speaker. The rest of the process remains the same.We use this model to help us predict which subset of characters was present in a given conversation.

We learn speaker-specific topic distributions using the ATM. In order to predict characters present in a test conversation, we train binary SVM (Shawe-Taylor and Cristianini, 2000) classifiers for each speaker in the following manner: we compute the distribution of the speaker-specific topics in each conversation, and use these as the features of the data point. If the speaker was present in the conversation, the data point corresponding to the conversation has a class label of +1, else -1. A linear SVM classifier is trained over the data. At test time, we compute the distribution of the speaker's topics in the conversation, and use the SVM to predict if the speaker was present or not.

4.2 Identifying Speakers From Utterances

In this section, we describe our approach to identifying speakers from the text of the utterance. The ATM (as described above) treats all the participants in the conversation as being potential contributors to each turn. However, we can also use the ATM to predict speakers directly. In this case, we will use each turn as analogous to a document. Each such *document* has only one *author* and the author topic model can be used to learn models for each author. The plate notation for this would look very similar to the one in Figure 2, except that instead of a subset of characters being observed, only one would be observed, and the number of possible topic distributions would be equal to the number of characters.

The ATM for this task does not take any context information into account. In the following subsection, we introduce a novel HMM based approach that seeks to leverage information from the sequence of turns.

4.2.1 Exponential State Hidden Markov Model

In this model, we assign a state to each speaker-addressee combination possible. If our data consists of N characters, only one of the N characters will be speaking at any given point. He/She may be speaking to any combination of the remaining $N-1$ characters. Thus, the number of states in this model is $N \times 2^{(N-1)}$. Note that the addressee is not observed directly from the data.

The sequence of turns in a conversation is modeled by a Hidden Markov Model (Rabiner, 1989). At each time instant, the speaker corresponding to the state speaks a turn, which is the observed emission, before transitioning to another state at the next time instant. The state at the next time instant is constrained to have a different speaker.

The model is trained using the standard Baum-Welch training. The emission probabilities are captured by a trigram language model, trained using the SRILM toolkit (Stolcke, 2002). The parameters of the model are initialized as follows: for emission probabilities, we take all the utterances by a speaker and distributing them uniformly among the

states that have that speaker, since we do not have direct information about the addressees. For transition probabilities, we initialize with a bias instead of uniformly. Given a conversation, for a state with speaker A and set of addressees (R, say – Note that R may have multiple characters), we give equal probabilities of transitioning to all states that have one of the characters in R as the speaker. Now, we pick the set of speakers (call it M) that uttered the next three turns (essentially, we look ahead in the data stream to see who the next 3 speakers are while training). We add a bias to every state with A as the speaker, and every possible combination of the speakers in M, to encode the hypothesis that the addressee would be likely to speak pretty soon, if not directly after.

The large state space in this model makes computation extremely expensive. However, an examination of the posterior probabilities show that a number of states are rarely, or never, entered. We prune away such states after every 5 iterations in the following manner – we use the current parameters of the model after each iteration to identify the speakers of each turn on the development set. Decoding of a sequence of turns at test time is done using the Viterbi algorithm. However, instead of using the best path only, we keep track of the top 10 best paths. Thus, after an iteration of training, we test on the development data, and obtain 10 possible sequences of speakers for each conversation. Over 5 iterations, we have the 50 best paths for each conversation. We then compute the average number of states entered in all the decoded paths obtained. If the average number of times a state was entered is μ, then any state that was entered less than $k \times \mu$ times ($k = 0.02$, for our experiments), according to the posterior probabilities was pruned out. In order to set the value of k, the development set was split into 2 halves, with one half being used to compute the average number of times a state is entered across the 10 best decodes for data in that half. For different values of k, accuracy of speaker identification on the 1-best decode was computed on the other half of the development set, for values of k from 0.005 to 0.1.

The optimal state sequence at test time also contains information about the addressee. For the tasks we evaluate, this information is not directly used. However, in other applications, such as those involving automated agents, this information could be valuable in triggering the agent.

4.2.2 Author-Recipient Topic Model

The Author Recipient Topic Model (McCallum et al., 2007) (ARTM, henceforth) was used for discovering topics and roles in social networks. It is built over the Author-Topic Model discussed previously, with the exception that messages are conditioned on the sender as well as the receivers. The graphical model in plate notation is shown in Figure 4.

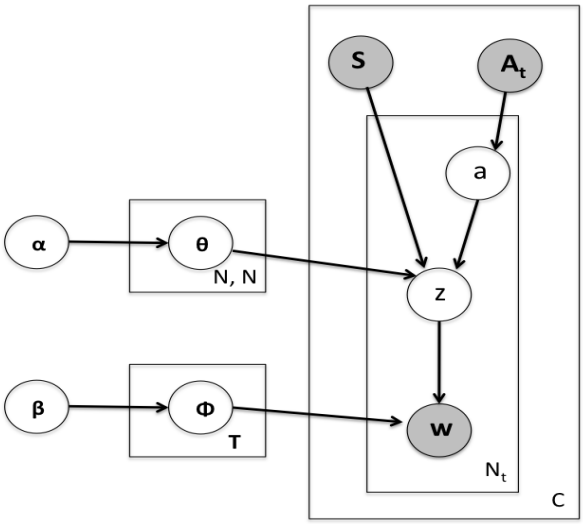

Figure 4: Graphical representation of the Author-Recipient Topic model in plate notation

Here, we model each turn as having a set of N_t words. Each turn has one speaker S, and a set of addressees A_t. The generative model works as follows: For each word in a turn, sample an addressee a from the set of addressees. Topic distributions are now conditioned over speaker-addressee pairs, instead of only the speaker as we saw in the ATM. A topic is now sampled from the speaker-addressee specific topic distribution. A word is now sampled from this topic using the topic specific word distributions. The parameters α, β, and z have the same meaning as in the ATM described earlier.

Note that the set of addressees in our setting is not explicitly observed. We know the participants in the conversation at training time, and we know the speaker, but we do not know who was addressed. Since we do not have information to make a better choice of addressee, we model the entire set of par-

ticipants without the speaker as the set of addressees, in this model.

For the task of identifying the speaker who uttered the turn, we employ an approach, similar to the one used for ATM. We train speaker-addressee-specific models. The feature set for this task includes features not only from the turn itself, but also from the context. Thus, we have the distribution of the topics in the turn for *every* speaker-addressee pair with the right speaker, the speakers of the previous two turns, and the distribution of topics of the speaker of the current turn over the previous two turns. (Thus, while the model does not explicitly model sequence, as an HMM does, it utilizes context information in its feature space.) Using these features, we train a linear SVM to predict whether or not the speaker uttered the turn. In this case, we could potentially have multiple speakers (or none of them) predicted to have uttered the same turn. In that case, we choose the speaker with the maximum distance from the margin.

4.3 Baseline Models

In this section, we set up simple baseline models to evaluate our performance against. We describe how we set up a random baseline, a Naive Bayes baseline and an HMM baseline model.

4.3.1 Random Baseline

For the task of identifying the set of characters present in a conversation, the random baseline would work as follows: it knows that the number of characters present in any conversation lies between 1 and *N* (N = 7, in this case). (Note that monologues, with only 1 person being present, are possible. Typically, in our data, they happen at the beginning or end of scenes.) Thus, it randomly decides if each of these characters are present or not in any given conversation.

Suppose that the total number of characters are *n* and *r* of them are actually present in the conversation. Let us say the random guess system predicts *t* of the characters to be present. If we use the uniform distribution for picking t, then $P(t) = \frac{1}{7}, \forall t \in [1, 2, ..., 7]$, in this case. For any given t, the probability that we get k correct is given by:

$$P(k|t) = \frac{\binom{r}{k} \times \binom{n-r}{t-k}}{\binom{n}{t}} \quad (1)$$

To compute the probability of getting k right, we marginalize out the number of characters guessed to be present, t:

$$P(k) = \sum_t P(k,t) = \sum_t P(k|t).P(t) \quad (2)$$

Now we can compute the probability of getting k correct by randomly guessing, for all k from 0 to r. Using these, we can compute the expected number of correct guesses, which turns out to be $0.571.r$ for an average recall would be 57.1%.

For the task of identifying the characters, every turn could have been uttered by one of the n characters ($n = 7$, for our case). Thus, the average accuracy at identifying turns would be $\frac{1}{7}$ or 14.29%.

4.3.2 Naive Bayes Classifier

For the task of predicting the subset of speakers, we set up a Naive Bayes using words as features. We build up a term-document matrix, with each conversation treated as a document. For each character, we train a binary classifier using the training data- conversations where the character was present were marked as a positive instance for that character, and ones where he was not present were marked as negative instances. We experimented both with using priors based on the empirical distribution in the training data and with using uniform prior (i.e. $P(character) = 0.5$). Given a test conversation, we use individual classifiers for each of the characters to determine whether he/she was present or not.

For the task of identifying speakers, given an utterance, the Naive Bayes classifier is set up as follows: Again, we create term-document matrices for each of the speakers, where a document is a turn uttered by the speaker. Turns uttered by that speaker are positive instances and those uttered by someone else are negative instances. For each speaker, we compute the Naive Bayes probability ratio (odds) of him uttering the turn and not uttering the turn, in order to decide. If multiple speakers are classified as having uttered the turn, or no speaker is classified as having uttered the turn, the speaker with the best odds of having uttered the turn is selected.

System	Precision	Recall
Author Topic Model	63.22%	74.71%
NB	52.33%	44.19%
NB-prior	68.31%	36.25%
Random Baseline	28.05%	57.1%

Table 1: Results for predicting subset of characters present

System	Accuracy
ESHMM	27.13%
Speaker-LM HMM	25.04%
ARTM	23.64%
Author Topic Model	26.2%
NB	23.41%
NB-prior	21.39%
Random Baseline	14.29%

Table 2: Results for predicting speakers of utterances

4.3.3 Single Speaker HMM

This model is only used to attribute speakers to turns. Section 4.2.1 described an HMM model that captures speaker-addressee information. In the single- speaker HMM, we have a state for each speaker. Emission probabilities are given by a trigram language model that is trained on the speaker's utterances in the training data. The transition probabilities are initialized as per the empirical transitions between speakers in the data. This model does not capture any kind of addressee information.

5 Results

In this section, we present results of our experiments with the models we described earlier, on the two tasks, identifying the set of speakers in any given conversation and identifying individual speakers who uttered each turn in a conversation.

For the task of identifying the set of speakers in any given conversation, we evaluate performance using precision and recall, which are defined as follows: If the conversation actually contained r characters, the system predicted that it contained t characters, and got k right, then:

$$Precision = \frac{k}{t}; Recall = \frac{k}{r} \quad (3)$$

The results are summarized in Table 1. In the table, NB-prior indicates that the prior for the binary classifier was determined based on the number of conversations each character appeared in, while NB indicates that the prior was uniform (i.e., for each character, $P(present) = P(absent) = 0.5$). We find that the results obtained using the author-topic model are significantly better than each of the other three models.

On average, the number of speakers in each conversation in the test data was 2.44 (the corresponding number in the training data appears to be somewhat higher at 2.65). Our attempts to restrict the set of characters in a real setting plays a significant role here as we shall discuss later.

The Naive Bayes classifier with empirical priors on average predicted that there were 1.3 characters present per conversation, while the version with uniform priors predicted 2.2 characters to be present per conversation on average. The author-topic model, on average, over-estimated the number of characters at 2.86 characters per conversation.

For the task of predicting the speaker, given an utterance, we have two kinds of Hidden Markov Models, the Exponential State HMM (ESHMM) and and HMM with emission probabilities based on individual speaker language models (Speaker LM HMM). We also have the topic model based systems- the ARTM and the ATM. Finally, we have the baseline models- the Naive Bayes with empirical priors and with uniform priors, and the random baseline. Table 2 summarizes their performance. In this case, we only report accuracy. Since each turn has only one speaker, we can constrain each of the models to produce one speaker, in order to calculate the accuracy.

The HMM and topic based models all incorporate sequence information in some form. In the case of the HMM based models, state transitions are conditioned on the previous speaker. In the case of the topic model based systems, the feature vectors contain context, although the task is modeled as a discriminative classification task. The ESHMM model worked the best on this dataset. With the exception of the ATM and the speaker LM HMM ($p < 0.10$), the improvements obtained by using the ESHMM over all other models were statistically significant ($p < 0.05$). Surprisingly, the single speaker LM

HMM and the ATM both outperform the ARTM on this task. One of the reasons for this could be that the ARTM does not suitably capture what we hoped it would, perhaps because of the fact that the recipients (addressees) are not observed.

6 Conclusion

In this paper, we presented a set of latent variable model based approaches to analyzing conversation structure using the text transcript of the conversations only. The initial set of experiments show promising improvements over simple baseline methods, though the overall results leave considerable room for improvement. Conversations are a dynamic process, with the content varying significantly with time, and the use of formulations such as dynamic topic modeling (Blei and Lafferty, 2006) may help.

We believe that the concept of modeling speakers and addressees would be a powerful one in modeling conversation structure and useful in applications such as those involving automated agents, or in understanding discourse on discussion forums, as well as understanding development of authority in such forums. The state sequences predicted by the ESHMM implicitly predict addressees for each turn. This is not directly used in our tasks, but could be useful for automated agents, in understanding appropriate moments to take its turn.

The dataset used in this case introduced some noise. We decided to subsume everyone aside from the 6 main characters under the moniker *other*, in order to keep the state space manageable. In reality, it was a collection of a few dozen characters, some of whom appeared intermittently through the episodes. As a result, the emission model for this state was not a stable one. The system rarely predicted this class, and had very low accuracy when it did.

Further, development of datasets with annotations specifying the addressees explicitly would probably accelerate development of methods that work well in such settings.

References

Andrew McCallum, Xuerui Wang and Andres Corrada-Emmanuel. 2007. Topic and Role Discovery in Social Networks with Experiments on Enron and Academic Email. In *Journal of Artificial Intelligence Research.*.

Andreas Stolcke. 2002. SRILM an Extensible Language Modeling Toolkit. In *ICSLP*.

D. A. Reynolds and P. Torres-Carrasquillo. 2005. Approaches and applications of audio diarization. In *Proc. of ICASSP*.

David M. Blei and John D. Lafferty. 2006. Dynamic Topic Models. In *Proceedings of ICML.*

George Doddington. 2001. Speaker Recognition based on Idiolectal Differences between Speakers. In *Eurospeech.*.

S.E. Tranter and D.A. Reynolds. 2006. An overview of automatic speaker diarization systems. In *IEEE Transactions on Audio, Speech and Language Processing.*.

Athanasios Noulas, Gwenn Englebienne, Ben J.A. Krse 2011. Multimodal Speaker Diarization. In *IEEE Transactions on Pattern Analysis and Machine Intelligence.*.

Howard Lei and Nikki Mirghafori. 2007. *Word-Conditioned Phone n-grams For Speaker Recognition*. In *Proceedings of ICASSP*

John Shawe Taylor and Nello Cristianini. 2000. *Support Vector Machines and other Kernel Based Learning Methods*. Cambridge University Press..

Kazuhiro Otsuka, Yoshinao Takemae and Junji Yamato. 2005. A probabilistic inference of multiparty-conversation structure based on Markov-switching models of gaze patterns, head directions, and utterances. In *Proceedings of the 7th international conference on Multimodal interfaces.*

L.R. Rabiner. 1989. A tutorial on Hidden Markov Models and selected applications in speech recognition. In *Proceedings of IEEE.*

Leonardo Canseco, Lori Lamel and Jean-Luc Gauvain 2005. A Comparative Study using Manual and Automatic Transcriptions for Diarization. In *Proceedings of ASRU*.

Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers and Padhraic Smyth. 2004. The Author-Topic Model for Authors and Documents. In *20th Conference on Uncertainty in Artificial Intelligence.*

Natasa Jovanovic, Rieks op den Akker and Anton Nijholt. 2006. Addressee Identification in Face-to-Face Meetings In *Proc. of EACL.*

Rieks op den Akker and David Traum. 2009. A Comparison of Addressee Detection Methods for Multiparty Conversations. In *Proc. of Diaholmia 2009.*

Thomas Hofmann. 1999. Probabilistic latent semantic analysis. In *Proc of UAI.*.

Xavier Anguera, Chuck Wooters and Javier Hernando 2005. Speaker Diarization for Multi-Party Meetings using Acoustic Fusion In *IEEE Workshop on Automatic Speech Recognition and Understanding.*.