Speech Act Modeling of Written Asynchronous Conversations with Task-Specific Embeddings and Conditional Structured Models

Shafiq Joty and **Enamul Hoque**
ALT Research Group
Qatar Computing Research Institute — HBKU, Qatar Foundation
{sjoty, mprince}@qf.org.qa

Abstract

This paper addresses the problem of speech act recognition in written asynchronous conversations (e.g., fora, emails). We propose a class of conditional structured models defined over arbitrary graph structures to capture the conversational dependencies between sentences. Our models use sentence representations encoded by a long short term memory (LSTM) recurrent neural model. Empirical evaluation shows the effectiveness of our approach over existing ones: (*i*) LSTMs provide better task-specific representations, and (*ii*) the global joint model improves over local models.

1 Introduction

Asynchronous conversations, where participants communicate with each other at different times (e.g., fora, emails), have become very common for discussing events, issues, queries and life experiences. In doing so, participants interact with each other in complex ways, performing certain communicative acts like asking questions, requesting information or suggesting something. These are called **speech acts** (Austin, 1962).

For example, consider the excerpt of a forum conversation from our corpus in Figure 1. The participant who posted the first comment C_1, describes his situation by the first two sentences and then asks a *question* in the third sentence. Other participants respond to the query by *suggesting* something or *asking* for clarification. In this process, the participants get into a conversation by taking turns, each of which consists of one or more speech acts. The two-part structures across posts like 'question-answer' and 'request-grant' are called **adjacency pairs** (Schegloff, 1968).

C_1: My son wish to do his bachelor degree in Mechanical Engineering in an affordable Canadian university.
Human: st, Local: st, Global: st

The info. available in the net and the people who wish to offer services are too many and some are misleading.
Human: st, Local: st, Global: st

The preliminary preparations,eligibility,the require funds etc., are some of the issues which I wish to know from any panel members of this forum .. (truncated)
Human: ques, Local: st, Global: st

C_3 (truncated)...take a list of canadian universities and then create a table and insert all the relevant information by reading each and every program info on the web.
Human: sug, Local: sug, Global: sug

Without doing a research my advice would be to apply to UVIC .. for the following reasons .. (truncated)
Human: sug, Local: sug, Global: sug

UBC is good too... but it is expensive particularly for international students due to tuition .. (truncated)
Human: sug, Local: sug, Global: sug

most of them accept on-line or email application.
Human: st, Local: st, Global: st

Good luck !!
Human: pol, Local: pol, Global: pol

C_4 snakyy21: UVIC is a short form of? I have already started researching for my brother and found "College of North Atlantic" and .. (truncated)
Human: ques, Local: st, Global: ques

but not sure about the reputation..
Human: st, Local: res, Global: st

C_5 thank you for sharing useful tips will follow your advise.
Human: pol, Local: pol, Global: pol

Figure 1: Example conversation with **Human** annotations and automatic predictions by a **Local** classifier and a **Global** classifier. The labels **st**, **ques**, **sug**, and **pol** refers to *Statement*, *Question*, *Suggestion*, and *Polite* speech acts, respectively.

Identification of speech acts is an important step towards deep conversation analysis in these media (Bangalore et al., 2006), and has been shown to be useful in many downstream applications including summarization (McKeown et al., 2007) and question answering (Hong and Davison, 2009).

Previous attempts to automatic (sentence-level)

speech act recognition in asynchronous conversation (Qadir and Riloff, 2011; Jeong et al., 2009; Tavafi et al., 2013; Oya and Carenini, 2014) suffer from at least one of the two major flaws.

Firstly, they use bag-of-word (BOW) *representation* (e.g., unigram, bigram) to encode lexical information in a sentence. However, consider the *suggestion* sentences in the example. Arguably, a model needs to consider the structure (e.g., word order) and the compositionality of phrases to identify the right speech act. Furthermore, BOW representation could be quite sparse and may not generalize well when used in classification models.

Secondly, existing approaches mostly disregard conversational dependencies between sentences. For instance, consider the example again, where we tag the sentences with the human annotations ('Human') and with the predictions of a local ('Local') classifier that considers word order for sentence representation but classifies each sentence separately. Prediction errors are underlined and highlighted in red. Notice the first and second sentences of comment 4, which are tagged mistakenly as *statement* and *response*, respectively, by our best local classifier. We hypothesize that some of the errors made by the local classifier could be corrected by employing a global joint model that performs a collective classification taking into account the conversational dependencies between sentences (e.g., adjacency relations).

However, unlike synchronous conversations (e.g., phone, meeting), modeling conversational dependencies between sentences in asynchronous conversation is challenging, especially in those where explicit thread structure (reply-to relations) is missing, which is also our case. The conversational flow often lacks sequential dependencies in its temporal order. For example, if we arrange the sentences as they arrive in the conversation, it becomes hard to capture any dependency between the act types because the two components of the adjacency pairs can be far apart in the sequence. This leaves us with one open research question: how to model the dependencies between sentences in a single comment and between sentences across different comments? In this paper, we attempt to address this question by designing and experimenting with conditional structured models over arbitrary graph structure of the conversation.

More concretely, we make the following contributions. Firstly, we propose to use Recurrent Neural Network (RNN) with Long Short Term Memory (LSTM) hidden layer to perform composition of phrases and to represent sentences using distributed condensed vectors (i.e., embeddings). We experiment with both unidirectional and bidirectional RNNs. Secondly, we propose conditional structured models in the form of pairwise Conditional Random Field (Murphy, 2012) over arbitrary conversational structures. We experiment with different variations of this model to capture different types of interactions between sentences inside the comments and across the comments. These models use the LSTM encoded vectors as feature vectors for performing the classification task jointly. As a secondary contribution, we also present and release a forum dataset annotated with a standard speech act tagset.

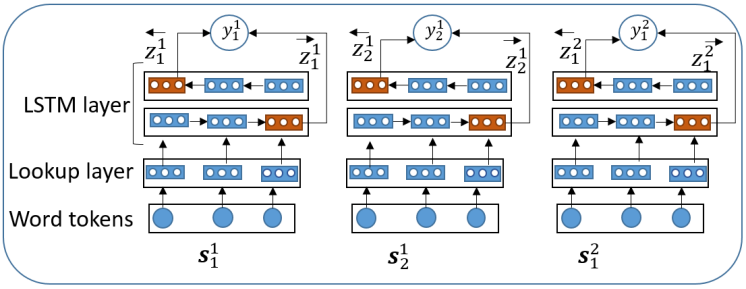
We train our models on different settings using synchronous and asynchronous corpora, and evaluate on two forum datasets. Our main findings are: (*i*) LSTM RNNs provide better representation than BOW; (*ii*) Bidirectional LSTMs, which encode a sentence using two vectors provide better representation than the unidirectional ones; and (*iii*) Global joint models improve over local models given that it considers the right graph structure. The source code and the new dataset are available at `http://alt.qcri.org/tools/speech-act/`

2 Our Approach

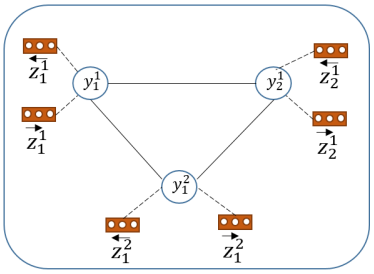
Let $\mathbf{s}_m^n$ denote the m-th sentence of comment n in a conversation. Our framework works in two steps as demonstrated in Figure 2. First, we use a recurrent neural network (RNN) to compose sentence representations semantically from their words and to represent them with distributed condensed vectors $\mathbf{z_m^n}$, i.e., sentence embeddings (Figure 2a). In the second step, a multivariate (graphical) model, which operates on the sentence embeddings, captures conversational dependencies between sentences in the conversation (Figure 2b). In the following, we describe the two steps in detail.

2.1 Sentence Representation

One of our main hypotheses is that a sentence representation method should consider the word order of the sentence. To this end, we use an LSTM RNN (Hochreiter and Schmidhuber, 1997) to encode a sentence into a vector by processing its words sequentially, at each time step combining



(a) Bidirectional LSTM-based RNN model



(b) Fully connected CRF model

Figure 2: Our two-step framework for speech act recognition in asynchronous conversation: (a) a bidirectional LSTM encodes each sentence $\mathbf{s}_m^n$ into a condensed vector $\mathbf{z_m^n}$ and classifies them separately; (b) a fully-connected CRF that takes the encoded vectors as input and performs joint learning and inference.

the current input with the previous hidden state. Figure 4b demonstrates the process for three sentences. Each word in the vocabulary V is represented by a D dimensional vector in a shared lookup table $L \in \mathbb{R}^{|V| \times D}$. L is considered a model parameter to be learned. We can initialize L randomly or by pretrained word embedding vectors like word2vec (Mikolov et al., 2013a).

Given an input sentence $\mathbf{s} = (w_1, \cdots, w_T)$, we first transform it into a feature sequence by mapping each token $w_t \in \mathbf{s}$ to an index in L. The lookup layer then creates an input vector $\mathbf{x_t} \in \mathbb{R}^D$ for each token w_t. The input vectors are then passed to the LSTM recurrent layer, which computes a compositional representation $\overrightarrow{\mathbf{h}}_t$ at every time step t by performing nonlinear transformations of the current input $\mathbf{x}_t$ and the output of the previous time step $\overrightarrow{\mathbf{h}}_{t-1}$. Specifically, the recurrent layer in a LSTM RNN is constituted with hidden units called *memory blocks*. A memory block is composed of four elements: (*i*) a memory cell c (a neuron) with a self-connection, (*ii*) an input gate i to control the flow of input signal into the neuron, (*iii*) an output gate o to control the effect of the neuron activation on other neurons, and (*iv*) a forget gate f to allow the neuron to adaptively reset its current state through the self-connection. The following sequence of equations describe how the memory blocks are updated at every time step t:

$$\mathbf{i}_t = \text{sigh}(U_i \mathbf{h}_{t-1} + V_i \mathbf{x}_t + \mathbf{b}_i) \quad (1)$$

$$\mathbf{f}_t = \text{sigh}(U_f \mathbf{h}_{t-1} + V_f \mathbf{x}_t + \mathbf{b}_f) \quad (2)$$

$$\mathbf{c}_t = \mathbf{i}_t \odot \tanh(U_c \mathbf{h}_{t-1} + V_c \mathbf{x}_t) + \mathbf{f}_t \odot \mathbf{c}_{t-1} \quad (3)$$

$$\mathbf{o}_t = \text{sigh}(U_o \mathbf{h}_{t-1} + V_o \mathbf{x}_t + \mathbf{b}_o) \quad (4)$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t) \quad (5)$$

where U_k and V_k are the weight matrices between two consecutive hidden layers, and between the input and the hidden layers, respectively, which are associated with gate k (input, output, forget and cell); and and $\mathbf{b}_k$ is the corresponding bias vector. The symbols sigh and tanh denote hard sigmoid and hard tan, respectively, and the symbol $\odot$ denotes a element-wise product of two vectors.

LSTM by means of its specifically designed gates (as opposed to simple RNNs) is capable of capturing long range dependencies. We can interpret $\mathbf{h}_t$ as an intermediate representation summarizing the past. The output of the last time step $\overrightarrow{\mathbf{h}}_T = \mathbf{z}$ thus represents the sentence, which can be fed to the output layer of the neural network (Fig. 4b) or to other models (e.g, a fully-connected CRF in Fig. 2b) for classification. The output layer of our LSTM-RNN uses a `softmax` for multiclass classification. Formally, the probability of k-th class for classification into K classes is

$$p(y = k|\mathbf{s}, \theta) = \frac{exp\ (\mathbf{w}_k^T \mathbf{z})}{\sum_{k=1}^{K} exp\ (\mathbf{w}_k^T \mathbf{z})} \quad (6)$$

where $\mathbf{w}$ are the output layer weights.

Bidirectionality The RNN described above encodes information that it gets only from the past. However, information from the future could also be crucial for recognizing speech acts. This is specially true for longer sentences, where a unidirectional LSTM can be limited in encoding the necessary information into a single vector. Bidirectional RNNs (Schuster and Paliwal, 1997) capture dependencies from both directions, thus provide two different views of the same sentence. This amounts to having a backward counterpart for each of the equations from 1 to 5. For classification, we use the concatenated vector $[\overrightarrow{\mathbf{h}}_T, \overleftarrow{\mathbf{h}}_T]$,

where $\overrightarrow{\mathbf{h}_T}$ and $\overleftarrow{\mathbf{h}_T}$ are the encoded vectors summarizing the past and the future, respectively.

2.2 Conditional Structured Model

Given the vector representation of the sentences in an asynchronous conversation, we explore two different approaches to learn classification functions. The first and the traditional approach is to learn a local classifier ignoring the structure in the output and to use it for predicting the label of each sentence separately. This is the approach we took above when we fed the output layer of the LSTM RNN with the sentence-level embeddings. However, this approach does not model the conversational dependency (e.g., *adjacency* relations between question-answer and request-accept pairs).

The second approach, which we adopt in this paper, is to model the dependencies between the output variables (labels) while learning the classification functions jointly by optimizing a global performance criterion. We represent each conversation by a graph $G{=}(V, E)$. Each node $i{\in}V$ is associated with an input vector $\mathbf{z}_i = \mathbf{z}^n_m$, representing the features of the sentence $\mathbf{s}^n_m$, and an output variable $y_i{\in}\{1, 2, \cdots, K\}$, representing the class label. Similarly, each edge $(i, j){\in}E$ is associated with an input feature vector $\phi(\mathbf{z}_i, \mathbf{z}_j)$, derived from the node-level features, and an output variable $y_{i,j}{\in}\{1, 2, \cdots, L\}$, representing the state transitions for the pair of nodes. We define the following conditional joint distribution:

$$p(\mathbf{y}|\mathbf{v}, \mathbf{w}, \mathbf{z}) = \frac{1}{Z(\mathbf{v}, \mathbf{w}, \mathbf{z})} \prod_{i \in V} \psi_n(y_i|\mathbf{z}, \mathbf{v}) \prod_{(i,j) \in E} \psi_e(y_{i,j}|\mathbf{z}, \mathbf{w}) \quad (7)$$

where ψ_n and ψ_e are node and the edge *factors*, and $Z(.)$ is the global normalization constant that ensures a valid probability distribution. We use a log-linear representation for the factors:

$$\psi_n(y_i|\mathbf{z}, \mathbf{v}) = \exp(\mathbf{v}^T \phi(y_i, \mathbf{z})) \quad (8)$$

$$\psi_e(y_{i,j}|\mathbf{z}, \mathbf{w}) = \exp(\mathbf{w}^T \phi(y_{i,j}, \mathbf{z})) \quad (9)$$

where $\phi(.)$ is a feature vector derived from the inputs and the labels. This model is essentially a pairwise conditional random field or PCRF (Murphy, 2012). The global normalization allows CRFs to surmount the so-called *label bias* problem (Lafferty et al., 2001), allowing them to take long-range interactions into account. The log likelihood for one data point $(\mathbf{z}, \mathbf{y})$ (i.e., a conversation) is:

$$f(\theta) = \sum_{i \in V} \mathbf{v}^T \phi(y_i, \mathbf{z}) + \sum_{(i,j) \in E} \mathbf{w}^T \phi(y_{i,j}, \mathbf{z}) - \log Z(\mathbf{v}, \mathbf{w}, \mathbf{z}) \quad (10)$$

This objective is convex, so we can use gradient-based methods to find the global optimum. The gradients have the following form:

$$f'(\mathbf{v}) = \sum_{i \in V} \phi(y_i, \mathbf{z}) - \mathbb{E}[\phi(y_i, \mathbf{z})] \quad (11)$$

$$f'(\mathbf{w}) = \sum_{(i,j) \in E} \phi(y_{i,j}, \mathbf{z}) - \mathbb{E}[\phi(y_{i,j}, \mathbf{z})] \quad (12)$$

where $\mathbb{E}[\phi(.)]$ denote the expected feature vector.

Training and Inference Traditionally, CRFs have been trained using offline methods like limited-memory BFGS (Murphy, 2012). Online training of CRFs using stochastic gradient descent (SGD) was proposed by Vishwanathan et al. (2006). Since RNNs are trained with online methods, to compare our two methods, we use SGD to train our CRFs. Algorithm 1 in the Appendix gives a pseudocode of the training procedure.

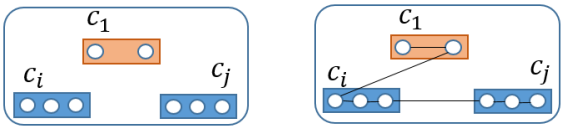
We use Belief Propagation or BP (Pearl, 1988) for inference in our graphical models. BP is guaranteed to converge to an exact solution if the graph is a tree. However, exact inference is intractable for graphs with loops. Despite this, it has been advocated by Pearl (1988) to use BP in loopy graphs as an approximation; see also (Murphy, 2012), page 768. The algorithm is then called "loopy" BP, or LBP. Although LBP gives approximate solutions for general graphs, it often works well in practice (Murphy et al., 1999), outperforming other methods such as mean field (Weiss, 2001).

Variations of Graph Structures One of the main advantages of our pairwise CRF is that we can define this model over arbitrary graph structures, which allows us to capture conversational dependencies at various levels. We distinguish between two types of dependencies: (*i*) *intra-comment*, which defines how the labels of the sentences in a comment are connected; and (*ii*) *across-comment*, which defines how the labels of the sentences across comments are connected.

Table 1 summarizes the connection types that we have explored in our models. Each configuration of intra- and across- connections yields a different pairwise CRF model. Figure 3 shows four such CRFs with three comments — C_1 being the first comment, and C_i and C_j being two other comments in the conversation.

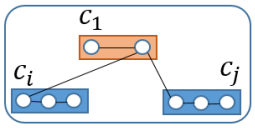
Tag	Connection type	Applicable to
NO	No connection between nodes	intra & across
LC	Linear chain connection	intra & across
FC	Fully connected	intra & across
FC_1	Fully connected with first comment only	across
LC_1	Linear chain with first comment only	across

Table 1: Connection types in CRF models.

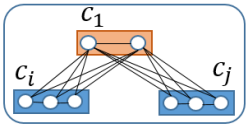


(a) NO-NO (MaxEnt)

(b) LC-LC



(c) $LC\text{-}LC_1$



(d) $LC\text{-}FC_1$

Figure 3: CRFs over different graph structures.

Figure 3a shows the structure for **NO-NO** configuration, where there is no link between nodes of both intra- and across- comments. In this setting, the CRF model is equivalent to MaxEnt. Figure 3b shows the structure for **LC-LC**, where there are linear chain relations between nodes of both intra- and across- comments. The linear chain across comments refers to the structure, where the last sentence of each comment is connected to the first sentence of the comment that comes next in the temporal order (i.e., posting time). Figures 3c shows the CRF for **LC-LC**$_1$, where sentences inside a comment have linear chain connections, and the last sentence of the first comment is connected to the first sentence of the other comments. Similarly, Figure 3d shows the graph structure for **LC-FC**$_1$ configuration, where sentences inside comments have linear chain connections, and sentences of the first comment are fully connected with the sentences of the other comments.

3 Corpora

There exist large corpora of utterances annotated with speech acts in synchronous spoken domains, e.g., Switchboard-DAMSL or SWBD (Jurafsky et al., 1997) and Meeting Recorder Dialog Act or MRDA (Dhillon et al., 2004). However, such large corpus does not exist in asynchronous domains. Some prior work (Cohen et al., 2004; Ravi and Kim, 2007; Feng et al., 2006; Bhatia et al., 2014) tackles the task at the comment level, and uses task-specific tagsets. In contrast, in this work we are interested in identifying speech acts at the sentence level, and also using a standard tagset like the ones defined in SWBD and MRDA.

	TA	BC3
Total number of conv.	200	39
Avg. nb of comments per conv.	4.02	6.54
Avg. nb of sentences per conv.	18.56	34.15
Avg. nb of words per sentence	14.90	12.61

Table 2: Statistics about TA and BC3 corpora.

Tag	Description	TA	BC3	MRDA
SU	Suggestion	7.71%	5.48%	5.97%
R	Response	2.4%	3.75%	15.63%
Q	Question	14.71%	8.41%	8.62%
P	Polite	9.57%	8.63%	3.77%
ST	Statement	65.62%	73.72%	66.00%

Table 3: Distribution of speech acts in our corpora.

More recent studies attempt to solve the task at the sentence level. Jeong et al. (2009) first created a dataset of TripAdvisor (TA) forum conversations annotated with the standard 12 act types defined in MRDA. They also remapped the BC3 email corpus (Ulrich et al., 2008) according to this tagset. Table 10 in the Appendix presents the tags and their relative frequency in the two datasets. Subsequent studies (Joty et al., 2011; Tavafi et al., 2013; Oya and Carenini, 2014) use these datasets. We also use these datasets in our work. Table 2 shows some basic statistics about these datasets. On average, BC3 conversations are longer than TA in both number of comments and number of sentences.

Since these datasets are relatively small in size, we group the 12 acts into 5 coarser classes to learn a reasonable classifier.[1] More specifically, all the question types are grouped into one general class *Question*, all response types into *Response*, and appreciation and polite mechanisms into *Polite* class. Also since deep neural models like LSTM RNNs require a lot of training data, we also utilize the MRDA meeting corpus. Table 3 shows the label distribution of the resultant datasets. Statement is the most dominant class, followed by Question, Polite and Suggestion.

QC3 Conversational Corpus Since both TA and BC3 are quite small to make a general comment about model performance in asynchronous

[1] Some prior work (Tavafi et al., 2013; Oya and Carenini, 2014) also took the same approach.

Speech Act	Distribution	κ
Suggestion	17.38%	0.86
Response	5.24%	0.43
Question	12.59%	0.87
Polite	6.13%	0.75
Statement	58.66%	0.78

Table 4: Corpus statistics for QC3.

Corpora	Type	Train	Dev.	Test
QC3	asynchronous	1252	157	156
TA	asynchronous	2968	372	371
BC3	asynchronous	1065	34	133
MRDA	synchronous	50865	8366	10492
Total	asyn. + sync.	56150	8929	11152

Table 5: Number of sentences in train, development and test sets for different datasets.

conversation, we have created a new dataset called Qatar Computing Conversational Corpus or QC3.

We selected 50 conversations from a popular community question answering site named Qatar Living[2] for our annotation. We used 3 conversations for our pilot study and used the remaining 47 for the actual study. The resultant corpus on average contains 13.32 comments and 33.28 sentences per conversation, and 19.78 words per sentence.

Two native speakers of English annotated each conversation using a web-based annotation framework. They were asked to annotate each sentence with the most appropriate speech act tag from the list of 5 speech act types. Since this task is not always obvious, we gave them detailed annotation guidelines with real examples. We use Cohens Kappa κ to measure the agreement between the annotators. Table 4 presents the distribution of the speech acts and their respective κ values.After Statement, Suggestion is the most frequent class, followed by Question and Polite. The κ varies from 0.43 (for Response) to 0.87 (for Question).

Finally, in order to create a consolidated dataset, we collected the disagreements and employed a third annotator to resolve those cases.

4 Experiments and Analysis

In this section we present our experimental settings, results and analysis. We evaluate our models on the two forum corpora QC3 and TA. For performance comparison, we use both *accuracy* and *macro-averaged* F_1 score. Accuracy gives the overall performance of a classifier but could be biased to most populated ones. Macro-averaged F_1 weights equally every class and is not influenced by class imbalance. Statistical significance tests are done using an *approximate randomization* test based on the accuracy.[3] We used SIGF V.2 (Padó, 2006) with 10,000 iterations.

[2]http://www.qatarliving.com/

[3]Significance tests operate on individual instances rather than individual classes; thus not applicable for macro F_1.

Because of the noise and informal nature of conversational texts, we performed a series of preprocessing steps. We normalize all characters to their lower-cased forms, truncate elongations to two characters, spell out every digit and URL. We further tokenized the texts using the CMU TweetNLP tool (Gimpel et al., 2011).

In the following, we first demonstrate the effectiveness of LSTM RNNs for learning representations of sentences automatically to identify their speech acts. Then in subsection 4.2, we show the usefulness of pairwise CRFs for capturing conversational dependencies in speech act recognition.

4.1 Effectiveness of LSTM RNNs

To show the effectiveness of LSTMs for learning sentence representations, we split each of our asynchronous corpora randomly into 70% *sentences* for training, 10% for development, and 20% for testing. For MRDA, we use the same train-test-dev split as Jeong et al. (2009). Table 5 summarizes the resultant datasets.

We compare the performance of LSTMs with that of MaxEnt (**ME**) and Multi-layer Perceptron (**MLP**) with one hidden layer.[4] Both ME and MLP were fed with the bag-of-word (**BOW**) representations of the sentence, i.e., vectors containing binary values indicating the presence or absence of a word in the training set vocabulary.

We train the models by optimizing the cross entropy using the gradient-based online learning algorithm ADAM (Kingma and Ba, 2014).[5] The learning rate and other parameters were set to the values as suggested by the authors. To avoid overfitting, we use dropout (Srivastava et al., 2014) of hidden units and early stopping based on the loss on the development set.[6] Maximum number of epochs was set to 25 for RNNs and 100 for ME and MLP. We experimented with $\{0.0, 0.2, 0.4\}$

[4]More hidden layers worsened the performance.

[5]Other algorithms (SGD, Adagrad) gave similar results.

[6]l_1 and l_2 regularization on weights did not work well.

dropout rates, $\{16, 32, 64\}$ minibatch sizes, and $\{100, 150, 200\}$ hidden layer units in MLP and in LSTMs. The vocabulary (V) in LSTMs was limited to the most frequent $P\%$ ($P \in \{85, 90, 95\}$) words in the training corpus. We initialize the word vectors in the loop-up table L in one of two ways: (*i*) by sampling randomly from the small uniform distribution $\mathcal{U}(-0.05, 0.05)$, and (*ii*) by using pretrained 300 dimensional Google word embeddings from Mikolov et al. (2013b). The dimension for random initialization was set to 128.

We experimented with four LSTM variations: (*i*) U-LSTM$_r$, referring to unidirectional with random initialization; (*ii*) U-LSTM$_p$, referring to unidirectional with pretrained initialization; (*iii*) B-LSTM$_r$, referring to bidirectional with random initialization; and (*iv*) B-LSTM$_p$, referring to bidirectional with pretrained initialization.

Table 6 shows the results for different models for the data splits in Table 5. The first two rows show the best results reported so far on the MRDA corpus from (Jeong et al., 2009) for classifying into 12 act types. The first row shows the results of the model that uses *n-grams* and the second row shows the results using *all* the features including speaker, part-of-speech, and dependency structure. Our LSTM RNNs and their n-gram model therefore use the same word sequence information. To compare our results with the state of the art, we ran our models on MRDA for both 5-class and 12-class classification tasks. The results are shown at the right most part of Table 6.

Notice that all of our LSTMs achieve state of the art results and B-LSTM$_p$ achieves even significantly better with 99% confidence level. This is remarkable since our LSTMs learn the sentence representation automatically from the word sequence and do not use any hand-engineered features.

Now consider the asynchronous domains QC3 and TA, where we show the results of our models based on 5-fold cross validation, in addition to the random (20%) testset. The 5-fold setting allows us to get more general performance of the models on a particular corpus. The comparison between our LSTMs shows that: (*i*) pretrained Google vectors provide better initialization to LSTMs than the random ones; (*ii*) bidirectional LSTMs outperform their unidirectional counterparts. When we compare these results with those of our baselines, the results are disappointing; the ME and MLP using BOW outperform LSTMs by a good margin.

	SU	R	Q	P	ST
SU	34	0	1	0	27
R	0	4	0	2	12
Q	0	0	64	0	13
P	0	0	1	35	6
ST	8	1	3	4	311

(a) B-LSTM$_p$

	SU	R	Q	P	ST
SU	21	1	1	0	39
R	0	6	0	1	11
Q	0	0	63	0	14
P	0	0	1	32	9
ST	8	2	0	2	316

(b) MLP

Figure 4: Confusion matrices for (a) B-LSTM$_p$ and (b) MLP on the testsets of QC3 and TA.

However, this is not surprising since deep neural networks like LSTMs have a lot of parameters, for which they require a lot of data to learn from.

To validate our claim, we create another *training* setting CAT by merging the training and development sets of the four corpora in Table 5 (see the Train and Dev. columns in the last row); the testset for each dataset however remains the same. Table 7 shows the results of the baselines and the B-LSTM$_p$ on the QC3 and TA testsets. In both datasets, B-LSTM$_p$ outperforms ME and MLP significantly. When we compare these results with those in Table 6, we notice that B-LSTM$_p$, by virtue of its distributed and condensed representation, generalizes well across different domains. In contrast, ME and MLP, because of their BOW representation, suffer from data diversity of different domains. These results also confirm that B-LSTM$_p$ gives better sentence representation than BOW, when it is given enough data.

To analyze further the cases where B-LSTM$_p$ makes a difference, Figure 4 shows the corresponding confusion matrices for B-LSTM$_p$ and MLP on the concatenated testsets of QC3 and TA. It is noticeable that B-LSTM$_p$ is less affected by class imbalance and it can detect more *suggestions* than MLP. This indicates that LSTM RNNs can model the grammar of the sentence when composing the words into phrases sequentially.

4.2 Effectiveness of CRFs

To demonstrate the effectiveness of CRFs for capturing inter-sentence dependencies in an asynchronous conversation, we create another dataset setting called CON, in which the random splits are done at the *conversation* (as opposed to sentence) level for the asynchronous corpora. This is required because our CRF models perform joint learning and inference based on a full conversation. As presented in Table 8, this setting contains 197 and 24 conversations for training and devel-

	QC3		TA		MRDA	
	Testset	5 folds	Testset	5 folds	5 classes	12 classes
Jeong et al. (ng)	-	-	-	-	-	57.53 (83.30)
Jeong et al. (All)	-	-	-	-	-	59.04 (83.49)
ME	55.12 (**75.64**)	50.23 (71.37)	61.4 (85.44)	59.23 (84.85)	65.25 (83.95)	57.79 (82.84)
MLP	**61.30** (74.36)	**54.57 (71.63)**	**68.17 (85.98)**	**62.41 (85.02)**	68.12 (84.24)	58.19 (83.24)
U-LSTM$_r$	51.57 (73.55)	48.64 (65.94)	56.54 (83.24)	56.39 (83.83)	71.29 (85.38)	58.72 (83.34)
U-LSTM$_p$	49.41 (70.97)	50.26 (65.62)	63.12(83.78)	59.10 (83.13)	72.32 (85.19)	59.05 (84.06)
B-LSTM$_r$	50.75 (72.26)	48.41 (66.19)	58.88 (82.97)	56.23 (83.34)	71.69 (85.62)	58.33 (83.49)
B-LSTM$_p$	53.22 (71.61)	51.59 (68.50)	60.73 (82.97)	59.68 (84.07)	72.02 (85.33)	**60.12 (84.46*)**

Table 6: Macro-averaged F_1 and raw accuracy (in parenthesis) for baselines and LSTM variants on the testset and 5-fold splits of different corpora. For MRDA, we use the same train-test-dev split as (Jeong et al., 2009). Accuracy significantly superior to state-of-the-art is marked with *.

	QC3 (Testset)	TA (Testset)
ME	50.64 (71.15)	72.49 (84.10)
MLP	58.60 (74.36)	73.07 (86.29)
B-LSTM$_p$	**66.40 (80.65*)**	**73.14 (87.01*)**

Table 7: Results on CAT dataset.

	Train	Dev	Test
QC3	38 (1332)	4 (111)	5 (122)
TA	160 (2957)	20 (310)	20 (444)
Total	197 (4289)	24 (421)	25 (566)

Table 8: Setting for CON dataset. The numbers inside parentheses indicate the number of sentences.

	QC3	TA
ME$_b$	56.67 (67.21)	63.29 (84.23)
B-LSTM$_p$	65.15 (77.87)	66.93 (85.13)
ME$_e$	59.94 (77.05)	59.55 (85.14)
CRF (LC-NO)	62.20 (77.87)	60.30 (85.81)
CRF (LC-LC)	62.35 (78.69)	60.30 (85.81)
CRF (LC-LC$_1$)	**65.94 (80.33*)**	61.58 (86.54)
CRF (LC-FC$_1$)	61.18 (77.87)	60.00 (85.36)
CRF (FC-FC)	64.54 (79.51*)	**61.64 (86.81*)**

Table 9: Results of CRFs on CON dataset.

opment, respectively.[7] The testsets contain 5 and 20 conversations for QC3 and TA, respectively.

As baselines, we use three models: (*i*) **ME**$_b$, a MaxEnt using BOW representation; (*ii*) **B-LSTM**$_p$, which is now trained on the concatenated set of sentences from MRDA and CON training sets; and (*iii*) **ME**$_e$, a MaxEnt using sentence embeddings extracted from the B-LSTM$_p$, i.e., the sentence embeddings are used as feature vectors.

We experiment with the CRF variants in Table 1. The CRFs are trained on the CON training set using the sentence embeddings that are extracted by applying the B-LSTM$_p$ model, as was done with ME$_e$. Table 9 shows our results. We notice that CRFs generally outperform MEs in accuracy. This indicates that there are conversational dependencies between the sentences in a conversation.

When we compare between CRF variants, we notice that the model that does not consider any link across comments perform the worst; see CRF (LC-NO). A simple linear chain connection between sentences in their temporal order does not improve much (CRF (LC-LC)), which indicates that the widely used linear chain CRF (Lafferty et al., 2001) is not the most appropriate model for capturing conversational dependencies in these conversations. The CRF (LC-LC$_1$) is one of the best performing models and perform significantly (with 99% confidence) better than B-LSTM$_p$.[8] This model considers linear chain connections between sentences inside comments and only to the first comment. Note that both QC3 and TA are forum sites, where participants in a conversation interact mostly with the person who posts the first comment asking for some information. This is interesting that our model can capture this aspect.

Another interesting observation is that when we change the above model to consider relations with every sentence in the first comment (CRF (LC-FC$_1$)), this degrades the performance. This could be due to the fact that the information seeking person first explains her situation, and then asks for the information. Others tend to respond to the requested information rather than to her situation. The CRF (FC-FC) also yields as good results as CRF (LC-LC$_1$). This could be attributed to the robustness of the fully-connected CRF, which learns

[7] We use the concatenated sets as train and dev. sets.

[8] Significance was computed on the concatenated testset.

from all possible relations.

To see some real examples in which CRF by means of its global learning and inference makes a difference, let us consider the example in Figure 1 again. We notice that the two sentences in comment C_4 were mistakenly identified as Statement and Response, respectively, by the B-LSTM$_p$ local model. However, by considering these two sentences together with others in the conversation, the global CRF (FC-FC) model could correct them.

5 Related Work

Three lines of research are related to our work: (*i*) semantic compositionality with LSTM RNNs, (*ii*) conditional structured models, and (*iii*) speech act recognition in asynchronous conversations.

LSTM RNNs for composition Li et al. (2015) compare recurrent neural models with recursive (syntax-based) models for several NLP tasks and conclude that recurrent models perform on par with the recursive for most tasks (or even better). For example, recurrent models outperform recursive on sentence level sentiment classification. This finding motivated us to use recurrent models rather than recursive. The application of LSTM RNNs to speech act recognition is novel to the best of our knowledge. LSTM RNNs have also been applied to sequence tagging in opinion mining (Irsoy and Cardie, 2014; Liu et al., 2015).

Conditional structured models There has been an explosion of interest in CRFs for solving structured output problems in NLP; see (Smith, 2011) for an overview. Linear chain (for sequence labeling) and tree structured CRFs (for parsing) are the common ones in NLP. However, speech act recognition in asynchronous conversation posits a different problem, where the challenge is to model arbitrary conversational structures. In this work we propose a general class of models based on pairwise CRFs that work on arbitrary graph structures.

Speech act recognition in asynchronous conversation Jeong et al. (2009) use semi-supervised boosting to tag the sentences in email and forum discussions with speech acts by adapting knowledge from spoken conversations. Other sentence-level approaches use supervised classifiers and sequence taggers (Qadir and Riloff, 2011; Tavafi et al., 2013; Oya and Carenini, 2014).

Cohen et al. (2004) first use the term *email speech act* for classifying emails based on their acts (deliver, meeting). Their classifiers do not capture any contextual dependencies between the acts. To model contextual dependencies, Carvalho and Cohen (2005) use a collective classification approach with two different classifiers, one for content and one for context, in an iterative algorithm. Our approach is similar in spirit to their approach with three crucial differences: (*i*) our CRFs are globally normalized to surmount the *label bias* problem, where their classifiers are normalized locally; (*ii*) the graph structure of the conversation is given in their case, which is not the case with ours; and (*iii*) their approach works at the comment level, where we work at the sentence level.

6 Conclusions and Future Work

We have presented a two-step framework for speech act recognition in asynchronous conversation. A LSTM RNN first composes sentences into vector representations by considering the word order. Then a pairwise CRF jointly models the inter-sentence dependencies in the conversation. We experimented with different LSTM variants (uni- vs. bi-directional, random vs. pretrained initialization), and different CRF variants depending on the underlying graph structure. We trained our models on many different settings using synchronous and asynchronous corpora and evaluated on two forum datasets, one of which is presented in this work.

Our results show that LSTM RNNs provide better representations but requires more data, and global joint models improve over local models given that it considers the right graph structure.

In the future, we would like to combine CRFs with LSTMs for doing the two steps jointly, so that the LSTMs can learn the embeddings using the global thread-level feedback. This would require the backpropagation algorithm to take error signals from the loopy BP inference. We would also like to apply our models to conversations, where the graph structure is extractable using the meta data or other clues, e.g., the fragment quotation graphs for email threads (Carenini et al., 2008).

Acknowledgments

We thank Aseel Ghazal for her effort in creating the QC3 corpus. This work is part of the Interactive sYstems for Answer Search (IYAS) project.

References

J. L. Austin. 1962. How to do things with words. *Harvard University Press*.

Srinivas Bangalore, Giuseppe Di Fabbrizio, and Amanda Stent. 2006. Learning the structure of task-driven human-human dialogs. In *Proceedings of the 44nd Annual Meeting on Association for Computational Linguistics, ACL'06*, pages 201–208. ACL.

Sumit Bhatia, Prakhar Biyani, and Prasenjit Mitra. 2014. Summarizing online forum discussions – can dialog acts of individual messages help? In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2127–2131, Doha, Qatar, October. ACL.

Giuseppe Carenini, Raymond T. Ng, and Xiaodong Zhou. 2008. Summarizing emails with conversational cohesion and subjectivity. In *Proceedings of the 46nd Annual Meeting on Association for Computational Linguistics, ACL'08*, pages 353–361, OH. ACL.

Vitor R. Carvalho and William W. Cohen. 2005. On the collective classification of email "speech acts". In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 345–352, New York, NY, USA. ACM Press.

William W. Cohen, Vitor R. Carvalho, and Tom M. Mitchell. 2004. Learning to classify email into "speech acts". In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 309–316.

Rajdip Dhillon, Sonali Bhagat, Hannah Carvey, and Elizabeth Shriberg. 2004. Meeting Recorder Project: Dialog Act Labeling Guide. Technical report, ICSI Tech. Report.

Donghui Feng, Erin Shaw, Jihie Kim, and Eduard Hovy. 2006. Learning to detect conversation focus of threaded discussions. In *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, HLT-NAACL '06, pages 208–215, Stroudsburg, PA, USA. Association for Computational Linguistics.

Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A Smith. 2011. Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 42–47. ACL.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Liangjie Hong and Brian D. Davison. 2009. A classification-based approach to question answering in discussion boards. In *32nd Annual International ACM SIGIR Conference on Research and Development on Information Retrieval*, pages 171–178, Boston, USA. ACM Press.

Ozan Irsoy and Claire Cardie. 2014. Opinion mining with deep recurrent neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 720–728, Doha, Qatar. ACL.

Minwoo Jeong, Chin-Yew Lin, and Gary Geunbae Lee. 2009. Semi-supervised speech act recognition in emails and forums. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1250–1259, Singapore. ACL.

Shafiq Joty, Giuseppe Carenini, and Chin-Yew Lin. 2011. Unsupervised Modeling of Dialog Acts in Asynchronous Conversations. In *Proceedings of the twenty second International Joint Conference on Artificial Intelligence*, IJCAI'11, pages 1–130, Barcelona.

Dan Jurafsky, Liz Shriberg, and Debra Biasca. 1997. Switchboard SWBD-DAMSL Shallow-Discourse-Function Annotation Coders Manual, Draft 13. Technical report, University of Colorado at Boulder & +SRI International.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA.

Jiwei Li, Thang Luong, Dan Jurafsky, and Eduard Hovy. 2015. When are tree structures necessary for deep learning of representations? In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2304–2314, Lisbon, Portugal, September. ACL.

Pengfei Liu, Shafiq Joty, and Helen Meng. 2015. Fine-grained opinion mining with recurrent neural networks and word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1433–1443, Lisbon, Portugal, September. ACL.

Kathleen McKeown, Lokesh Shrestha, and Owen Rambow. 2007. Using question-answer pairs in extractive summarization of email conversations. In *CICLing*, pages 542–550.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.

Kevin P. Murphy, Yair Weiss, and Michael I. Jordan. 1999. Loopy belief propagation for approximate inference: An empirical study. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, UAI'99, pages 467–475, Stockholm, Sweden. Morgan Kaufmann Publishers Inc.

Kevin Murphy. 2012. *Machine Learning A Probabilistic Perspective*. The MIT Press.

Tatsuro Oya and Giuseppe Carenini. 2014. Extractive summarization and dialogue act modeling on email threads: An integrated probabilistic approach. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, page 133–140, Philadelphia, PA, U.S.A. ACL.

Sebastian Padó, 2006. *User's guide to `sigf`: Significance testing by approximate randomisation*.

Judea Pearl. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

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. Association for Computational Linguistics.

Sujith Ravi and Jihie Kim. 2007. Profiling Student Interactions in Threaded Discussions with Speech Act Classifiers. In *Proceedings of AI in Education Conference (AIED 2007)*.

Emanuel A. Schegloff. 1968. Sequencing in conversational openings1. *American Anthropologist*, 70(6):1075–1095.

Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Noah A. Smith. 2011. *Linguistic Structure Prediction*. Synthesis Lectures on Human Language Technologies. Morgan and Claypool, May.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

Maryam Tavafi, Yashar Mehdad, Shafiq Joty, Giuseppe Carenini, and Raymond Ng. 2013. Dialogue act recognition in synchronous and asynchronous conversations. In *Proceedings of the 14th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, page 117–121, Metz, France. ACL.

Jan Ulrich, Gabriel Murray, and Giuseppe Carenini. 2008. A publicly available annotated corpus for supervised email summarization. In *AAAI'08 EMAIL Workshop*, Chicago, USA. AAAI.

S. V. N. Vishwanathan, Nicol N. Schraudolph, Mark W. Schmidt, and Kevin P. Murphy. 2006. Accelerated training of conditional random fields with stochastic gradient methods. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 969–976, Pittsburgh, USA. ACM.

Y. Weiss. 2001. Comparing the mean field method and belief propaga- tion for approximate inference in mrfs. In *Advanced Mean Field Methods*. MIT Press.

Algorithm 1: Online learning algorithm for conditional random fields

1. Initialize the model parameters $\mathbf{v}$ and $\mathbf{w}$;
2. **repeat**
   for *each thread* $G = (V, E)$ **do**
      a. Compute node and edge factors $\psi_n(y_i|\mathbf{z}, \mathbf{v})$ and $\psi_e(y_{i,j}|\mathbf{z}, \mathbf{w})$;
      b. Infer node and edge marginals using sum-product loopy BP;
      c. Update: $\mathbf{v} = \mathbf{v} - \eta \frac{1}{|V|} f'(\mathbf{v})$;
      d. Update: $\mathbf{w} = \mathbf{w} - \eta \frac{1}{|E|} f'(\mathbf{w})$;
   end
   until *convergence*;

Tag	Description	BC3	TA
S	Statement	69.56%	65.62%
P	Polite mechanism	6.97%	9.11%
QY	Yes-no question	6.75%	8.33%
AM	Action motivator	6.09%	7.71%
QW	Wh-question	2.29%	4.23%
A	Accept response	2.07%	1.10%
QO	Open-ended question	1.32%	0.92%
AA	Acknowledge and appreciate	1.24%	0.46%
QR	Or/or-clause question	1.10%	1.16%
R	Reject response	1.06%	0.64%
U	Uncertain response	0.79%	0.65%
QH	Rhetorical question	0.75%	0.08%

Table 10: Dialog act tags and their relative frequencies in the BC3 and TA corpora.