Discovering Conversational Dependencies between Messages in Dialogs

Wenchao Du and **Pascal Poupart**
David R. Cheriton School of Computer Science
University of Waterloo
{w8du,ppoupart}@uwaterloo.ca

Wei Xu
Department of Computer Science and Engineering
The Ohio State University
xu.1265@osu.edu

Abstract

We investigate the task of inferring conversational dependencies between messages in one-on-one online chat, which has become one of the most popular forms of customer service. This task is important to recover the conversational structure of live chats that do not follow perfect turn taking and whose threads of discussion are entangled. It is also useful to identify coherent pairs of messages to train question answering and response generation techniques. We propose a simple probabilistic classifier that leverages conversational, lexical and semantic information. In particular, we developed a novel semantic model that considers the influences between messages, and used its predictions as features. The approach is evaluated empirically on set of customer service chat logs from a Chinese e-commerce website. It outperforms simple heuristic baselines, and the features provided by the semantic model were shown to be effective.

Introduction

Exposing conversational structure (Shen et al. 2006; Elsner and Charniak 2010) is a key step towards organizing the information in dialogues and is very useful for many applications, such as automatic response generation (Ritter, Cherry, and Dolan 2010; Sordoni et al. 2015) and discourse parsing (Afantenos et al. 2015). There has been a significant rise of interest in conversational response generation using statistical and neural machine translation. These approaches typically require a large number of message-response pairs or context-message-response triples as training data, and such data is usually obtained from human annotations. What remains a challenge is identifying coherent threads of discussion in conversations automatically, which is the goal of this paper. For question answering, finding the relevant context of questions through dependency modelling can help choosing the answer; for text generation, dependency modelling provides an effective way to annotate the suitable context-message-response triples for training. Such partitions/clusters are also useful for analysis on topics and frequently asked questions of conversations.

In online text messaging, one party may send two questions successively, and the other party may answer these questions in any order. In another scenario, one may send a message that does not respond to any of the other party's messages, but elaborate on oneself. These situations complicate the understanding of conversations. Fig. 1 illustrates a typical online chat where the correspondence between utterances is crucial for the understanding of conversations. When annotating the chats, each message is linked to the most relevant message and the candidates comprise of its previous messages and itself. As a result, each chat admits a structure of 1-regular directed graph, which has no cycle except self loops. The fact that the annotation structure is a forest and the number of trees in the forest is not known beforehand eliminates the possibility of applying standard decoding algorithms for discourse parsing such as maximum spanning tree and min-cut.

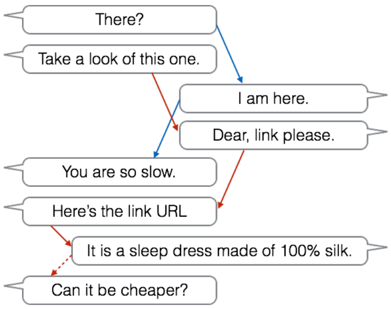

Figure 1: Example of the conversational structure between a customer and a customer service representative (solid arrow is a sure link; dotted arrow is a loose connection)

Related Work

Ritter et al. (2010) and Zhai et al. (2014) focused on speech acts, which only displays the discourse role of utterances (i.e. it tells you whether a message is question or an answer, but not tell you where its answer/question is). Afantenos et al. (2015) considered the discourse parsing problem in multi-party dialogue, but their model does not contain any semantic factor.

Data

We use the customer service logs from a Chinese e-commerce website. Customers mostly ask about products, promotions, delivery, and sometimes make bargains with

0	1	2	3	4	5	>5
33.4%	51.0%	10.4%	3.2%	1.2%	0.5%	0.4%

Table 1: Percentages of annotated links of various range

agents. Customers also ask for refund after products are delivered. Our dataset comprises 9000 chats of 5 to 60 utterances each. 2 and 3 show the lengths and exchange ratios (percentage of consecutive messages by different speakers) of these conversations. We randomly annotated 800 chats of 10 to 35 utterances with an exchange ratio between 0.4 and 0.6. This is because conversations of low exchange ratio have higher disorders in turn taking and thus more interesting, but if the ratio is too low, then the conversation did not proceed normally (usually it is that the agent was absent for too long). Annotation is carried out by 6 annotators who are native speakers. Each annotator is shown 3-5 example chats for training purpose. Each chat is annotated by 3 different people. Annotators were asked to link each message to a previous one with strongest coherence relation in the form of message-response pair, response-continuation pair, question-context pair, or link to itself (if there is no dependency on previous messages). A response-continuation pair is formed only when it is inappropriate to use the continuing message to answer a question directly. When in doubt, annotators were told to think like an agent and to select the most relevant dependency as if they were trying to respond to customers' messages themselves. Among the 800 chats annotated, 54.2% received the same label by all 3 annotators and 94.8% received at least 2 identical labels. Fleiss' Kappa (degree of agreement in classification over that which would be expected by chance) was 0.482. The number of classes for each message was 6 (link to itself or any of the last 5 messages). shows the statistics on the range of links.

It is not uncommon that text messages from customers are ungrammatical and have spelling errors, which makes using state-of-the-art tagging and dependency parsing tools difficult. We preprocessed the text with an open source word segmenter, python jieba, and replaced non-character lexicons and rare words by their type (e.g., links, emoticons, and geographical names). For our semantic model, we only used the most frequent 5,400 words.

Methods

For each message we consider the following binary features: 1) identity of speaker, 2) contains question words or question mark, 3) contains answer words, 4) contains URL, 5) contains image, 6) contains emoticon. Let $I[f_i^k = a] = 1$ when the k^{th} feature of message i takes value a (and 0 otherwise). We also consider the distance between two messages. Let $I[d_{ij} = m] = 1$ when message i is m utterances after message j (and 0 otherwise). We define the probability that

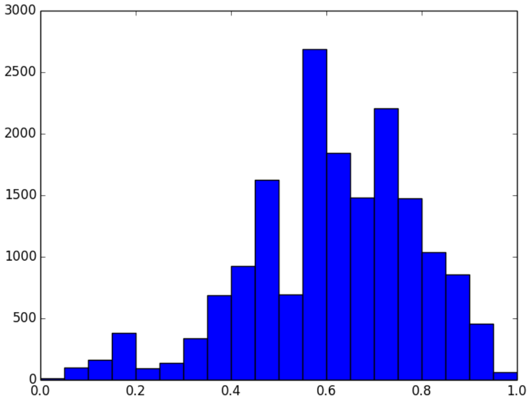

Figure 2: Histogram of exchange ratios

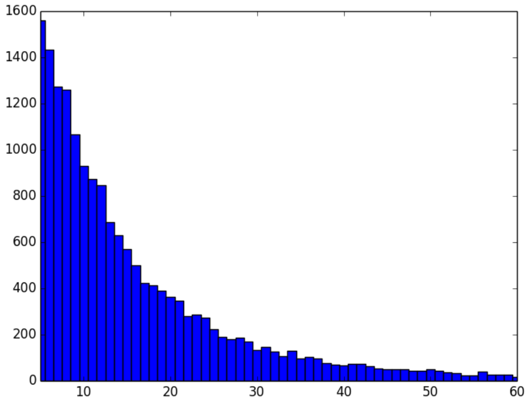

Figure 3: Histogram of lenghts of chats

message i depends on message j as follows:

$$p(c_i = j|f, d, \eta, \tau, \pi) \propto \exp\{(\sum_{k,l,a,b} I[f_i^k = a]I[f_j^l = b]I[i \neq j]\eta_{klab} + \sum_{k,a} I[i = j]I[f_i^k = a]\pi_{ka}) \sum_m I[d_{ij} = m]\tau_m\} \tag{1}$$

We train the coefficients η_{klab}, τ_m and π_{ka} by maximizing the conditional likelihood of identifying the correct links in the labeled chats with L2 regularization. This optimization was done by limited memory BFGS implemented in SciPy. Note that with this model, for the first few messages of a conversation there are less candidates so the normalization terms are different. Therefore we train separate classifiers for the first few messages of each conversation. We call this model without semantic features *Discriminative*.

Now we want to incorporate semantic information. Suppose we have a semantic model that tells us the rankings of candidates based on semantic similarity. Let $I[s_ij = p]$ be the indicator of message j being ranked p^{th} relevant to message i by the semantic model. Then we define the probability that message i depends on message j as

$$
\begin{aligned}
&p(c_i = j|f, d, \eta, \tau, \pi) \propto \\
&\quad \exp\{(\sum_{k,l,a,b} I[f_i^k = a] I[f_j^l = b] I[i \neq j] \eta_{klab} + \\
&\quad \sum_{k,a} I[i = j] I[f_i^k = a] \pi_{ka})(\sum_m I[d_{ij} = m] \tau_m + \\
&\quad \sum_p I[s_{ij} = p] \lambda_p + \sum_{m,p} I[d_{ij} = m] I[s_{ij} = p] \mu_{mp})\}
\end{aligned} \tag{2}
$$

Semantic Model

We want to measure semantic similarity. A natural choice is model the topics of messages through LDA and base the similarity measure on the posterior of LDA. One inherent problem of this approach is, LDA requires making assumption on how the messages are grouped, and it further assumes that each group of messages generates topics independently. However, coherent pair of messages usually have similar topics, while we don't know the partition of messages beforehand, so standard LDA will fail to capture this information. Instead, we relax the independence assumption use a distance measure to hypothesize how the messages are linked and the topics are generated.

Our semantic model has a generative process descirbed as follows:

1. Each new message coming in, i, is assigned to a previous message j or itself based on probabilities given by $p(c_i = j|\mathbf{x}, \eta)$. The target is denoted by $c(i)$
2. The new message i generates topics z_i using $\theta_{c(i)}$
3. Topics z_i generate words w_i using β

The generating process is inspired from distance dependent Chinese restaurant process. The difference is here we do not consider the table partition resulted from assignments. The likelihood of this process can be written as

$$
\begin{aligned}
&p(\mathbf{w}, \mathbf{z}, \mathbf{c}, \theta|\alpha, \beta, \mathbf{x}) = p(\mathbf{c}) \cdot \\
&\prod_{j=1}^{N} p(\theta_j|\alpha) \prod_{i=1, c_i = j}^{N} p(z_i|\theta_j) p(w_i|z_i, \beta)
\end{aligned} \tag{3}
$$

where

$$
p(\mathbf{c}) = \prod_{i=1}^{N} p(c_i) \tag{4}
$$

Rewrite the distribution by putting c into the exponent of $p(z|\theta)$

$$
\begin{aligned}
&p(\mathbf{w}, \mathbf{z}, \mathbf{c}, \theta|\alpha, \beta) = p(\mathbf{c}) \cdot \\
&\prod_{j=1}^{N} \Big(p(\theta_j|\alpha) \prod_{i=1}^{N} p(z_i|\theta_j)^{c_{ij}} \Big) p(w_j|z_j, \beta)
\end{aligned} \tag{5}
$$

where w_i, z_i, θ_i are the words, topics, and topic mixture parameter of i-th message, respectively. $c_{ij} = 1$ if i-th message is assigned to j-th message, otherwise 0. For conversation modelling, we use window decay distance measure, that is, the probability of linking is uniform over last 5 messages and self.

In order to weigh down the effect of common words, we adjust the multiplicity of words of messages based on their inverse document frequency (IDF). Specifically, the multiplicity of each occurrence of word i is given by

$$
\min\{1, \ln \frac{\#messages}{\#messages\ having\ i}\}
$$

Inference and Learning

We proceed with mean-field approximation of the posterior. The variational lower bound is given by

$$
\begin{aligned}
&\log p(\mathbf{w}|\alpha, \beta) \\
&\geq E_q[p(\mathbf{w}, \mathbf{z}, \mathbf{c}, \theta|\alpha, \beta)] - \\
&\quad E_q[q(\mathbf{z}, \mathbf{c}, \theta)] \\
&= E_q[\sum_{i,j} \log p(c_{ij}) - \log q(c_{ij}) + \\
&\quad \sum_j \log p(\theta_j|\alpha) - \log q(\theta_j) + \log p(w_j|z_j, \beta) + \\
&\quad \sum_i \sum_j c_{ij} \log p(z_i|\theta_j) - \log q(z_i)]
\end{aligned} \tag{6}
$$

Updating the variational posterior for z and θ is similar to LDA:

$$
\phi_{nl} \propto \sum_{j=1}^{N} \beta_{lv} exp(q(c_{ij})(\Psi(\gamma_{jl}) - \Psi(\sum_{l=1}^{k} \gamma_{jl})))
$$

$$
\gamma_{jl} = \alpha_{jl} + \sum_{i=1}^{N} q(c_{ij}) \phi_{il}
$$

Updating $q(c_{ij})$ is given by:

$$
q(c_{ij}) \propto \exp(\log p(c_{ij}) + E_q[\log p(z_i|\theta_j)])
$$

Semi-Supervised Learning

In order to leverage labelled messages, we assign more weight to the probability of messages with known links.

$$
t \sum_{i\ labelled} \log p(w_i, c_i|\alpha, \beta) + \sum_j \log p(w_j|\alpha, \beta) \tag{7}
$$

where t is the weight on labelled data.

Experiments

We compare with two rule-based baselines. Rule1: Each message is linked to its immediate precedent. Rule2: Each message is linked to its immediate precedent if the precedent is from the customer, otherwise it is linked to itself (i.e., customer/agent to customer, but not customer/agent to agent).

The table below compares our discriminative learning technique with and without the semantic similarity feature from LDA to the baselines. We report the average probability that each method would have labeled a message in the same way as one annotator (chosen uniformly at random among the 3 annotators) based on 5-fold cross validation. We also report the F1 measure (weighted average of harmonic mean of precision and recall of each class). Our discriminative learning technique outperforms the baselines, but the semantic similarity feature based on LDA did not yield a significant improvement. We also estimated human performance by scoring each annotator against the other two annotators, which yielded 0.677 ± 0.020. We can compute an upper bound on the best performance possible by choosing the label with highest agreement among the annotators for each data point, which yielded an accuracy of 0.830.

	Accuracy	Average F1
Rule-based Baseline 1	0.546	0.385
Rule-based Baseline 2	0.513	0.476
Discriminative	0.652	0.617
Discriminative + ddLDA	0.675	0.645

Table 2: Evaluation

Conclusion

We investigated how to expose the structure of conversations that do not follow perfect turn taking by identifying dependencies between utterances. We identified a set of relevant features and showed how to train a simple probabilistic model that can infer links. We explored a variant of LDA that takes into account influences between messages. We exploited the correlation between features to gain improvement. For future work, We plan to refine the definition of dependencies in order to obtain more consistent annotations which will help to improve the accuracy of the classifiers trained based on those annotations. We also plan to incorporate discourse relation into the model so it is more useful for dialog systems.

References

Afantenos, S.; Kow, E.; Asher, N.; and Perret, J. 2015. Discourse parsing for multi-party chat dialogues. In *EMNLP*.

Elsner, M., and Charniak, E. 2010. Disentangling chat. *Computational Linguistics* 36(3):389–409.

Ritter, A.; Cherry, C.; and Dolan, B. 2010. Unsupervised modeling of Twitter conversations. In *NAACL*.

Shen, D.; Yang, Q.; Sun, J.-T.; and Chen, Z. 2006. Thread detection in dynamic text message streams. In *SIGIR*.

Sordoni, A.; Galley, M.; Auli, M.; Brockett, C.; Ji, Y.; Mitchell, M.; Nie, J.-Y.; Gao, J.; and Dolan, B. 2015. A neural network approach to context-sensitive generation of conversational responses. In *NAACL*.

Zhai, K., and Williams, J. D. 2014. Discovering latent structure in task-oriented dialogues. In *ACL (1)*, 36–46.