Context-based Message Expansion for Disentanglement of Interleaved Text Conversations

Lidan Wang
Computer Science Dept./UMIACS
University of Maryland, College Park
College Park, MD 20742
lidan@cs.umd.edu

Douglas W. Oard
College of Information Studies/UMIACS
and HLT Center of Excellence
University of Maryland, College Park
College Park, MD 20742
oard@umd.edu

Abstract

Computational processing of text exchanged in interactive venues in which participants engage in simultaneous conversations can benefit from techniques for automatically grouping overlapping sequences of messages into separate conversations, a problem known as "disentanglement." While previous methods exploit both lexical and non-lexical information that exists in conversations for this task, the inter-dependency between the meaning of a message and its temporal and social contexts is largely ignored. Our approach exploits contextual properties (both explicit and hidden) to probabilistically expand each message to provide a more accurate message representation. Extensive experimental evaluations show our approach outperforms the best previously known technique.

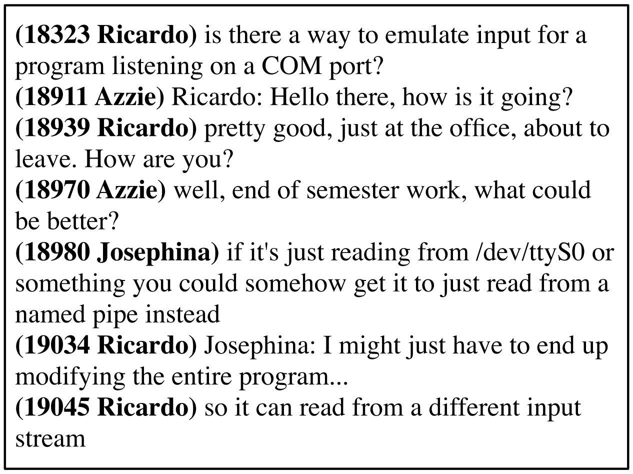
1 Introduction

Conversational media such as the text messages found in Internet Relay Chat presents both new opportunities and new challenges. Among the challenges are that individual messages are often quite short, for the reason that conversational participants are able to assemble the required context over the course of a conversation. A natural consequence of this is that many tasks that we would like to perform on conversational media (e.g., search, summarization, or automated response) would benefit from reassembly of individual messages into complete conversations. This task has been studied extensively in the context of email (where it is often referred to as "threading") (Yeh et al., 2006). The extensive metadata associated with email and the relatively rich content of some email messages makes email somewhat of a special case in the broad set of conversation recovery tasks, however. At the opposite extreme, conversation "threading" in multi-party spoken interactions (e.g., meetings) would be a compelling application, but the word error rate of current automated transcription techniques somewhat limits access to the lexical evidence that we know is useful for this task. The recent interest in identifying individual conversations from online-discussions, a task that some refer to as "disentanglement," therefore seems to be something of a middle ground in the research space: computationally tractable, representative to some degree of a broader class of problems, and directly useful as a pre-processing step for a range of important applications.

One way to think of this task is as a clustering problem—we seek to partition the messages into a set of disjoint clusters, where each cluster represents a conversation among a set of participants on a topic. This formulation raises the natural question of how we should design a similarity measure. Since the messages are often too short to be meaningful by themselves, techniques based solely on lexical overlap (e.g., inner products of term vectors weighted by some function of term frequency, document frequency and message length) are unlikely to be successful. For instance, consider the multi-party exchange in Figure 1, in which a single message may not convey much about the topic without considering what has been said before, and who said it.

Fortunately for us, additional sources of evidence

Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the ACL, pages 200–208,
Boulder, Colorado, June 2009. ©2009 Association for Computational Linguistics



(18323 Ricardo) is there a way to emulate input for a program listening on a COM port?
(18911 Azzie) Ricardo: Hello there, how is it going?
(18939 Ricardo) pretty good, just at the office, about to leave. How are you?
(18970 Azzie) well, end of semester work, what could be better?
(18980 Josephina) if it's just reading from /dev/ttyS0 or something you could somehow get it to just read from a named pipe instead
(19034 Ricardo) Josephina: I might just have to end up modifying the entire program...
(19045 Ricardo) so it can read from a different input stream

Figure 1: An example of the text message stream. The number before each author's name denotes the time-stamp of the message.

are available. As we describe below, messages are strongly correlated both temporally (i.e., across time) and socially (i.e,, across participants). For example, in our running example in Figure 1, Ricardo's message (19045 Ricardo) "so it can read from a different input stream" elaborates on his previous message (19034 Ricardo) to Josephina. Messages that are close in time and from the same speaker can share related meanings. Similarly, we see that Ricardo's messages to Josephina (19034 Ricardo and 19045 Ricardo) are responses to earlier comments made by Josephina (18980 Josephina), and that fact is signaled by Ricardo invoking Josephena's name. This is an example of social correlation: lexicalized references to identity can also provide useful evidence. If we take social and temporal context into account, we should be able to do better at recognizing conversations than we could using lexical overlap alone.

In recent years, several approaches have been developed for detecting conversational threads in dynamic text streams (Elsner et al., 2008; Shen et al., 2006; Wang et al., 2008). Although they use both lexical and non-lexical information (e.g., time, name mentions in message) for this task, they have ignored the temporal and social contexts a message appears in, which provide valuable cues for interpreting the message. *Correlation clustering* used in a two-step approach (Elsner et al., 2008) exploits message contexts to some degree, but its performance is largely limited by the classifier used in the first-step which computes message similarity without considering the temporal and social contexts of each message.

Our approach exploits contextual properties (both explicit and hidden) to probabilistically expand each message to provide a more accurate message representation. The new representation leads to a much improved performance for conversation disentanglement. We note that this is a general approach and can be applied to the representation of non-chat data that exhibits temporal and social correlations as well. The results that we obtain with this technique are close to the limit of what we can measure using present test collections and evaluation measures. To the best of our knowledge, our work is the first to apply document expansion to the conversation disentanglement problem.

2 Related Work

Previous work in conversation disentanglement (i.e. thread detection) has shown the conventional lexical-based clustering is not suitable for text streams because messages are often too short and incomplete. They focus on using discourse/chat-specific features to bias the lexical-based message similarity (Elsner et al., 2008; Shen et al., 2006; Wang et al., 2008). These features provide the means to link messages that may not have sufficient lexical overlap but are nevertheless likely to be topically related. However, our work is different from them in several aspects:

(1) They treat individual messages as the basic elements for clustering, and ignore the social and temporal contexts of the messages. In our work, each message is probabilistically expanded using reliable information from its contexts and the expanded messages are the basic elements for clustering.

(2) Messages have different amount of explicit information. For example, messages that initiate conversations may have more name mentions than subsequent messages (i.e. for establishing conversations). Previous work only uses what are explicitly present in each message, and clusters may be erroneously assigned for messages that lack enough explicit in-

formation. Our work exploits both explicit and implicit context for each message due to how we define contexts (Section 3.2.1).
(3) Most work imposes a fixed window size for clustering and it may break up long conversations or may not be fine-grained enough for short conversations. Given each message, we use an exponential decay model to naturally encode time effect and assign differential weights to messages in its contexts.

Another thread of related work is document expansion. It was previously studied in (Singhal et al., 1999) in the context of the speech retrieval, helping to overcome limitations in the transcription accuracy by selecting additional terms from lexically similar (text) documents. Document expansion has also been applied to cross-language retrieval in (Levow et al., 2005), in that case to overcome limitations in translation resources. The technique has recently been re-visited (Tao et al., 2006; Kurland et al., 2004; Liu et al., 2004) in the language modeling framework, where lexically related documents are used to enlarge the sample space for a document to improve the accuracy of the estimated document language model. However, these lexical-based approaches are less well suited to conversational interaction, because conversational messages are often short, they therefore may not overlap sufficiently in words with other messages to provide a useful basis for expansion. Our technique can be viewed as an extension of these previous methods to text streams.

Our work is also related to text segmentation (Ji et al., 2003) and meeting segmentation (Malioutov et al., 2006; Malioutov et al., 2007; Galley et al., 2003; Eisenstein et al., 2008). Text segmentation identifies boundaries of topic changes in long text documents, but we form threads of messages from streams consisting of short messages. Meeting conversations are not as highly interleaving as chat conversations, where participants can create a new conversation at any time.

3 Method

This section describes our technique for clustering messages into threads based on the lexical similarity of documents that have been expanded based on social and temporal evidence.

3.1 Context-Free Message Model

To represent the semantic information of messages and threads (clusters of messages), most of the prior approaches build a document representation on each message *alone* (using word features and time-stamp and/or discourse features found in the message). We call such a model a context-free message model. Most commonly, a message is represented as a vector (Salton, 1989). Each dimension corresponds to a separate term. If a term occurs in the message, its value in the vector is non-zero. Several different ways of computing these values, known as term weights, have been developed. One of the best known schemes is tf-idf weighting.

However, in conversational text, a context-free model cannot fully capture the semantics of messages. The meaning of a message is highly dependent on other messages in its context. For example, in our running example in Figure 1, to fully interpret the message 19045 Ricardo, we need to first read his previous message (19034 Ricardo) to Josephina. Further, messages on the same topic may have little or no overlap in words (Figure 1), and the messages between participants are highly interactive and are often too short and incomplete to fully capture a topic on their own.

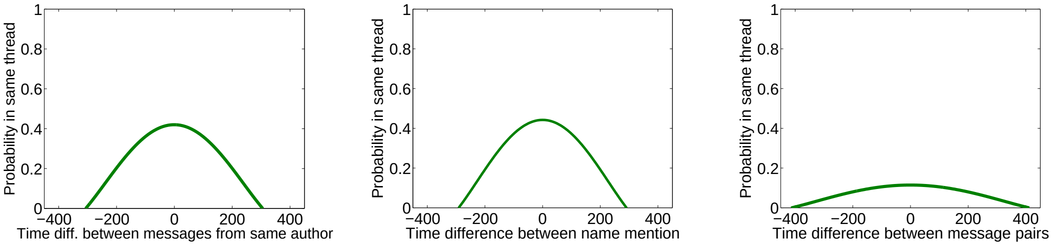
3.2 Context-Sensitive Message Model

Our main idea is to exploit the temporal and social aspects of the conversations to build a context-sensitive document model for each message. We do this by first identifying the temporal and social contexts for each message, then probabilistically expanding the content of each message with selected messages in each context. As we have seen, a message's contexts provide valuable cues for interpreting the message. Finally, we cluster the messages into distinct conversations based on their new representation models.

We present the formal definitions of each context and discuss how to model them in Section 3.2.1. In Section 3.2.2, we show how to efficiently identify the related messages in each context, and how to use them to expand our representation of the message.

3.2.1 Social and Temporal Contexts

Social contexts: we define two kinds of social contexts: author context and conversational context. We



(i) (ii) (iii)

Figure 2: (i) Relationship between messages from the same author (ii) Relationship between messages that mention each other's authors, and (iii) All pairs of messages as a function of time. Estimation is based on training data used in experiments.

explain them in detail below.

Author context: the author context of a message m, denoted by $C_A(m)$, is the set of other messages written by m's author a_m:

$$C_A(m) = \{m_i | a_{m_i} = a_m, m \neq m_i\}$$

Further, because of the nature of human conversations, we would be less surprised to find messages from the same person belonging to the same conversation if they are close in time rather than far apart. This is illustrated in Figure 2(i) [1], which shows the probability that a pair of messages written by the same person belong to the same conversation as a function of the time difference between them. Not surprisingly, messages in m's author context have probabilities which are influenced by their temporal proximity to m.

We use a normal distribution (Figure 2(i)) to encode the notion of author context. Given two messages m_i and m_j written by the *same* author, each with time-stamp t_i and t_j, respectively, the probability that m_j is topically related to m_i given their time difference $d = t_j - t_i$ is:

$$P_a(d) = N(\mu_a, \sigma_a^2) = \frac{1}{\sigma_a \sqrt{2\pi}} e^{-\frac{(d-\mu_a)^2}{2\sigma_a^2}}$$

The exponential decay helps to limit the influence from temporally remote messages. For message m_i, this distribution models the uncertainty that messages in its author context (i.e. other messages m_j from the same author) belong to the same conversation by assigning assigning a high value to m_j if $t_j - t_i$ is small. The mean μ_a is chosen to be zero so that the curve is centered at each message. The variance can be readily estimated from training data.

Conversational context: the second kind of social context is the conversational context, which is constructed from name mentions. As pointed out by previous linguistic studies of discourse, especially analysis of multi-party conversation (ONeill et al., 2003), one key difference between multi-party conversation and typical two-party conversation is the frequency with which participants mention each others' names. Name mentioning is hypothesized as a strategy for participants to compensate for the lack of cues normally present in face-to-face dialogue (ONeill et al., 2003; Elsner et al., 2008). Although infrequent, name mentions (such as Azzie's comments to Ricardo in Figure 1) provide a means for linking two speakers and their messages.

The conversational context of m, $C_C(m)$, is defined to be the set of all messages written by people whose names are mentioned in *any* of a_m's messages (where a_m is the author of m), or who mention a_m in their messages. Let M_a denote all messages written by author a. The conversational context of m is:

$$\begin{aligned} C_C(m) &= \{\forall a\ M_a | mention(a_m, a)\} \\ &\cup \{\forall a\ M_a | mention(a, a_m)\} \end{aligned}$$

where $mention(a_m, a) = true$ if author a_m mentions a in *any* of a_m's messages. $Mention(a, a_m)$ is similarly defined.

Discussion: From the definition, m_j is included in m_i's conversational context if the author of m_i men-

[1] Gaussian kernels shown for illustration purpose in Figure 2 are un-normalized.

tions the author of m_j in any of m_i's messages, or vice versa. For instance, the conversational context for Ricardo's message (19034 Ricardo) in Figure 1 includes the messages from Josephina (18980 Josephina) due to the mentioning of Josephina in his message. However, it may well be the case that m_i does *not* contain any name mentions, e.g. Ricardo's message to Azzie (18939 Ricardo). In this case, if Ricardo is being mentioned by another author (here Azzie asks Ricardo a question by starting with his name in 18939 Azzie), message (18939 Ricardo)'s conversational context will contain *all* of Azzie's messages (18911 and 18970 Azzie) according to the above definition. This intuitively captures the implicit question-answer patterns in conversational speech: Ricardo's subsequent answer is a response to Azzie's comments, hence they are in each other's conversational context.

Our definition also accounts for another source of implicit context. In interactive conversations name mention is a tool for getting people's attention and starting a conversation. Once a participant a_i establishes a conversation with a_j (such that a_i may mention a_j's name in an initial message m_p to a_j), a_i may stop mentioning a_j's name in subsequent messages (m_q) to a_j. This is illustrated in Ricardo's last message to Josephina in Figure 1. Our definition accounts for the *conversation continuity* between a_j and a_i by including messages from a_j in the conversational context of subsequent messages m_q from a_i (note m_q may or may not mention a_j). For instance, message 19045 Ricardo continues the conversation with Josephina from 19034 Ricardo, message 19045 Ricardo thus has Josephina's messages as part of its conversational context.

In general, a person can participate in multiple conversations over time, but as time goes on the topic of interest may shift and the person may start talking to other people. So the messages in the conversational context of m_i due to earlier discussions with other people should be assigned a lower confidence value for m_i. For example, five hours later Ricardo may still be active, but it is unlikely he still chats with Josephina on the same topic, so the earlier messages by Josephina should receive a small confidence value in the conversational context of Ricardo's later messages. We illustrate this idea in Figure 2(ii). It shows the probability that message m_j, where $m_j \in C_C(m_i)$, belongs to the same thread as m_i, given their time difference $t_j - t_i$. This is encoded with a normal probability distribution, $N(\mu_c, \sigma_c)$ where $\mu_c = 0$ and variance is estimated from training data. Let $d = t_j - t_i$, the probability they are topically related given $m_j \in C_C(m_i)$ is:

$$P_c(d) = \frac{1}{\sigma_c\sqrt{2\pi}} e^{-\frac{d^2}{2\sigma_c^2}}$$

Temporal context: temporal context for message m, $C_T(m)$, refers to all other messages:

$$C_T(m) = M \setminus m$$

where M denotes the entire set of messages. The intuition is that nearby messages to m can provide further evidence to the semantics of m. This is illustrated in Figure 2(iii). From the viewpoint of document smoothing, this can also be regarded as using temporally nearby messages to smooth the representation of m. So given m_i, we again model its temporal context by fitting a normal probability distribution $N(\mu_t, \sigma_t)$, so that if $m_j \in C_T(m_i)$ and $d = t_j - t_i$, the probability that m_j is topically related to m_i is:

$$P_t(d) = \frac{1}{\sigma_t\sqrt{2\pi}} e^{-\frac{d^2}{2\sigma_t^2}}$$

3.2.2 Constructing Expanded Messages

We have shown how to use the social and temporal aspects of conversational text to identify and model the contexts of each message, and how to assign confidence values to messages in its contexts. We now show how to use a message's contexts and their associated messages to probabilistically expand the given message. We hypothesize that the expanded message provides a more accurate message representation and that this improved representation can lead to improved accuracy for conversation disentanglement. We will test this hypothesis in the experiment section.

Each message m is represented as a vector of estimated term counts. We expand m using the normalized messages in its contexts. For the expanded message m' of m we estimate the term counts as a linear mixture of term counts from each message in each context:

$$\begin{aligned} c(w,m') &= \alpha c(w,m) + (1-\alpha)\{ \\ & \lambda_C \sum_{m_j \in C_C(m)} P_c(d_{ji}) \times c(w,m_j) \\ &+ \lambda_A \sum_{m_j \in C_A(m)} P_a(d_{ji}) \times c(w,m_j) \\ &+ \lambda_T \sum_{m_j \in C_T(m)} P_t(d_{ji}) \times c(w,m_j)\} \end{aligned}$$

These parameter values are tuned on training data: α controls how much relative weight we give to lexical content of m (0.45 in our experiments), and λ_C, λ_A and λ_T are the relative weights assigned to the conversational, author and temporal contexts (0.6, 0.3, and 0.1 in our experiments, respectively). A context with large variance in its normal density graph should receive a small λ value. This is because a large variance in context k implies more uncertainty on a message m_j being topically related to m while m_j is in the context k of m. In Figure 2, the conversational context (Figure 2(ii)) has the minimum variance among all contexts, hence, it is more accurate for linking messages related in topic and it is assigned a higher λ value (0.6), while the temporal context has the lowest λ value (0.1). Finally, for a message m_j in context k of m_i, $P_k(d_{ji})$ indicates how strongly we believe m_j is topically related to m_i, given their time difference d_{ji}.

Because of the exponential decays of the normal densities that model contexts k, messages in a context will contribute differentially to m_i. Temporally distant messages will have a very low density.

3.3 Single-Pass Clustering

The expanded messages are the basic elements for clustering. The cosine is used to measure similarity:

$$sim(m_i, m_j) = \sum_w \frac{c(w,m_i)c(w,m_j)}{\|m_i\|\|m_j\|}$$

Single-pass clustering is then performed: treat the first message as a single-message cluster T; for each remaining message m compute $\forall T$:

$$sim(m,T) = max_{m_i \in T} sim(m_i, m)$$

For the thread T that maximizes $sim(m,T)$, if $sim(m,T) > t_{sim}$, where t_{sim} is a threshold (0.7 in our experiments) empirically estimated from training data, add m to T; else, start a new cluster containing only m. The time complexity of this algorithm is $O(n^2)$, which is tractable for problems of moderate size.

	Min	**Mean**	**Max**
Number of Conversations	50.00	81.33	128.00
Avg. Conv. Length	6.20	10.60	16.00
Avg. Conv. Density	2.53	2.75	2.92

Table 1: Statistics on the IRC chat transcript data (Elsner et al., 2008). The reported values are based on annotations from six different annotations for the 800 lines of chat transcript.

4 Experiments

The collection used in the experiments consists of real text streams produced in Internet Relay Chat, created by (Elsner et al., 2008) and annotated independently by six annotators. As an upper (human) baseline for each of the three measures reported below, we report the average agreement between all pairs of annotators (i.e., treating one annotator as truth and another as a "system"). For our experiment results, we report the average across all annotators of the agreement between our system and each annotator.

The test collection also contains both a development set and an evaluation set. We used the development set to approximate the normal densities used in our context models and the evaluation set to obtain the results reported below. Some statistics for the 800 annotated messages in the chat transcript of the evaluation collection are shown in Table 1. As that table shows, the average number of active conversation at a given time is 2.75, which makes thread detection a non-trivial task.

4.1 Evaluation Measures

We conduct comparisons using three commonly used evaluation measures for the thread detection task. As a measure of the systems ability to group related messages we report the F-measure (Shen et al., 2006):

$$F = \sum_i \frac{n_i}{n} max_j(F(i,j))$$

where i is a ground-truth conversation with length n_i, and n is the length of entire transcript. $F(i,j)$ is the harmonic mean of recall (fraction of the messages in the i also present in j) and precision (fraction of messages in j also present in i), and F is a weighted sum over all ground-truth conversations (i.e., F is microaveraged).

Two other evaluation measures are "one-to-one accuracy" and "local agreement" (Elsner et al., 2008). "One-to-one accuracy" measures how well we extract whole conversations intact (e.g., as might be required for summarization). It is computed by finding the max-weight bipartite matching between the set of detected threads and the set of real threads, where weight is defined in terms of percentage overlaps for each ground truth and detected thread pair.

Some applications (e.g., real-time monitoring) may not require that we look at entire conversations ar once; in this case a "*local agreement*" measure might make more sense. "loc_3" between system and human annotations as the average (over all possible sets of three consecutive messages) of whether those 3 consecutive messages are assigned consistently by the ground truth and the system. For example, if both the ground truth and the system cluster the first and third messages together and place the second message in a different cluster, then agreement would be recorded.

4.2 Methods Used in Comparison

We compare with the following methods:

Elsner et al. 2008 (best previously known technique): Message similarity is computed with lexical and discourse features, but without document expansion.

Blocks of k: Every consecutive group of k messages is a conversation.

Pause of k: Every pause of k seconds or more separate two conversations.

Speaker: Each speaker's messages are treated as a single conversation.

All different: Each utterance is a separate thread.

All same: The entire transcript is one conversation.

4.3 Results

Figure 3 compares the effectiveness of different schemes in terms of the F measure. We show results from the best baseline, Elsner and our technique (which we call the Context model). The average F between human annotators is shown with the dotted line at 0.55; we would expect this to be an upper bound for any model. Our method substantially outperforms the other methods, with a 24% improvement over Elsner and 48% improvement over the best baseline (speaker). Viewed another way, our system achieves 98% of human performance, while Elsner and the best baseline achieve 79% and 66% of that bound, respectively. From this, we can conclude that our Context model is quite effective at clustering messages from same conversation together.

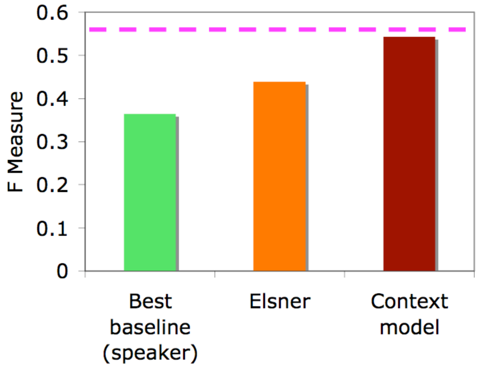


Figure 3: F measure. The dotted line represents inter-annotator agreement.

To illustrate the impact of conversation length, we binned the lengths of ground-truth conversations from a single assessor into bins of size 5 (i.e., 3–7 messages, 8–12 messages, . . .; there were no ground truth bins of size 1 or 2). Figure 4 plots the approximated microaveraged F at the center value of each bin (i.e., the F for each ground truth cluster, scaled by the number of messages in the cluster). These fine-grained values provide insight into the contribution of conversations of different sizes to the overall microaveraged F. The Context model performs well for every conversation length, but particularly so for conversations containing 35 or more messages as shown by the widened gap in that region. Long conversations usually have richer social and temporal contexts for each message. The context model can benefit more from drawing evidences from these sources and using them to expand the message, thus makes it possible to group messages of the same

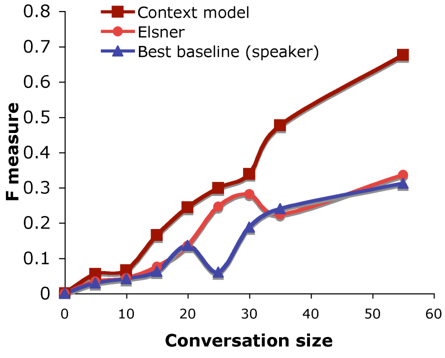


Figure 4: Dependence of F on ground-truth conversation size, in number of messages.

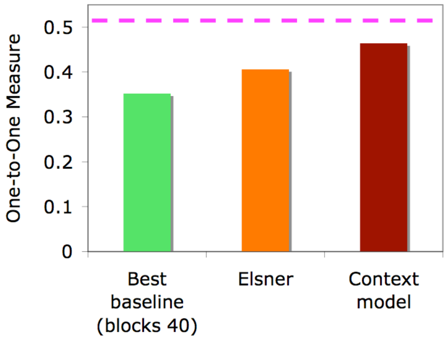


Figure 5: One-to-one measure. The dotted line represents inter-annotator agreement.

conversation together. The other two methods that ignore contextual properties do not do well in comparison.

To measure how well we extract whole conversations intact, Figure 5 shows the results in terms of the one-to-one measure, where each real conversation is matched up with a distinct detected conversation thread. It is computed by max-weight bipartite matching such that the total message overlap is maximized between the sets of detected threads and real threads. The average by this measure between human annotators is 0.53. In this case, the proposed context model achieves an 14% increase over Elsner and 32% increase over the best baseline, and it is within 88% of human performance. This fairly clearly indicates that our Context model can disentangle interleaved conversations relatively well.

Finally, Figure 6 presents the results for "local-3" to evaluate the system's ability to do local annotations. The difference between the best baseline and maximum upper bound is small, implying limited room for potential improvement by any non-baseline techniques. Our result again compares favorably with the previously reported result and the best baseline, although with a smaller margin of 20% over the best baseline and 3% over Elsner as a result of the relatively high baseline for this measure.

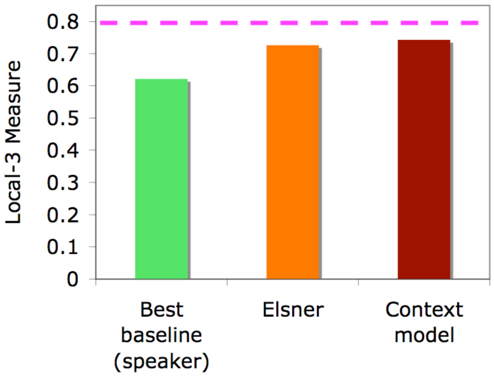


Figure 6: Local-3 measure. The dotted line represents inter-annotator agreement.

5 Conclusion and Future Work

We have presented an approach that exploits contextual properties to probabilistically expand each message to provide a more accurate message representation for dynamic conversations. It is a general approach and can be applied to the representation of non-chat data that exhibits temporal and social correlations as well. For conversation disentanglement, it outperforms the best previously known technique. Our work raises three important questions: (1) to what extent is the single test collection that we have used representative of the broad range of "text chat" applications?, (2) to what extent do the measures we have reported correlate to effective performance of downstream tasks such as summarization or automated response?, and (3) can we re-conceptualize the formalized problem in a way that would result in greater inter-annotator agreement, and hence provide scope for further refinements in our technique. These problems will be the focus of our future work.

References

Micha Elsner and Eugene Charniak. 2008. You talking to me? A Corpus and Algorithm for Conversation Disentanglement. In *ACL 2008: Proceedings of the 46th Annual Meeting on Association for Computational Linguistics,* pages 834-842, Columbus, OH, USA. Association for Computational Linguistics.

Dou Shen, Qiang Yang, Jian-Tao Sun, and Zheng Chen. 2006. Thread Detection in Dynamic Text Message Streams. In *SIGIR 2006: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval,* pages 35-42, Seattle, WA, USA. Association for Computing Machinery.

Yi-Chia Wang, Mahesh Joshi, William Cohen, and Carolyn Rose. 2008. Recovering Implicit Thread Structure in Newsgroup Style Conversations. In *ICWSM 2008: Proceedings of the 2nd International Conference on Weblogs and Social Media,* pages 152-160, Seattle, WA, USA. Association for the Advancement of Artificial Intelligence.

Tao Tao, Xuanhui Wang, Qiaozhu Mei, and ChengXiang Zhai. 2006. Language Model Information Retrieval with Document Expansion. In *HLT-NAACL 2006: Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL,* pages 407-414, New York, NY, USA. Association for Computational Linguistics.

Oren Kurland and Lillian Lee. 2004. Corpus Structure, Language Models, and AdHoc Information Retrieval. In *SIGIR 2004: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval,* pages 194-201, Sheffield, UK. Association for Computing Machinery.

Xiaoyong Liu and W Croft. 2004. Cluster-based Retrieval Using Language Models. In *SIGIR 2004: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval,* pages 186-193, Sheffield, UK. Association for Computing Machinery.

Amit Singhal and Fernando Pereira. 1999. Document Expansion for Speech Retrieval. In *SIGIR 1999: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval,* pages 34-41, Berkeley, CA, USA. Association for Computing Machinery.

Xiang Ji and Hongyuan Zha 2003. Domain-Independent Text Segmentation using Anisotropic Diffusion and Dynamic Programming. In *SIGIR 2003: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval,* pages 322-329, Toronto, Canada. Association for Computing Machinery.

Michel Galley, Kathleen McKeown, Eric Lussier, and Hongyan Jing. 2003. Discourse Segmentation of Multi-Party Conversation. In *ACL 2003: Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics,* pages 562-569, Sapporo, Japan. Association for Computational Linguistics.

Jacob Eisenstein and Regina Barzilay. 2008. Bayesian Unsupervised Topic Segmentation. In *EMNLP 2008: Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing,* pages 334-343, Honolulu, Hawaii, USA. Association for Computational Linguistics.

Igor Malioutov and Regina Barzilay 2006. Minimum-Cut Model for Spoken Lecture Segmentation. In *ACL 2006: Proceedings of the 44rd Annual Meeting of the Association for Computational Linguistics,* pages 25-32, Sydney, Australia. Association for Computational Linguistics.

Igor Malioutov, Alex Park, Regina Barzilay, and James Glass. 2007. Making Sense of Sound: Unsupervised Topic Segmentation over Acoustic Input. In *ACL 2007: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics,* pages 504-511, Prague, Czech Republic. Association for Computational Linguistics.

Jen-Yuan Yeh and Aaron Harnly. 2006. Email Thread Reassembly Using Similarity Matching. In *CEAS 2006: The 3rd Conference on Email and Anti-Spam,* pages 64-71, Mountain View, CA, USA.

Jacki ONeill and David Martin. 2003. Text Chat in Action. In *ACM SIGGROUP 2003: Proceedings of the 2003 International ACM SIGGROUP Conference on Supporting Group Work,* pages 40-49, New York, NY, USA. ACM Press.

Gerard Salton. 1989. Automatic Text Processing: the Transformation, Analysis and Retrieval of Information by Computer. *Addison-Wesley Longman Publishing Co., Inc.,* Boston, MA, USA, 1989.

Gina-Anne Levow, Douglas Oard, and Philip Resnik. 2005. Dictionary-based techniques for cross-language information retrieval. In *Information Processing and Management Special Issue: Cross-Language Information Retrieval,* 41(3): 523-547.