

You talking to me? A Corpus and Algorithm for Conversation Disentanglement

Micha Elsner and Eugene Charniak

Brown Laboratory for Linguistic Information Processing (BLLIP)

Brown University

Providence, RI 02912

{melsner, ec}@cs.brown.edu

Abstract

When multiple conversations occur simultaneously, a listener must decide which conversation each utterance is part of in order to interpret and respond to it appropriately. We refer to this task as disentanglement. We present a corpus of Internet Relay Chat (IRC) dialogue in which the various conversations have been manually disentangled, and evaluate annotator reliability. This is, to our knowledge, the first such corpus for internet chat. We propose a graph-theoretic model for disentanglement, using discourse-based features which have not been previously applied to this task. The model's predicted disentanglements are highly correlated with manual annotations.

1 Motivation

Simultaneous conversations seem to arise naturally in both informal social interactions and multi-party typed chat. Aoki et al. (2006)'s study of voice conversations among 8-10 people found an average of 1.76 conversations (floors) active at a time, and a maximum of four. In our chat corpus, the average is even higher, at 2.75. The typical conversation, therefore, is one which is interrupted— frequently.

Disentanglement is the clustering task of dividing a transcript into a set of distinct conversations. It is an essential prerequisite for any kind of higher-level dialogue analysis: for instance, consider the multi-party exchange in figure 1.

Contextually, it is clear that this corresponds to two conversations, and Felicia's¹ response “excel-

(Chanel) Felicia: google works :)
(Gale) Arlie: you guys have never worked in a factory before have you
(Gale) Arlie: there's some real unethical stuff that goes on
(Regine) hands Chanel a trophy
(Arlie) Gale, of course ... thats how they make money
(Gale) and people lose limbs or get killed
(Felicia) excellent

Figure 1: Some (abridged) conversation from our corpus.

lent” is intended for Chanel and Regine. A straightforward reading of the transcript, however, might interpret it as a response to Gale's statement immediately preceding.

Humans are adept at disentanglement, even in complicated environments like crowded cocktail parties or chat rooms; in order to perform this task, they must maintain a complex mental representation of the ongoing discourse. Moreover, they adapt their utterances to some degree to make the task easier (O'Neill and Martin, 2003), which suggests that disentanglement is in some sense a “difficult” discourse task.

Disentanglement has two practical applications. One is the analysis of pre-recorded transcripts in order to extract some kind of information, such as question-answer pairs or summaries. These tasks should probably take as input each separate conversation, rather than the entire transcript. Another

¹Real user nicknames are replaced with randomly selected

identifiers for ethical reasons.

application is as part of a user-interface system for active participants in the chat, in which users target a conversation of interest which is then highlighted for them. Aoki et al. (2003) created such a system for speech, which users generally preferred to a conventional system—when the disentanglement worked!

Previous attempts to solve the problem (Aoki et al., 2006; Aoki et al., 2003; Camtepe et al., 2005; Acar et al., 2005) have several flaws. They cluster speakers, not utterances, and so fail when speakers move from one conversation to another. Their features are mostly time gaps between one utterance and another, without effective use of utterance content. Moreover, there is no framework for a principled comparison of results: there are no reliable annotation schemes, no standard corpora, and no agreed-upon metrics.

We attempt to remedy these problems. We present a new corpus of manually annotated chat room data and evaluate annotator reliability. We give a set of metrics describing structural similarity both locally and globally. We propose a model which uses discourse structure and utterance contents in addition to time gaps. It partitions a chat transcript into distinct conversations, and its output is highly correlated with human annotations.

2 Related Work

Two threads of research are direct attempts to solve the disentanglement problem: Aoki et al. (2006), Aoki et al. (2003) for speech and Camtepe et al. (2005), Acar et al. (2005) for chat. We discuss their approaches below. However, we should emphasize that we cannot compare our results directly with theirs, because none of these studies publish results on human-annotated data. Although Aoki et al. (2006) construct an annotated speech corpus, they give no results for model performance, only user satisfaction with their conversational system. Camtepe et al. (2005) and Acar et al. (2005) do give performance results, but only on synthetic data.

All of the previous approaches treat the problem as one of clustering speakers, rather than utterances. That is, they assume that during the window over which the system operates, a particular speaker is engaging in only one conversation. Camtepe et al. (2005) assume this is true throughout the entire tran-

script; real speakers, by contrast, often participate in many conversations, sequentially or sometimes even simultaneously. Aoki et al. (2003) analyze each thirty-second segment of the transcript separately. This makes the single-conversation restriction somewhat less severe, but has the disadvantage of ignoring all events which occur outside the segment.

Acar et al. (2005) attempt to deal with this problem by using a fuzzy algorithm to cluster speakers; this assigns each speaker a distribution over conversations rather than a hard assignment. However, the algorithm still deals with speakers rather than utterances, and cannot determine which conversation any particular utterance is part of.

Another problem with these approaches is the information used for clustering. Aoki et al. (2003) and Camtepe et al. (2005) detect the arrival times of messages, and use them to construct an affinity graph between participants by detecting turn-taking behavior among pairs of speakers. (Turn-taking is typified by short pauses between utterances; speakers aim neither to interrupt nor leave long gaps.) Aoki et al. (2006) find that turn-taking on its own is inadequate. They motivate a richer feature set, which, however, does not yet appear to be implemented. Acar et al. (2005) adds word repetition to their feature set. However, their approach deals with all word repetitions on an equal basis, and so degrades quickly in the presence of *noise words* (their term for words which shared across conversations) to almost complete failure when only 1/2 of the words are shared.

To motivate our own approach, we examine some linguistic studies of discourse, especially analysis of multi-party conversation. O’Neill and Martin (2003) point out several ways in which multi-party text chat differs from typical two-party conversation. One key difference is the frequency with which participants mention each others’ names. They hypothesize that mentioning is a strategy which participants use to make disentanglement easier, compensating for the lack of cues normally present in face-to-face dialogue. Mentions (such as Gale’s comments to Arlie in figure 1) are very common in our corpus, occurring in 36% of comments, and provide a useful feature.

Another key difference is that participants may create a new conversation (floor) at any time, a process which Sacks et al. (1974) calls *schisming*. Dur-

ing a schism, a new conversation is formed, not necessarily because of a shift in the topic, but because certain participants have refocused their attention onto each other, and away from whoever held the floor in the parent conversation.

Despite these differences, there are still strong similarities between chat and other conversations such as meetings. Our feature set incorporates information which has proven useful in meeting segmentation (Galley et al., 2003) and the task of detecting addressees of a specific utterance in a meeting (Jovanovic et al., 2006). These include word repetitions, utterance topic, and *cue words* which can indicate the bounds of a segment.

3 Dataset

Our dataset is recorded from the IRC (Internet Relay Chat) channel `##LINUX` at *freenode.net*, using the freely-available *gaim* client. `##LINUX` is an unofficial tech support line for the Linux operating system, selected because it is one of the most active chat rooms on freenode, leading to many simultaneous conversations, and because its content is typically inoffensive. Although it is notionally intended only for tech support, it includes large amounts of social chat as well, such as the conversation about factory work in the example above (figure 1).

The entire dataset contains 52:18 hours of chat, but we devote most of our attention to three annotated sections: development (706 utterances; 2:06 hr) and test (800 utts.; 1:39 hr) plus a short pilot section on which we tested our annotation system (359 utts.; 0:58 hr).

3.1 Annotation

Our annotators were seven university students with at least some familiarity with the Linux OS, although in some cases very slight. Annotation of the test dataset typically took them about two hours. In all, we produced six annotations of the test set².

Our annotation scheme marks each utterance as part of a single conversation. Annotators are instructed to create as many, or as few conversations as they need to describe the data. Our instructions state that a conversation can be between any number of

people, and that, “We mean conversation in the typical sense: a discussion in which the participants are all reacting and paying attention to one another. . . it should be clear that the comments inside a conversation fit together.” The annotation system itself is a simple Java program with a graphical interface, intended to appear somewhat similar to a typical chat client. Each speaker’s name is displayed in a different color, and the system displays the elapsed time between comments, marking especially long pauses in red. Annotators group sentences into conversations by clicking and dragging them onto each other.

3.2 Metrics

Before discussing the annotations themselves, we will describe the metrics we use to compare different annotations; these measure both how much our annotators agree with each other, and how well our model and various baselines perform. Comparing clusterings with different numbers of clusters is a non-trivial task, and metrics for agreement on supervised classification, such as the κ statistic, are not applicable.

To measure global similarity between annotations, we use *one-to-one accuracy*. This measure describes how well we can extract whole conversations intact, as required for summarization or information extraction. To compute it, we pair up conversations from the two annotations to maximize the total overlap³, then report the percentage of overlap found.

If we intend to monitor or participate in the conversation as it occurs, we will care more about local judgements. The *local agreement* metric counts agreements and disagreements within a context k . We consider a particular utterance: the previous k utterances are each in either the *same* or a *different* conversation. The loc_k score between two annotators is their average agreement on these k same/different judgements, averaged over all utterances. For example, loc_1 counts pairs of adjacent utterances for which two annotations agree.

²One additional annotation was discarded because the annotator misunderstood the task.

	Mean	Max	Min
Conversations	81.33	128	50
Avg. Conv. Length	10.6	16.0	6.2
Avg. Conv. Density	2.75	2.92	2.53
Entropy	4.83	6.18	3.00
1-to-1	52.98	63.50	35.63
loc_3	81.09	86.53	74.75
M-to-1 (by entropy)	86.70	94.13	75.50

Table 1: Statistics on 6 annotations of 800 lines of chat transcript. Inter-annotator agreement metrics (below the line) are calculated between distinct pairs of annotations.

3.3 Discussion

A statistical examination of our data (table 1) shows that that it is eminently suitable for disentanglement: the average number of conversations active at a time is 2.75. Our annotators have high agreement on the local metric (average of 81.1%). On the 1-to-1 metric, they disagree more, with a mean overlap of 53.0% and a maximum of only 63.5%. This level of overlap does indicate a useful degree of reliability, which cannot be achieved with naive heuristics (see section 5). Thus measuring 1-to-1 overlap with our annotations is a reasonable evaluation for computational models. However, we feel that the major source of disagreement is one that can be remedied in future annotation schemes: the specificity of the individual annotations.

To measure the level of detail in an annotation, we use the information-theoretic entropy of the random variable which indicates which conversation an utterance is in. This quantity is non-negative, increasing as the number of conversations grow and their size becomes more balanced. It reaches its maximum, 9.64 bits for this dataset, when each utterance is placed in a separate conversation. In our annotations, it ranges from 3.0 to 6.2. This large variation shows that some annotators are more specific than others, but does not indicate how much they agree on the general structure. To measure this, we introduce the many-to-one accuracy. This measurement is asymmetrical, and maps each of the conversations of the *source* annotation to the single con-

(**Lai**) need money
 (**Astrid**) suggest a paypal fund or similar
 (**Lai**) Azzie [sic; typo for Astrid?]: my shack guy here said paypal too but i have no local bank acct
 (**Felicia**) second’s Azzie’s suggestion
 (**Gale**) we should charge the noobs \$1 per question to [Lai’s] paypal
 (**Felicia**) bingo!
 (**Gale**) we’d have the money in 2 days max
 (**Azzie**) Lai: hrm, Have you tried to set one up?
 (**Arlie**) the federal reserve system conspiracy is keeping you down man
 (**Felicia**) Gale: all ubuntu users .. pay up!
 (**Gale**) and susers pay double
 (**Azzie**) I certainly would make suse users pay.
 (**Hildegard**) triple.
 (**Lai**) Azzie: not since being offline
 (**Felicia**) it doesn’t need to be “in state” either

Figure 2: A schism occurring in our corpus (abridged): not all annotators agree on where the thread about charging for answers to techical questions diverges from the one about setting up Paypal accounts. Either Gale’s or Azzie’s first comment seems to be the schism-inducing utterance.

versation in the *target* with which it has the greatest overlap, then counts the total percentage of overlap. This is not a statistic to be optimized (indeed, optimization is trivial: simply make each utterance in the source into its own conversation), but it can give us some intuition about specificity. In particular, if one subdivides a coarse-grained annotation to make a more specific variant, the many-to-one accuracy from fine to coarse remains 1. When we map high-entropy annotations (fine) to lower ones (coarse), we find high many-to-one accuracy, with a mean of 86%, which implies that the more specific annotations have mostly the same large-scale boundaries as the coarser ones.

By examining the local metric, we can see even more: local correlations are good, at an average of 81.1%. This means that, in the three-sentence window preceding each sentence, the annotators are of-

³This is an example of max-weight bipartite matching, and can be computed optimally using, eg, max-flow. The widely used greedy algorithm is a two-approximation, although we have not found large differences in practice.

ten in agreement. If they recognize subdivisions of a large conversation, these subdivisions tend to be contiguous, not mingled together, which is why they have little impact on the local measure.

We find reasons for the annotators’ disagreement about appropriate levels of detail in the linguistic literature. As mentioned, new conversations often break off from old ones in schisms. Aoki et al. (2006) discuss conversational features associated with schisming and the related process of *affiliation*, by which speakers attach themselves to a conversation. Schisms often branch off from asides or even normal comments (*toss-outs*) within an existing conversation. This means that there is no clear beginning to the new conversation— at the time when it begins, it is not clear that there are two separate floors, and this will not become clear until distinct sets of speakers and patterns of turn-taking are established. Speakers, meanwhile, take time to orient themselves to the new conversation. An example schism is shown in Figure 2.

Our annotation scheme requires annotators to mark each utterance as part of a single conversation, and distinct conversations are not related in any way. If a schism occurs, the annotator is faced with two options: if it seems short, they may view it as a mere digression and label it as part of the parent conversation. If it seems to deserve a place of its own, they will have to separate it from the parent, but this severs the initial comment (an otherwise unremarkable aside) from its context. One or two of the annotators actually remarked that this made the task confusing. Our annotators seem to be either “splitters” or “lumpers”— in other words, each annotator seems to aim for a consistent level of detail, but each one has their own idea of what this level should be.

As a final observation about the dataset, we test the appropriateness of the assumption (used in previous work) that each speaker takes part in only one conversation. In our data, the average speaker takes part in about 3.3 conversations (the actual number varies for each annotator). The more talkative a speaker is, the more conversations they participate in, as shown by a plot of conversations versus utterances (Figure 3). The assumption is not very accurate, especially for speakers with more than 10 utterances.

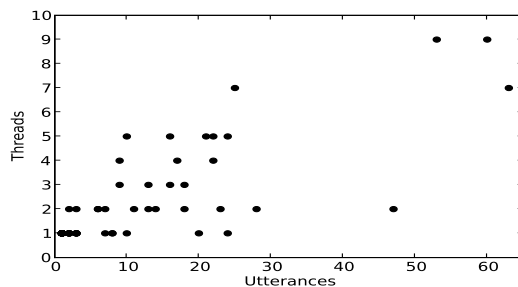


Figure 3: Utterances versus conversations participated in per speaker on development data.

4 Model

Our model for disentanglement fits into the general class of graph partitioning algorithms (Roth and Yih, 2004) which have been used for a variety of tasks in NLP, including the related task of meeting segmentation (Malioutov and Barzilay, 2006). These algorithms operate in two stages: first, a binary classifier marks each pair of items as alike or different, and second, a consistent partition is extracted.⁴

4.1 Classification

We use a maximum-entropy classifier (Daumé III, 2004) to decide whether a pair of utterances x and y are in *same* or *different* conversations. The most likely class is *different*, which occurs 57% of the time in development data. We describe the classifier’s performance in terms of raw accuracy (correct decisions / total), precision and recall of the *same* class, and F-score, the harmonic mean of precision and recall. Our classifier uses several types of features (table 2). The chat-specific features yield the highest accuracy and precision. Discourse and content-based features have poor accuracy on their own (worse than the baseline), since they work best on nearby pairs of utterances, and tend to fail on more distant pairs. Paired with the time gap feature, however, they boost accuracy somewhat and produce substantial gains in recall, encouraging the model to group related utterances together.

The time gap, as discussed above, is the most widely used feature in previous work. We exam-

⁴Our first attempt at this task used a Bayesian generative model. However, we could not define a sharp enough posterior over new sentences, which made the model unstable and overly sensitive to its prior.

Chat-specific (Acc 73: Prec: 73 Rec: 61 F: 66)	
Time	The time between x and y in seconds, bucketed logarithmically.
Speaker	x and y have the same speaker.
Mention	x mentions y (or vice versa), both mention the same name, either mentions any name.
Discourse (Acc 52: Prec: 47 Rec: 77 F: 58)	
Cue words	Either x or y uses a greeting (“hello” &c), an answer (“yes”, “no” &c), or thanks.
Question	Either asks a question (explicitly marked with “?”).
Long	Either is long (> 10 words).
Content (Acc 50: Prec: 45 Rec: 74 F: 56)	
Repeat(i)	The number of words shared between x and y which have unigram probability i , bucketed logarithmically.
Tech	Whether both x and y use technical jargon, neither do, or only one does.
Combined (Acc 75: Prec: 73 Rec: 68 F: 71)	

Table 2: Feature functions with performance on development data.

ine the distribution of pauses between utterances in the same conversation. Our choice of a logarithmic bucketing scheme is intended to capture two characteristics of the distribution (figure 4). The curve has its maximum at 1-3 seconds, and pauses shorter than a second are less common. This reflects turn-taking behavior among participants; participants in the same conversation prefer to wait for each others’ responses before speaking again. On the other hand, the curve is quite heavy-tailed to the right, leading us to bucket long pauses fairly coarsely.

Our discourse-based features model some pair-

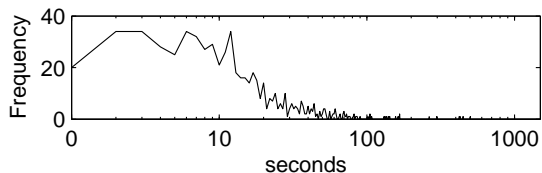


Figure 4: Distribution of pause length (log-scaled) between utterances in the same conversation.

wise relationships: questions followed by answers, short comments reacting to longer ones, greetings at the beginning and thanks at the end.

Word repetition is a key feature in nearly every model for segmentation or coherence, so it is no surprise that it is useful here. We bucket repeated words by their unigram probability⁵ (measured over the entire 52 hours of transcript). The bucketing scheme allows us to deal with “noise words” which are repeated coincidentally.

The point of the repetition feature is of course to detect sentences with similar topics. We also find that sentences with technical content are more likely to be related than non-technical sentences. We label an utterance as technical if it contains a web address, a long string of digits, or a term present in a guide for novice Linux users⁶ but not in a large news corpus (Graff, 1995)⁷. This is a light-weight way to capture one “semantic dimension” or cluster of related words, in a corpus which is not amenable to full LSA or similar techniques. LSA in text corpora yields a better relatedness measure than simple repetition (Foltz et al., 1998), but is ineffective in our corpus because of its wide variety of topics and lack of distinct document boundaries.

Pairs of utterances which are widely separated in the discourse are unlikely to be directly related—even if they are part of the same conversation, the link between them is probably a long chain of intervening utterances. Thus, if we run our classifier on a pair of very distant utterances, we expect it to default to the majority class, which in this case will be *different*, and this will damage our performance in case the two are really part of the same conversation. To deal with this, we run our classifier only on utterances separated by 129 seconds or less. This is the last of our logarithmic buckets in which the classifier has a significant advantage over the majority baseline. For 99.9% of utterances in an ongoing conversation, the previous utterance in that conversation is within this gap, and so the system has a

⁵We discard the 50 most frequent words entirely.

⁶“Introduction to Linux: A Hands-on Guide”. Machtelt Garrels. Edition 1.25 from <http://tldp.org/LDP/intro-linux/html/intro-linux.html>.

⁷Our data came from the LA times, 94-97— helpfully, it predates the current wide coverage of Linux in the mainstream press.

chance of correctly linking the two.

On test data, the classifier has a mean accuracy of 68.2 (averaged over annotations). The mean precision of *same conversation* is 53.3 and the recall is 71.3, with mean F-score of 60. This error rate is high, but the partitioning procedure allows us to recover from some of the errors, since if nearby utterances are grouped correctly, the bad decisions will be outvoted by good ones.

4.2 Partitioning

The next step in the process is to cluster the utterances. We wish to find a set of clusters for which the weighted accuracy of the classifier would be maximal; this is an example of *correlation clustering* (Bansal et al., 2004), which is NP-complete⁸. Finding an exact solution proves to be difficult; the problem has a quadratic number of variables (one for each pair of utterances) and a cubic number of triangle inequality constraints (three for each triplet). With 800 utterances in our test set, even solving the linear program with CPLEX (Ilog, Inc., 2003) is too expensive to be practical.

Although there are a variety of approximations and local searches, we do not wish to investigate partitioning methods in this paper, so we simply use a greedy search. In this algorithm, we assign utterance j by examining all previous utterances i within the classifier’s window, and treating the classifier’s judgement $p_{i,j} - .5$ as a vote for $cluster(i)$. If the maximum vote is greater than 0, we set $cluster(j) = \argmax_c vote_c$. Otherwise j is put in a new cluster. Greedy clustering makes at least a reasonable starting point for further efforts, since it is a natural online algorithm— it assigns each utterance as it arrives, without reference to the future.

At any rate, we should not take our objective function too seriously. Although it is roughly correlated with performance, the high error rate of the classifier makes it unlikely that small changes in objective will mean much. In fact, the objective value of our output solutions are generally higher than those for true so-

⁸We set up the problem by taking the weight of edge i, j as the classifier’s decision $p_{i,j} - .5$. Roth and Yih (2004) use log probabilities as weights. Bansal et al. (2004) propose the log odds ratio $\log(p/(1 - p))$. We are unsure of the relative merit of these approaches.

lutions, which implies we have already reached the limits of what our classifier can tell us.

5 Experiments

We annotate the 800 line test transcript using our system. The annotation obtained has 63 conversations, with mean length 12.70. The average density of conversations is 2.9, and the entropy is 3.79. This places it within the bounds of our human annotations (see table 1), toward the more general end of the spectrum.

As a standard of comparison for our system, we provide results for several baselines— trivial systems which any useful annotation should outperform.

All different Each utterance is a separate conversation.

All same The whole transcript is a single conversation.

Blocks of k Each consecutive group of k utterances is a conversation.

Pause of k Each pause of k seconds or more separates two conversations.

Speaker Each speaker’s utterances are treated as a monologue.

For each particular metric, we calculate the best baseline result among all of these. To find the best block size or pause length, we search over multiples of 5 between 5 and 300. This makes these baselines appear better than they really are, since their performance is optimized with respect to the test data.

Our results, in table 3, are encouraging. On average, annotators agree more with each other than with any artificial annotation, and more with our model than with the baselines. For the 1-to-1 accuracy metric, we cannot claim much beyond these general results. The range of human variation is quite wide, and there are annotators who are closer to baselines than to any other human annotator. As explained earlier, this is because some human annotations are much more specific than others. For very specific annotations, the best baselines are short blocks or pauses. For the most general, marking all utterances the same does very well (although for all other annotations, it is extremely poor).

	Other Annotators	Model	Best Baseline	All Diff	All Same
Mean 1-to-1	52.98	40.62	34.73 (Blocks of 40)	10.16	20.93
Max 1-to-1	63.50	51.12	56.00 (Pause of 65)	16.00	53.50
Min 1-to-1	35.63	33.63	28.62 (Pause of 25)	6.25	7.13
Mean loc_3	81.09	72.75	62.16 (Speaker)	52.93	47.07
Max loc_3	86.53	75.16	69.05 (Speaker)	62.15	57.47
Min loc_3	74.75	70.47	54.37 (Speaker)	42.53	37.85

Table 3: Metric values between proposed annotations and human annotations. Model scores typically fall between inter-annotator agreement and baseline performance.

For the local metric, the results are much clearer. There is no overlap in the ranges; for every test annotation, agreement is highest with other annotator, then our model and finally the baselines. The most competitive baseline is one conversation per speaker, which makes sense, since if a speaker makes two comments in a four-utterance window, they are very likely to be related.

The name mention features are critical for our model’s performance. Without this feature, the classifier’s development F-score drops from 71 to 56. The disentanglement system’s test performance decreases proportionally; mean 1-to-1 falls to 36.08, and mean loc_3 to 63.00, essentially baseline performance. On the other hand, mentions are not sufficient; with only name mention and time gap features, mean 1-to-1 is 38.54 and loc_3 is 67.14. For some utterances, of course, name mentions provide the only reasonable clue to the correct decision, which is why humans mention names in the first place. But our system is probably overly dependent on them, since they are very reliable compared to our other features.

6 Future Work

Although our annotators are reasonably reliable, it seems clear that they think of conversations as a hierarchy, with digressions and schisms. We are interested to see an annotation protocol which more closely follows human intuition and explicitly includes these kinds of relationships.

We are also interested to see how well this feature set performs on speech data, as in (Aoki et al., 2003). Spoken conversation is more natural than text chat, but when participants are not face-to-face, disentanglement remains a problem. On the other hand, spoken dialogue contains new sources of information,

such as prosody. Turn-taking behavior is also more distinct, which makes the task easier, but according to (Aoki et al., 2006), it is certainly not sufficient.

Improving the current model will definitely require better features for the classifier. However, we also left the issue of partitioning nearly completely unexplored. If the classifier can indeed be improved, we expect the impact of search errors to increase. Another issue is that human users may prefer more or less specific annotations than our model provides. We have observed that we can produce lower or higher-entropy annotations by changing the classifier’s bias to label more edges same or different. But we do not yet know whether this corresponds with human judgements, or merely introduces errors.

7 Conclusion

This work provides a corpus of annotated data for chat disentanglement, which, along with our proposed metrics, should allow future researchers to evaluate and compare their results quantitatively⁹. Our annotations are consistent with one another, especially with respect to local agreement. We show that features based on discourse patterns and the content of utterances are helpful in disentanglement. The model we present can outperform a variety of baselines.

Acknowledgements

Our thanks to Suman Karumuri, Steve Sloman, Matt Lease, David McClosky, 7 test annotators, 3 pilot annotators, 3 anonymous reviewers and the NSF PIRE grant.

⁹Code and data for this project will be available at <http://cs.brown.edu/people/melsner>.

References

- Evrin Acar, Seyit Ahmet Camtepe, Mukkai S. Krishnamoorthy, and Blent Yener. 2005. Modeling and multiway analysis of chatroom tensors. In Paul B. Kantor, Gheorghe Muresan, Fred Roberts, Daniel Dajun Zeng, Fei-Yue Wang, Hsinchun Chen, and Ralph C. Merkle, editors, *ISI*, volume 3495 of *Lecture Notes in Computer Science*, pages 256–268. Springer.
- Paul M. Aoki, Matthew Romaine, Margaret H. Szymanski, James D. Thornton, Daniel Wilson, and Allison Woodruff. 2003. The mad hatter’s cocktail party: a social mobile audio space supporting multiple simultaneous conversations. In *CHI ’03: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 425–432, New York, NY, USA. ACM Press.
- Paul M. Aoki, Margaret H. Szymanski, Luke D. Plurkowski, James D. Thornton, Allison Woodruff, and Weillie Yi. 2006. Where’s the “party” in “multi-party”? analyzing the structure of small-group social talk. In *CSCW ’06: Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*, pages 393–402, New York, NY, USA. ACM Press.
- Nikhil Bansal, Avrim Blum, and Shuchi Chawla. 2004. Correlation clustering. *Machine Learning*, 56(1-3):89–113.
- Seyit Ahmet Camtepe, Mark K. Goldberg, Malik Magdon-Ismael, and Mukkai Krishnamoorthy. 2005. Detecting conversing groups of chatters: a model, algorithms, and tests. In *IADIS AC*, pages 89–96.
- Hal Daumé III. 2004. Notes on CG and LM-BFGS optimization of logistic regression. Paper available at <http://pub.hal3.name#daume04cg-bfgs>, implementation available at <http://hal3.name/megam/>, August.
- Peter Foltz, Walter Kintsch, and Thomas Landauer. 1998. The measurement of textual coherence with latent semantic analysis. *Discourse Processes*, 25(2&3):285–307.
- Michel Galley, Kathleen McKeown, Eric Fosler-Lussier, and Hongyan Jing. 2003. Discourse segmentation of multi-party conversation. In *ACL ’03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 562–569, Morristown, NJ, USA. Association for Computational Linguistics.
- David Graff. 1995. *North American News Text Corpus*. Linguistic Data Consortium. LDC95T21.
- Ilog, Inc. 2003. Cplex solver.
- Natasa Jovanovic, Rieks op den Akker, and Anton Nijholt. 2006. Addressee identification in face-to-face meetings. In *EACL*. The Association for Computer Linguistics.
- Igor Malioutov and Regina Barzilay. 2006. Minimum cut model for spoken lecture segmentation. In *ACL*. The Association for Computer Linguistics.
- Jacki O’Neill and David Martin. 2003. Text chat in action. In *GROUP ’03: Proceedings of the 2003 international ACM SIGGROUP conference on Supporting group work*, pages 40–49, New York, NY, USA. ACM Press.
- Dan Roth and Wen-tau Yih. 2004. A linear programming formulation for global inference in natural language tasks. In *Proceedings of CoNLL-2004*, pages 1–8. Boston, MA, USA.
- Harvey Sacks, Emanuel A. Schegloff, and Gail Jefferson. 1974. A simplest systematics for the organization of turn-taking for conversation. *Language*, 50(4):696–735.