Disentangling utterances and recovering coherent multi party distinct conversations

Gibson O. Ikoro
School of Electronic Engineering and Computer Science, Queen Mary University of London
g.o.ikoro@qmul.ac.uk

Raul Mondragon
School of Electronic Engineering and Computer Science, Queen Mary University of London
r.j.mondragon@qmul.ac.uk

Graham White
School of Electronic Engineering and Computer Science, Queen Mary University of London
graham.white@qmul.ac.uk

Abstract

Automatic separation of interposed sequence of utterances into distinct conversations is an essential prerequisite for any kind of higher-level dialogue analysis. Unlike most models that involve highly computational intensive methods such as clustering techniques, our proposed approach uses a simple and efficient sequential thread detection method which is less computational intensive. It uses the waiting time (time gap between the current speaker and the next speaker), similarity between utterances, turn-taking and participant-based features.

```
 1
 2   40:29 A→(B):grins I think it’s the proxy servers
 3               called Kevin and Perry that need kicking!
 4   40:55 B→(A):what happened last night..the whole
 5               fecking lot of it got or needed a kicking!
 6   41:13 C→(D,E,F,G,H): lsaysl cH kissing bandit...l
 7   41:45 H→(I,J):Kissing bandits are  predators and
 8                 should not be tolerated
 9   41:46 A→(B): it was a Janet router that went again,
10                second tie in a week that one has died for
11   42:08 C→(D,E,F,G,H):lsaysl cYou’re just jealous he
12                took your job
13   42:16 A→(B):grins...janet is the nae of the network
14               that the universities and schools are on.
15               A router is soething that forwards on
16               inforation to the correct coputer, so
17               when you send your essage one TCZ, lo
18   42:21 H→(I,J): And I haven’t gotten any action since
19
```

Figure 1: Sample of conversation from our corpus.

1 Motivation:Disentanglement problem

Chat rooms are where people can meet each other to chat on the internet (Davies, 2010). Chat room logs are not a single continuous conversation of two or a group of people at a time rather each time widow is a sequence of frequently broken utterances (Elsner and Charniak, 2008). A typical conversation, therefore, does not form an adjacent segment of the chat-room transcript, but sequences of frequently broken utterances due to interposed utterances from other conversations. For example, consider the time window (40:29 - 42:21) mm:ss in Fig 1 the utterances in line 2, 4, 9 and 13 show an ongoing conversation which is being interposed by the utterances in line 6, 7, and 11. However, a typical chat room log that consists of millions of utterances and conversations which interpose each other will be difficult to separate into distinct conversations using traditional methods.

Another challenge in disentangling chat log is schism, a process where some participants create a new conversation which is different from the already existing one. This often occurs when two or more users change their attention to themselves and away from whoever held the floor (the current speaker) in the parent conversation (Elsner and Charniak, 2008). Disentanglement or thread detection is a task that extracts the different interposed utterances in a chat log and separates them into distinct conversations. If we want our statistical methods to be useful for conversational analysis, we have to disentangle the logs. It is only when we have disentangled the logs, that we can apply other methods to find out about structures like question-answer pairs (Purver, 2011).

However, any form of automated semantic analysis is tedious and likely to be unsuccessful on account of the extremely unstructured lexicon used (Camtepe et al., 2005). Hence there is need for a model that combines both the pragmatic information and statistical approach.

2 Related work

Thread disentanglement is most commonly studied using clustering methods (Uthus and Aha, 2013; Elsner and Charniak, 2008). In a recent study, (Elsner and Charniak, 2011) employed coherence models to investigate chat disentanglement. They validated their models using recorded telephone conversations for thread disentanglement. In their method they used tabu search method to search for a solution to this problem; this involves conducting two sets of experiments with each chat corpus. The first is to disentangle single messages and the second disentangles the entire chat log.

Another interesting method on chat disentanglement is described in (Elsner and Schudy, 2009) and (Elsner and Charniak, 2010). Their work utilized correlation clustering for thread detection. The method involves searching for group of clusters that maximizes the degree of similarities between pairs within a cluster and maximizes the degree of dissimilarity among pairs across clusters. The two models employ maximum-entropy classifier to determine if two messages belong to the same conversation. Elsner and Schudy employ two methods: greedy method and local search method for the NP-hard problem of searching the best solution for correlation clustering while Elsner and Charniak, employed voting schema for correlation clustering (Uthus and Aha, 2013).

In another recent work, Mayfield et al. (Mayfield et al., 2012) utilized a two-pass method for thread detection. In the first pass, the method labels sentences using a negotiation framework. After the labelling process, a single-pass clustering algorithm is used to detect sequences.

Our approach is unique in the sense that it does not involve any conventional clustering method or other highly computational intensive techniques which may lead to depreciation in the accuracy of results. Before discussing our proposed techniques, we will introduce the dataset.

3 Description of the dataset

Walford is a text-based online social community that was set up more than a decade ago (Healey et al., 2008). It has roughly 2446 regular users. It is a corpus that contains 24040 hours (26/11/2001 - 24/08/2004) of chat. For each communication, the following data is recorded: the time, the originator, the originators location, the recipient(s) and their location.

In Walford, the participants can construct a friend-list. Walford has a tool that permits users to send direct message to all the members in their friend list who are online at the same time (Healey et al., 2008). The ability to reach everyone in one's friend list simultaneously helps Walford users to perform group chat.

4 Methodology

The proposed approach for the ongoing project uses a simple and efficient method for chat disentanglement. The algorithm involves three-pass process. In the first pass, the algorithm predicts the occurrence of schism and use turn-taking allocation rule and timing to extract the users who are involved in the schism. In the second pass, the algorithm separates the individual utterances to form different datasets using the waiting time (time gap) and turn-taking allocation rule. In the third pass, The algorithm recovers a complete distinct conversation thread from the utterances by looking at the participants-based features and the content similarity between the utterances.

4.1 Schism detection

There are two ways in which new conversations can start, one is through a schism and the other is through a conversation initiating statement. According to (Uthus and Aha, 2013) "Schism occurs when a conversation splits into two conversations the new conversation is formed due to certain participants branching off from a specific message and refocusing their attention upon each other". This implies that the users who are involved in schism were once an audience of the current speaker in the main conversation before the schism occurred and secondly, the two conversations seems to occur at the same time. With these features we can predict when and where schism starts.

4.2 Waiting time

We considered the waiting time or time gap in a chat room communication as the time difference between successive messages. The waiting time is calculated using approach in (Mihaljev et al., 2011) as

$$dt = t_{i+1} - t_i$$

,

where t_i is the time at i and t_{i+1} is the time at i+1.

For example in Fig 1, the waiting time or time gap between $A \rightarrow B$ and $B \rightarrow A$ in line 2 and 3 respectively is 29 seconds ($40:55 - 40:29$) and the waiting time or time gap between $B \rightarrow A$ and

$C \rightarrow (D, E, F, G, H)$ in line 3 and 4 respectively is 44 seconds $(40:29 - 41:13)$.

Since we know the temporal distribution of waiting time in a given conversation, we use this data to estimate the likelihood of a particular utterance belonging to a given conversation.

4.3 Content and participant based features

The content based features will involve comparing the number of word-similarity between two utterances. For example the number of words shared between utterance X and utterance Y suggests that the two utterances may belong to the same conversation (Joty et al., 2013). The participant-based features is described as follows:

- Pairs or group of utterances X and Y may be closely connected in the discourse and are likely to be directly related if those participating in utterance X are the same people participating in utterance Y and the time between them falls within the extracted waiting time distribution.
- Pairs or group of utterances X and Y may be widely separated in the discourse and are unlikely to be directly related if those participating in utterance X is totally different from those people participating in utterance Y.

5 Summary

We have proposed a simple and efficient approach for chat disentanglement. It will avoid using methods that are highly computational intensive, instead it uses simple data characteristics such as utterance similarities, response waiting time, turn-taking and the participant-based feature. With this approach, we hope to achieve results that will be nearer-human performance on an annotated corpus.

References

Seyit Ahmet Camtepe, Mark Goldberg, Mukkai Krishnamoorty, and Malik Magdon-ismail. 2005. Detecting conversing groups of chatters: a model, algorithms, and tests. In *In Proceedings of the IADIS International Conference on Applied Computing*, pages 89–96.

Faith Davies. 2010. The history of chat rooms.

Micha Elsner and Eugene Charniak. 2008. You talking to me? a corpus and algorithm for conversation disentanglement. In *Proceedings of ACL-08: HLT*, pages 834–842, Columbus, Ohio, June. Association for Computational Linguistics.

Micha Elsner and Eugene Charniak. 2010. Disentangling chat. *Comput. Linguist.*, 36(3):389–409.

Micha Elsner and Eugene Charniak. 2011. Disentangling chat with local coherence models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 1179–1189, Stroudsburg, PA, USA. Association for Computational Linguistics.

Micha Elsner and Warren Schudy. 2009. Bounding and comparing methods for correlation clustering beyond ilp. In *Proceedings of the Workshop on Integer Linear Programming for Natural Langauge Processing*, ILP '09, pages 19–27, Stroudsburg, PA, USA. Association for Computational Linguistics.

PatrickG.T. Healey, Graham White, Arash Eshghi, AhmadJ. Reeves, and Ann Light. 2008. Communication spaces. *Computer Supported Cooperative Work (CSCW)*, 17(2-3):169–193.

Shafiq Joty, Giuseppe Carenini, and Raymond T. Ng. 2013. Topic segmentation and labeling in asynchronous conversations. *J. Artif. Int. Res.*, 47(1):521–573, May.

Elijah Mayfield, David Adamson, and Carolyn Penstein Ros. 2012. Hierarchical conversation structure prediction in multi-party chat. In *In Proceedings of SIGDIAL Meeting on Discourse and Dialogue*.

T. Mihaljev, L. de Arcangelis, and H. J. Herrmann. 2011. Interarrival times of message propagation on directed networks. , 84(2):026112, August.

Matthew Purver. 2011. Topic segmentation. In G. Tur and R. de Mori, editors, *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*, pages 291–317. Wiley.

David C. Uthus and David W. Aha. 2013. Multiparticipant chat analysis: A survey. *Artificial Intelligence*, 199200(0):106 – 121.