| | |
|---|---|
| Title | Reconstruction of Threaded Conversations in Online Discussion Forums |
| Author(s) | Aumayr, Erik; Jeffrey, Chan; Hayes, Conor |
| Publication Date | 2011-07-18 |
| Publication Information | Reconstruction of Threaded Conversations in Online Discussion Forums, Erik Aumayr, Jeffrey Chan, Conor Hayes; paper delivered at the Fifth International AAAI Conference on Weblogs and Social Media (2011) |
| Publisher | Fifth International AAAI Conference on Weblogs and Social Media |
| Link to publisher's version | http://www.icwsm.org/2011/papers.php |
| Item record | http://hdl.handle.net/10379/4659 |

# Reconstruction of Threaded Conversations in Online Discussion Forums

**Erik Aumayr** and **Jeffrey Chan** and **Conor Hayes**
Digital Enterprise Research Institute
NUI Galway, Ireland
Email: {erik.aumayr, jkc.chan, conor.hayes}@deri.org

**Abstract**

Online discussion boards, or Internet forums, are a significant part of the Internet. People use Internet forums to post questions, provide advice and participate in discussions. These online conversations are represented as threads, and the conversation trees within these threads are important in understanding the behaviour of online users. Unfortunately, the reply structures of these threads are generally not publicly accessible or not maintained. Hence, in this paper, we introduce an efficient and simple approach to reconstruct the reply structure in threaded conversations. We contrast its accuracy against three baseline algorithms, and show that our algorithm can accurately recreate the in and out degree distributions of forum reply graphs built from the reconstructed reply structures.

## 1 Introduction

Internet forums are an important part of the Web. Along with Twitter, web logs and wikis, they provide a platform for questions to be asked and answered, information to be disseminated and public discussions on all types of topics. According to Internet Brands[1], 11% of Internet traffic in 2009 consists of visits to online forums, showing forums are still an integral part of the Web.

In forums, conversations are represented as sequences of posts, or threads, where the posts reply to one or more earlier posts. For example, Figure 1 shows a thread from the poker forum on www.boards.ie. It consists of a sequence of posts discussing how to become a Texas Hold'em poker dealer. Links exist between posts if one is the reply of another.

The threaded nature of forums allows us to follow the conversations, and thus study interesting problems. For example, users can be profiled and analysed based on their replying behaviour, which is extracted from the reply structure of forums. In (Chan, Daly, and Hayes 2010), users were profiled using this method, then grouped together into user roles of common behaviour. The roles were then used to decompose forums into percentage of users playing particular roles. Another sample application is in topic and trend tracking (Allan 2002). By recovering the reply structure, we can follow the actual conversation stream in threads, which might not be in the order the posts are posted. As it can be seen, the reply structure of threads have many applications.

There are many forums and many datasets of forums online. However, the reply structure of threads is not always available. This can be due to the failure of the board system to properly log them, it is not maintained by the providers, it is not publicly available or even lost. Therefore in this paper, we propose a new method to reconstruct the reply structure of posts in forums.

Prior work in reconstructing the thread structure is limited (Wang et al. 2008). They focus on either detecting question and answers in forums (Cong et al. 2008), which is only one type of thread in online forums, or only use content to reconstruct thread structure, which results in low accuracy (Wang et al. 2008). We propose a new approach to reconstructing thread structures. It uses a set of simple features and a classifier (a decision tree) to reconstruct the reply structure of threads. We evaluate the accuracy of the algorithm against the existing and a baseline algorithm using traditional notions of precision and recall and the ability of the algorithms to recreate the in and out degree distributions of reply graphs built from the reconstructed replying structure. We also analyse how well the algorithms perform in recreating the local in and out degrees and the clustering coefficient for individual vertices in the reply graphs.

In summary, the contributions of this work are:

- Proposal of a classification approach to reconstruct reply behaviours in threads, that uses content and non-content features.
- Show that the algorithm can accurately recreate the in and out degree distributions of the forum reply graphs that are created from the reconstructed reply structures.
- Show that the difference in accuracy of our algorithm and a baseline algorithm result in significant differences in the local degree and clustering coefficient values of the reply forum graphs.

The remainder of this paper is as follows. In Section 2 we describe related work, then we explain our approach to reconstructing threaded conversations in Section 3. In the next section, we present our evaluate and contrast the results of the different approaches. Finally, Section 5 concludes this paper and presents possible future work.



[1] http://www.internetbrands.com

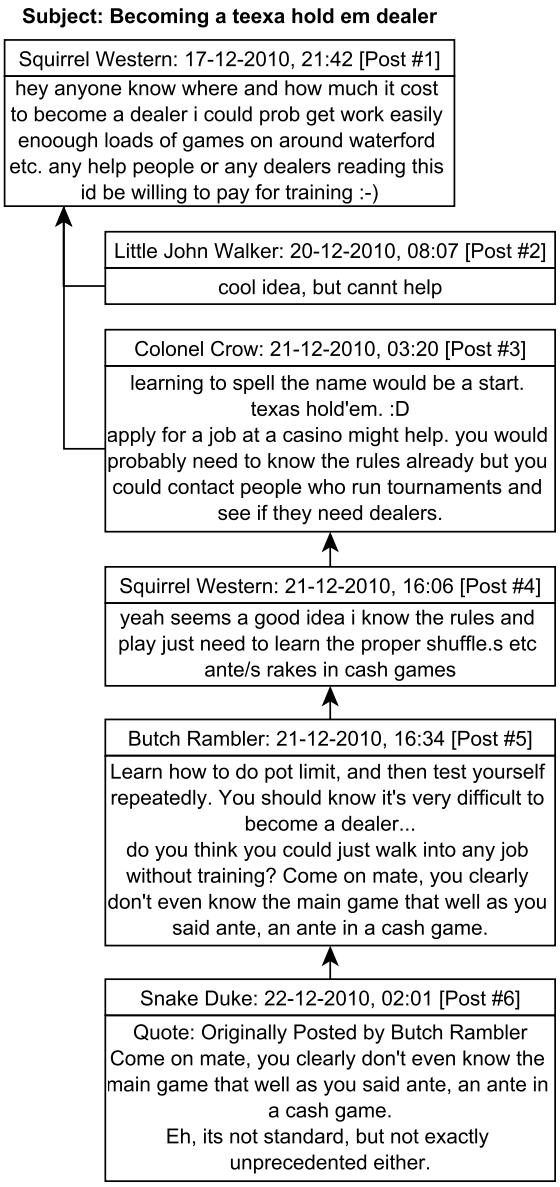



Figure 1: *Example thread in the poker forum of Boards.ie.*

## 2 Related Work

There is a number of related areas of work, including topic summarisation and discourse disentanglement, question and answer (Q & A) detection and ranking, and thread reconstruction. In this section, we will discuss each area in turn.

Much work has been done in summarising threaded conversations (Zhou, Hovy, and Rey 2005) (Shen et al. 2006) (Elsner and Charniak 2010), particularly from the perspective of topic detection and tracking (Allan 2002) and discourse disentanglement. All of these works summarize the conversations by defining a similarity on posts, typically content, and then clustering similar posts into segments of conversations. An exception is in discourse disentanglement (Elsner and Charniak 2010) (Wang and Oard 2009), that uses similar features to our work and can be used indirectly to build the reply structures. Elsner and Charniak used logistic regression to group utterances in a chat room conversation to separate topics, while Wang and Oard probabilistically expanded the messages by information about their context. However, their emphasis lies on grouping utterances, not finding the explicit thread structure, hence their features are not directly applicable to our problem. Nevertheless, we still think it is appropriate to compare Elsner and Charniak's method to ours in section 4. In summary, all these works in topic detection and tracking and discourse disentanglement are successful in summarizing and detecting topic segments, but this is a different task from reply structure reconstruction, as posts can be replies to each other but not related to each other.

In question and answering, there are two streams of similar work: one is to find the best set of answers for a query, and the other is to identify question and answer pairs to build a querying system knowledge base. Xi et al. (Xi, Lind, and Brill 2004) associated posts that are most likely to answer a user's query. They used content, thread based, and author based features to train a linear regression and SVM classifier. In (Feng et al. 2006), Feng et al. aimed to discover the most relevant answer in question threads. They combine lexical similarity, speech acts and reputation of the author of posts into a similarity measure, and use the HITS algorithm to find posts that are most likely to be answers to the initial question post. Similar to Feng et al., Hong et al. (Hong and Davison 2009) detected threads whose first posts are questions and its corresponding answers in the thread. To detect answers, they used features including the positions of the candidate answer posts, authorship (experts vs. newbies) and likelihood models based on content. Hong et al. tested different combinations of these features with an SVM classifier and found post position and authorship result in the highest accuracies. Cong et al. (Cong et al. 2008) took a more general approach, where they allow questions to occur anywhere in a thread. Questions were found using a rule classifier that discovered which frequent sequential Part-of-speech patterns in sentences lead to questions. For answers, Cong et al. used language models to construct weighted similarity graphs between each question and the set of candidate answers. For each detected question, a page-rank like propagation algorithm was utilised to determine and rank the set of candidate answers. In (Ding et al. 2008) and (Yang, Cao, and Lin 2009), the same set of authors extended their previous work (Cong et al. 2008) to incorporate the idea of context sentences and posts that provide indirect links between a question and answer pair. The authors used conditional random fields and structural support vectors to learn the contexts. In summary, much novel work has been conducted in question and answer detection, but the focus of this type of work is to detect a question and rank the candidate answers, while in the reply reconstruction task, we are trying to link posts together, regardless whether they are question-answer pairs or not.

There are few works in the literature that directly address

the problem of thread reply reconstruction. In (Yeh and Harnly 2006), Yeh and Harnly reconstructed reply chains of emails, or email threads. They proposed two approaches, one based on using header meta-information, the other based on timing, subject and content of the emails themselves. The reconstruction heuristic consists of a number of threshold tests on the features. The main difference is that this approach is specific for emails, and the features and heuristic cannot be easily extended for forum thread reconstruction. The work by Wang et al. (Wang et al. 2008) is most similar to ours. In that work, the authors reconstructed the reply structure of threads using content similarity and three different windowing schemes. The windowing schemes define different temporal constraints placed on possible parent post candidates. In their testing, very low precision and recall were obtained for a small forum dataset. In our experiments, we will show that this approach also has low precision and recall on our testing datasets.

# 3 Methodology

In this section we describe our approaches to reconstructing the threaded conversations. We first define the information a thread provides and the nature of replies in it, then introduce the classification features we extract from this information and finally the classification methods.

## 3.1 Definitions

In order to recover the reply structure in a thread, we need to specify what information we expect from it. Our goal is to apply our approach to as many forums as possible. Therefore we want to only rely on a few basic criteria. Most forums provide the following information about a post:

- Creation date
- Name of author
- Quoted section of another post
- Content
- Thread length

Although some forums may save the id of a quoted post, most do not, instead they save the name of the quoted user. Therefore in our work, we only assume we know the name of the quoted user and of course the quoted text section. The length of a thread simply states how many posts it consists of. Note that the creation dates of posts establish a chronological ordering. From that ordering we can compute the distance of one post to another. For instance, in figure 1 post #3 has a distance of 2 to post #1 because there is one other post in between.

Next, we describe the nature of replies. In the current forum user interfaces, a reply is initiated when a user hits the Reply-button of the post she wants to answer. The data we used stores this type of reply interaction, i.e., each post can only reply to one other posts. Although it is possible for people to reply to several posts at once, and it is possible for our approaches to return more than one reply candidate, in our evaluation we limit each post to one reply as this was the replying structure of the training and testing dataset.

## 3.2 Baseline Approaches

In the dataset we used, we found most replies are located in close reply distance to the post they are replying to. Most of them (79.7%) follow immediately after the post they are replying to. This implies that the post reply distance is likely to be a strong feature and that we can construct a simple reconstruct method based on this fact.

Therefore, as a baseline approach, we link each post to its immediate predecessor. Any proposed algorithm should perform better than this baseline. We will refer to this baseline approach as the "1-Distance" linking algorithm.

Further, we implemented the approach from (Wang et al. 2008) and investigated the logistic regression classifier from (Elsner and Charniak 2010). The authors of the latter work trained the classifier with features similar to ours, like time difference, content similarity and distance. However, they used discourse based features like the likelihood of a post being an answer to a question post. When evaluating the method of (Elsner and Charniak 2010), we did not apply these features as they are not relevant to the reply reconstruction problem.

## 3.3 Reply Reconstruction Features

When choosing the features for our classification approach we carefully focus on features that are simple, easy to obtain and fast to compute. This enables a real-time reconstruction of the thread reply structures. The features we use are:

- In section 3.2 we found that the **reply distance** might be a strong feature. Because of the chronological order the distance between two posts expresses how many other messages had been posted in the time between.
- Similar to the distance the **time difference** between posts can express the message behaviour of users and therefore is considered a useful feature. In figure 1 the time difference between post #1 and post #2 is two days and 13.5 hours whereas the time difference between post #4 and post #5 is only 28 minutes.
- Only 20% of replies have **quotes**. But whenever a quote occurs in the content of a message, the user name and a section of the quoted message is given, see figure 1 in post #6. This provides a very accurate reply indication.
- The most complex feature we investigate is **cosine similarity**. Cosim calculates the angle between two vectors. In this case these two vectors are the contents of two messages. Because a message content is typically a plain text tf-idf (term frequency-inverse document frequency) is used to convert text into vectors with numerical elements which then can be computed by cosim. Before applying tf-idf we pre-process the contents to get texts that are as general as possible. To reduce the dimensions of the vectors we filter out stop words (inclusive the word "quote" since it marks a quotation) and words that appear only once. In addition we stem the words by using the Porter Stemmer (Porter 1980).
- When having the distance the **thread length** is an additional feature that, combined with the distance, could work as an indicator which limits the reply distance. The

length of a thread is the number of posts it contains. Since a classifier will not learn when a thread begins or ends, the length categorizes threads which can help to improve the reply distance feature. Thread length will not work as a feature on its own.

The computation of tf-idf and cosine similarity between two messages is clearly the most costly operation we choose to perform. We get the information about post distance and time difference from the time stamp of posts and the length of a thread can easily be summed up.

We also considered author activity as a possible feature, and investigated whether there is viable information in the difference between the amount of messages an author posts and receives. We found that users who post a lot of messages also receive a lot of messages simply by providing many posts other users can reply to. Rather than finding popularity among users and thus a higher probability to receive replies, the sending and receiving is linearly dependent and does not hold any information a classifier can make use of.

### 3.4 Classifiers

In this subsection, we describe the classifiers we use. There is a huge diversity of classifiers that can be used to predict missing data. We focus on two widely used classifiers, namely Support Vector Machines (SVM) and the C4.5 Decision Tree.

**Support Vector Machines** are commonly used for classification and regression analysis. For example, recall Hong et al. (Hong and Davison 2009) used them for detecting questions and corresponding answers in Internet forums. An SVM partitions the training data into two classes using a set of multi-dimensional hyperplanes. These hyperplanes separate the data into the two classes. The SVM then predicts the class of new data points by deciding which side of the hyperplane(s) the data points lie. We use the SVN implementation for Weka (EL-Manzalawy and Honavar 2005).

Although Support Vector Machines can be very powerful in terms of accuracy they are known to be very slow. Therefore we use **C4.5 decision tree algorithm** (Salzberg 1994) as a comparison. The decision trees' big advantage is its ability to handle huge amounts of data very efficiently, due to the relative simplicity of its model and training approach. It consists of a tree of branching decisions like "is the reply distance $\leq 5$?". If so, the tree evaluates further features and thresholds until it can decide which class a data point belongs to. An additional benefit is that the resulting tree is easy to interpret by humans.

However, decision trees can overfit the training data by growing too large and hence, at times, not generalize well. To overcome this the decision tree must be restricted in size and number of resulting leaves. C4.5 provides backward pruning to do so. We use C4.5's Java implementation in Weka, called J48. In the next section, we will these two classifiers, with different combinations of the features.

## 4 Evaluation

In the following we present the results of our thread reconstruction algorithm and compare them to the results of the baseline approaches. First, we provide details about the dataset we use for evaluation. We then describe the accuracy measurements and compare the accuracy of using different combinations of the features and classifiers. Finally, we evaluate the 1-Distance approach against the decision tree approach in recreating the in and out degree and clustering coefficient distributions.

### 4.1 Dataset

The dataset we use for training and evaluation is from the forum Boards.ie. It is the largest Irish online discussion board and the dataset covers ten years. The dataset has the replying structure of posts, which allows us to train and test our supervised approaches. We limit the data for our experiments to threads around the years 2006 and 2007 and we further introduce a lower and upper boundary regarding the number of posts a thread contains. The lower boundary of 3 posts filters out threads with only one or two posts, which are trivial to classify and cause a lot of bias regarding the post distance. 27.7% of the original threads have one or two posts and are subsequently removed from the dataset. The restriction of 40 posts filters out very long, uncommon threads which, if left in the dataset, would cause much bias. Some of these threads have several thousands of posts. An example is a sticky thread in which moderators welcome new members and explain forum rules. This action reduces the dataset by another 5.1% to 120,100 threads, with $\approx$ 827,900 posts. In our experiments we have noticed that the SVM takes a very long time to train on this amount of data. Therefore we further reduce the size of the considered data by random sampling over the same period of time and finally end up with 13,100 threads containing 133,200 messages. The smaller dataset shows how well the classifiers work with limited data.

The next step towards classification is to extract both positive and negative samples to train the classifiers. Our positive instances are pairs of posts which are actually replying, i.e., the second post replies to the first. Negative samples are created in the following way: in each thread, we link each post to a random set of earlier posts that it is not actually replying to. Using this method we generated an equally distributed set of positive and negative samples for training and testing, with 213,800 pairs of posts.

### 4.2 Evaluation Approach

In order to compare the different approaches we use the well known measurements precision, recall and F-score.

$$Precision = \frac{TP}{TP + FP},\ Recall = \frac{TP}{TP + FN}$$

TP, FP and FN stand for true positive, false positive and false negative respectively and express correct or incorrect classified instances. The F-score is the harmonic mean between Precision and Recall:

$$F\text{-}Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

For training and testing of the classifiers and features, we use the Weka toolkit (Waikato Environment for Knowledge

Analysis, (Hall et al. 2009)) and applied 5 fold cross validation to minimise bias.

### 4.3 Evaluation of the Existing and Baseline Approaches

In this subsection, we first present the results for the existing and baseline algorithms.

- (Wang et al. 2008): Evaluating this thread structure recovering approach, we obtained a precision, recall and F-score of 0.444 on our dataset, which is slightly higher than on their dataset.
- 1-Distance linking: We found in section 3.2 that 79.7% of the replies have a post distance of 1. Assuming a reply distance of 1 over all messages results in both a precision and recall as well as F-score of 0.797.
- (Elsner and Charniak 2010): With the logistic regression classifier, we achieved an F-score of 0.875 (0.824 and 0.932, precision and recall respectively).

We will contrast the C4.5 classifier with the best working set of features against these three approaches at the end of section 4.5.

### 4.4 Evaluation of the Features

In this subsection we evaluate the usefulness of single features and combinations of features for reply structure reconstruction. The results for SVM (Table 2) and decision tree (Table 1) have similar trends for the attributes, hence we limit our discussion to the results of C4.5 here.

As Table 1 shows, the post distance is the best single feature. It achieves the highest recall (0.773) and F-score (0.848), whereas the quotes feature has the highest precision of 0.981. However, the quoting feature has very low recall (0.235), due to the fact that only about 20% of posts in the training and testing data have quotes. Using the most promising single features, we constructed different combinations of two or more features. For two features, the best performing combination is distance + quotes with the highest precision (0.943), recall (0.898) and accordingly F-score (0.92). This pair of features works best because it combines the high precision of the quotes feature with the high recall of the post distance. We also added the thread length to the distance to see whether the thread length categorization helps. The distance + length combination has increased recall but decreased precision, however we will show that the thread length does not perform well in larger feature combinations. Again taking the best numbers together and forming triples of features we get the highest recall with distance + time difference + quotes (0.942) and the highest precision and F-score with distance + quotes + cosim (0.91 and 0.925). The features post distance and quotes reappear in these two triples, augmented by time difference (precision) and cosim (recall). Now combining all single features distance + time difference + quotes + cosim the precision drops a little from 0.942 to 0.939 but recall and F-score rises to 0.915 and 0.927. As said earlier the thread length, which is not a single feature, does not help the classifier much. Table 1 shows that thread length has very little impact on the figures when using C4.5, i.e., there is almost no difference between the combination of the four single features and the same combination with the thread length. Feeding the thread length into the SVM classifier makes it even worse as shown in Table 2. So the best combination for C4.5 is all features combined, for the SVM it is distance + quotes + cosim.

| Features | Prec. | Recall | F-score |
|---|---|---|---|
| Distance | 0.938 | **0.773** | **0.848** |
| Time Difference | 0.671 | 0.574 | 0.619 |
| Quotes | **0.981** | 0.235 | 0.379 |
| Cosine Similarity | 0.763 | 0.377 | 0.505 |
| Distance + Length | 0.923 | 0.79 | 0.851 |
| Distance + TimeDiff | 0.928 | 0.793 | 0.855 |
| Distance + Quotes | **0.943** | **0.898** | **0.92** |
| Distance + Cosim | 0.926 | 0.891 | 0.908 |
| Dist + TimeDiff + Quotes | **0.942** | 0.904 | 0.922 |
| Dist + TimeDiff + Cosim | 0.928 | 0.895 | 0.911 |
| Dist + Quotes + Cosim | 0.94 | **0.91** | **0.925** |
| All Single Features | 0.939 | 0.915 | 0.927 |
| All Singles + Length | **0.939** | **0.918** | **0.928** |

Table 1: *Impact of Features with Decision Tree – the best results for each measure and feature category is highlighted in bold.*

### 4.5 Classifiers

We compare the results of the classifiers in the following. As mentioned in section 4.1 we had to cut down the dataset drastically to get any results back from the SVM in under a week of runtime. It can not handle big amounts of data in an acceptable period of time, making a real-time thread reconstruction by the SVM infeasible. Training it on one single feature took the SVM 3-5 hours, 5-20 hours when processing two to four features, and almost 30 hours to complete the 5 fold cross validation of five features (including thread length). In terms of precision and recall the SVM can perform well, although it has a long runtime. Table 2 presents the results for the SVM.

Comparing the results, it can be seen that that the difference between the two is small. In a some cases the SVM is more accurate and in other cases the C4.5 decision tree algorithm is more accurate. In fact. with all single features combined (with or without thread length) the decision tree has superior accuracy. In the remainder of the paper, we will refer to the decision tree combined with the best working set of features as "ThreadRecon".

The main advantage of the decision tree is the time it needs to train and classify. It took less than two minutes to train and be evaluated on. We ran the decision tree on a much larger dataset of 3.9 million samples and it still finishes the classification task within 15 minutes.

We conclude this subsection by contrasting the best results of the three baseline approaches and ThreadRecon, shown in Table 3. ThreadRecon is more accurate then the other approaches, even the state of the art approach proposed in (Elsner and Charniak 2010).

| Features | Prec. | Recall | F-score |
|---|---|---|---|
| Distance | 0.938 | **0.773** | **0.848** |
| Time Difference | 0.656 | 0.609 | 0.632 |
| Quotes | **0.981** | 0.235 | 0.379 |
| Cosine Similarity | 0.828 | 0.32 | 0.462 |
| Distance + Length | 0.925 | 0.789 | 0.851 |
| Distance + TimeDiff | 0.919 | 0.792 | 0.851 |
| Distance + Quotes | **0.943** | **0.898** | **0.92** |
| Distance + Cosim | 0.935 | 0.876 | 0.905 |
| Dist + TimeDiff + Quotes | **0.942** | 0.885 | 0.912 |
| Dist + TimeDiff + Cosim | 0.934 | 0.858 | 0.895 |
| Dist + Quotes + Cosim | **0.942** | **0.903** | **0.922** |
| All Single Features | 0.941 | 0.888 | 0.914 |
| All Singles + Length | 0.936 | 0.872 | 0.903 |

Table 2: *Impact of Features with SVM – the best results for each measure and feature category is highlighted in bold.*

| Wang 2008 | 1-Distance | Elsner 2010 | ThreadRecon |
|---|---|---|---|
| 0.444 | 0.797 | 0.875 | 0.926 |

Table 3: *F-score comparison between ThreadRecon and baseline approaches.*

### 4.6 Reply Graph Reconstruction

To qualify the accuracy difference between 1-Distance Linking and ThreadRecon and demonstrate the importance of increasing the accuracy, we computed two important statistics for social network analysis – analysing the in degree distribution and the clustering coefficient of forum reply graphs. We constructed the forum reply graphs from the actual and predicted reply structures of the 1-Distance and our thread reconstruction algorithms, and compared the in and out degree and clustering coefficient statistics of the actual reply graph versus the predicted reply graphs. The forum reply graph represents the replying structure of the users in a forum, and is used in applications like role analysis (Chan, Daly, and Hayes 2010). The vertices in the graph represent the users, and a directed edge represents one user replying to another user. Edge weights can represent the amount of replies from one user to another, but in the following analysis we just concentrate on the existence of edges and not consider the edge weights.

**In/Out Degree** To evaluate the in and out degree distributions, we initially plotted the histogram of the in and out degree distributions of the three reconstructed reply graphs, actualGraph, oneDistGraph and threadReconGraph for the graphs constructed from the actual, the 1-Distance algorithm and our thread-reconstruction algorithm respectively. Figure 2a shows the in-degree histograms for the actualGraph. We do not plot the other two histograms because they are very similar to Figure 2a. To quantify their similarity, we use the KL-divergence measure (Cover and Thomas 2006), a well known measure for comparing histograms. It is an information theoretic measure which measures the uncertainty remaining about a distribution (or histogram) given knowledge of the other. The lower the measure, the closer the two distributions are. To use the KL-divergence measure, we normalise the frequencies of each bin, such that the normalised frequencies sum to 1. The KL-divergence of the in-degree distributions of the actualGraph and the oneDistGraph is 0.0139 bits, and for the in-degree distribution of the actualGraph and the threadReconGraph, it is 0.0036 bits. Both these values are very low. We also analysed the number of elements that were placed in the incorrect bin, and found only 487 and 227, out of 9877 elements, that were misplaced for oneDistGraph and threadReconGraph respectively. We got similar results for the out-degree distributions. These results suggest both approaches can reproduce reply graphs that have highly accurate in and out degree distributions.

To investigate if the in and out degree of the reconstructed graphs were similar on a vertex level, we computed the difference or error between the in and out degrees of each vertex. As most vertices have low in and out degrees (see histograms of Figures 2a and 3a respectively), indicating that most vertices would have low absolute error by default, we computed the relative or percentage error instead. It is defined as:

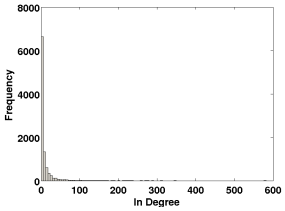
$$PE(v_{ka}, v_{kp}) = \frac{|d_{v_{ka}} - d_{v_{kp}}|}{max(d_{v_{ka}}, d_{v_{kp}})}$$

where $d_{v_{ka}}$ is the degree (in or out) of vertex $v_{ka}$. The percentage error measures the relative difference in the degrees of the vertices of the actualGraph against oneDistGraph and threadReconGraph and is more agnostic to the skewed degree distributions. However, low degree vertices can have a very large percentage error, and to avoid them unnecessarily distorting the results, we focus on vertices with a total degree of 10 or more in the actualGraph.

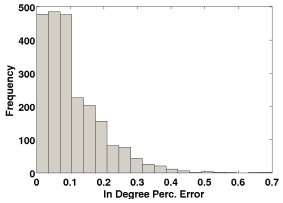
We plotted the in-degree histograms of the percentage error in Figures 2b and 2c and the out-degree histograms in Figures 3b and 3c. The figures confirm that on a vertex basis, there is some difference between the graphs. In addition, the figures indicate that the threadReconGraph has more vertices with a small percentage error, compared to the results of the oneDistGraph. This is shown more clearly in the cumulative distribution function (CDF) plot of the the two results, in Figures 2d and 3d. For example, 80% of vertices in the threadReconGraph have a percentage error 0.05 or less, compared with only 30% for the oneDistGraph. This indicates the difference in the precision, recall and F-score values do result in differences in the individual in and out degrees of the reconstructed reply graphs, and has an effect on subsequent analysis.

This analysis shows that the reply reconstruction algorithms can accurately reproduce the in and out degree distributions. However, when the degrees are analysed vertex by vertex basis, we found local differences, with the threadReconGraph doing better than the oneDistGraph.

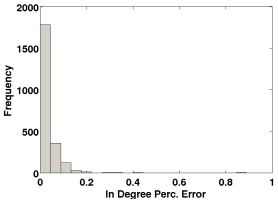
**Clustering Coefficient** The clustering coefficient measures the amount connectivity around the neighbourhood of specific vertices. In this subsection, we investigate the difference between the clustering coefficient of each vertex in the actual reply graph and the reconstructed graphs from



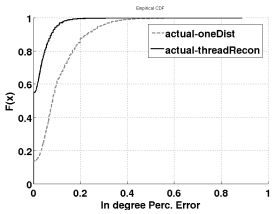

(a) *Histogram of the in-degree of all vertices of the actualGraph reply graph.*




(b) *In-degree percentage error between the vertices of actualGraph and oneDistGraph, for all vertices with degree* $\geq 10$.

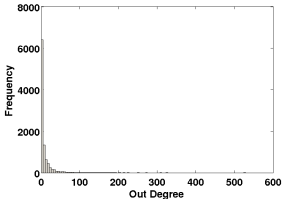



(c) *In-degree percentage error between the vertices of actualGraph and threadReconGraph, for all vertices with degree* $\geq 10$.

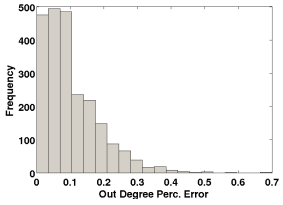



(d) *CDF plot of the in-degree percentage error, comparing the results for oneDistGraph and threadReconGraph.*

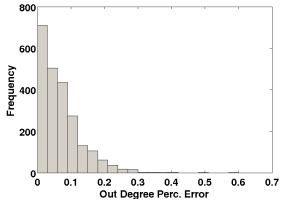
Figure 2: *Histogram and CDF plots for the in-degree evaluation.*



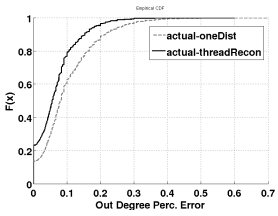

(a) *Histogram of the out-degree of all vertices of the actualGraph reply graph.*




(b) *Out-degree percentage error between the vertices of actualGraph and oneDistGraph, for all vertices with degree* $\geq 10$.




(c) *Out-degree percentage error between the vertices of actualGraph and threadReconGraph, for all vertices with degree* $\geq 10$.




(d) *CDF plot of the out-degree percentage error, comparing the results for oneDistGraph and threadReconGraph.*

Figure 3: *Histogram and CDF plots for the out-degree evaluation.*

the two algorithms. The clustering coefficient for directed graphs $G(V, E)$ is defined as:

$$C(v_k) = \frac{|\{e_{i,j}\}|}{d_k(d_k - 1)}, v_i, v_j \in N_k, e_{i,j} \in E$$

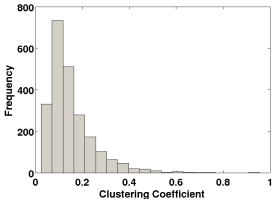
Figure 4a shows the clustering coefficient histogram of actualGraph, for all vertices that have degree 10 or more. Similarly to the in and out degree analysis, we limit the analysis to vertices of degree of 10 or more to reduce the distortion of low clustering coefficient vertices. To measure the difference between the clustering coefficients, we again used the percentage error of the coefficient for each 10 degree plus vertex in the actualGraph against the corresponding coefficients in the reconstructed graphs.

Figure 4b and 4c shows the histogram of the percentage difference of the coefficients between the 1-Distance reconstructed graph and the actualGraph, and between the threadReconGraph and the actualGraph respectively. As the figures illustrate, the threadReconGraph has a larger percentage of low difference vertices than the 1-Distance reconstructed graph. This is made more clear in Figure 4d, which is the CDF of the histogram figures. For example, 80% of vertices of the threadReconGraph has percentage difference less than 17% while it is only 22% for the oneDistGraph.

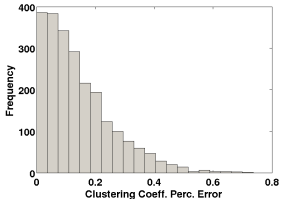
In this section, we have evaluated different combinations of the features and two reconstruction algorithms on their precision, recall and F-score. We found using all the features and the C4.5 decision tree resulted in the best balance of accuracy and efficiency. We have also shown the difference in the in and out degrees and clustering coefficient of individual vertices between the 1-Distance algorithm and our thread-reconstruction algorithm ThreadRecon (C4.5 decision tree with all features). This quantifies the effect that the differences in precision, recall and F-score has on two important real graph analysis measures.

## 5 Conclusion and Future Work

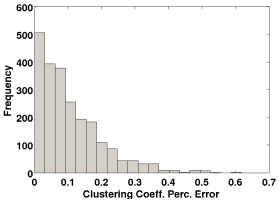
The goal of this work is to investigate methods to recover reply structures in forum threads in an accurate and efficient way. We have extracted a number of features from the basic information a thread provides, i.e., post distance, time difference, quoting and cosine similarity. We found that the features thread length and author activity do not augment the accuracy. For the classification task we examined support vector machines and the C4.5 decision tree algorithm. The decision tree outperforms the SVM in learning and classification speed as well as in the size of data it can process in a feasible time. In terms of precision, recall and F-score the decision tree achieves the best results (0.939, 0.918 and 0.926) with the combination of all features. Compared to the three baseline approaches (Wang et al. 2008), "1-Distance" Linking and (Elsner and Charniak 2010), this is an F-score



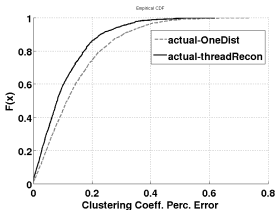

(a) *Histogram of the clustering coefficient for the actualGraph reply graph.*




(b) *Clustering coefficient percentage error between the vertices of actualGraph and oneDistGraph.*




(c) *Clustering coefficient percentage error between the vertices of actualGraph and threadReconGraph.*




(d) *CDF plot of the clustering coefficient percentage error, comparing the results for oneDistGraph and threadReconGraph.*

Figure 4: *Histogram and CDF plots for the clustering coefficient evaluation, based on vertices with degree* $\geq 10$.

improvement of 52.1%, 13.9% and 5.5% respectively. We have also investigated the in and out degree and the clustering coefficient between the vertices of the reconstructed reply graph and showed the reconstruction approaches can accurately reproduce in and out degree distributions. Furthermore, on the vertex level, we have shown how the decision tree algorithm produces more accurate reconstructions than the 1-Distance baseline algorithm.

For future work, we look at improving our set of features for the thread reconstruction. One additional characteristic we can consider is the history of interaction between users. When users reply to each other in a thread, there might be an increased probability that they reply to each other again later on; for example, this can occur in a dialogue between two people. We are also interested in investigating contextual features, like the topic and trends of a thread. If we know the sub-topics within a thread, we can limit our search to posts related to a sub-topic, as the likelihood of replies across topics is lower.

## Acknowledgements

This work was carried out in the CLIQUE Strategic Research Cluster which is funded by Science Foundation Ireland (SFI) under grant number 08/SRC/I1407. We would like to thank John Breslin for providing the Boards.ie data.

## References

Allan, J. 2002. *Topic detection and tracking*. Norwell, MA, USA: Kluwer Academic Publishers.

Chan, J.; Daly, E.; and Hayes, C. 2010. Decomposing discussion forums and boards using user roles. In *Proceedings of the AAAI ICWSM conference*.

Cong, G.; Wang, L.; Lin, C.; Song, Y.; and Y. 2008. Finding question-answer pairs from online forums. *Proceedings of the ACM SIGIR conference*.

Cover, T. M., and Thomas, J. A. 2006. *Elements of information theory*. New York, NY, USA: Wiley-Interscience, 2 edition.

Ding, S.; Cong, G.; Lin, C.-Y.; and Zhu, X. 2008. Using conditional random fields to extract contexts and answers of questions from online forums. *Proceedings of the ACL-HLT conference* 710–718.

EL-Manzalawy, Y., and Honavar, V. 2005. *WLSVM: Integrating LibSVM into Weka Environment*. Software available at `http://www.cs.iastate.edu/~yasser/wlsvm`.

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

Feng, D.; Shaw, E.; Kim, J.; and Hovy, E. 2006. Learning to detect conversation focus of threaded discussions. In *Proceedings of the NAACL-HLT conference*, 208–215.

Hall, M.; Frank, E.; Holmes, G.; Pfahringer, B.; Reutemann, P.; and Witten, I. H. 2009. The WEKA Data Mining Software: An update. 11.

Hong, L., and Davison, B. D. 2009. A Classification-based Approach to Question Answering in Discussion Boards. In *Proceedings of the ACM SIGIR conference*, 171–178.

Porter, M. F. 1980. An algorithm for suffix stripping. *Program* 14(3):130–137.

Salzberg, S. L. 1994. C4.5: Programs for Machine Learning by J. Ross Quinlan. Morgan Kaufmann Publishers, Inc., 1993. *Machine Learning* 16:235–240.

Shen, D.; Yang, Q.; Sun, J.-t.; and Chen, Z. 2006. Thread Detection in Dynamic Text Message Streams. In *Proceedings of the ACM SIGIR conference*, 35–42.

Wang, L., and Oard, D. W. 2009. Context-based message expansion for disentanglement of interleaved text conversations. *Proceedings of the NAACL-HLT conference* 200–208.

Wang, Y.; Joshi, M.; Cohen, W.; and Rosé, C. 2008. Recovering implicit thread structure in newsgroup style conversations. In *Proceedings of the AAAI ICWSM conference*, 152–160.

Xi, W.; Lind, J.; and Brill, E. 2004. Learning effective ranking functions for newsgroup search. In *Proceedings of the ACM SIGIR conference*, 394–401.

Yang, W.-Y.; Cao, Y.; and Lin, C.-Y. 2009. A Structural Support Vector Method for Extracting Contexts and Answers of Questions from Online Forums. *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing* 514–523.

Yeh, J.-y., and Harnly, A. 2006. Email Thread Reassembly Using Similarity Matching. In *3rd Conference on Email and Anti-Spam*.

Zhou, L.; Hovy, E.; and Rey, M. 2005. Digesting Virtual " Geek " Culture : The Summarization of Technical Internet Relay Chats. In *Proceedings of the ACL conference*, 298–305.