The Utility of Discourse Structure in Identifying Resolved Threads in Technical User Forums

Li Wang[1,2] *Su Nam Kim*[3] *Timothy Baldwin*[1,2]
(1) Dept. of Computing and Information Systems, The University of Melbourne
(2) NICTA Victoria Research Laboratory
(3) Faculty of Information Technology, Monash University
li@liwang.info, sunamkim@gmail.com, tb@ldwin.net

Abstract

Online discussion forums are a valuable means for users to resolve specific information needs, both interactively for the participants and statically for users who search/browse over historical thread data. However, the complex structure of forum threads can make it difficult for users to extract relevant information. Automatically identifying whether the problem in a thread has been solved or not can help direct users to threads where the original problem has been solved, hence enhancing their prospects of solving their particular problem. In this paper, we investigate the task of Solvedness classification by exploiting the discourse structure of forum threads. Experimental results show that simple features derived from thread discourse structure can greatly boost the accuracy of Solvedness classification, which has been shown to be very difficult in previous research.

Keywords: Discourse Structure, Web User Forum, Social Media, Dialogue Act.

Wang, Li, Su Nam Kim and Timothy Baldwin (2012) The Utility of Discourse Structure in Identifying Resolved Threads in Technical User Forums, In Proceedings of the 24th International Conference on Computational Linguistics (COLING 2012), Mumbai, India, pp. 2739–2756.

1 Introduction

Web user forums (or simply "forums") are online platforms for people to discuss information and obtain information via a text-based threaded discourse, generally in a pre-determined domain (e.g. IT support or DSLR cameras). With the advent of Web 2.0, there has been an explosion of web authorship in this area, and forums are now widely used in various areas such as customer support, community development, interactive reporting and online eduction. In addition to providing the means to interactively participate in discussions or obtain/provide answers to questions, the vast volumes of data contained in forums make them a valuable resource for "support sharing", i.e. looking over records of past user interactions to potentially find an immediately applicable solution to a current problem. On the one hand, more and more answers to questions over a wide range of domains are becoming available on forums; on the other hand, it is becoming harder and harder to extract and access relevant information due to the sheer scale and diversity of the data.

In the domain of troubleshooting-oriented forums, one potential way to enhance information access and support sharing is to automatically identify threads where the original information need has been resolved. By filtering out threads which do not contain a valid answer, we can focus the attention of users on threads which have a greater chance of containing the required solution. Baldwin et al. (2007) explore this task of Solvedness classification, and find that it is an extremely difficult problem. Figure 1 shows an example thread, made up of 5 posts from 3 distinct participants, from the ILIAD (Improved Linux Information Access by Data Mining) data set of Baldwin et al. (2007). In this thread, Post1 and Post3 are both from the thread's initiator UserA. Post1 asks a question, and Post3 asks for more information about an answer provided by UserB in Post2. In response to Post3, UserB adds more information to his/her original answer, and Post5 provides another independent answer. In threads like this, it is important to identify whether the problem is solved or not, and also where solution(s) are likely to be found.

This research proposes to use information derived from thread discourse structure (Kim et al., 2010b; Wang et al., 2011) to help predict Solvedness of threads, without validating the answers provided in the threads. The discourse structure of the thread is modelled as a rooted Directed Acyclic Graph (DAG), and each post in the thread is represented as a node in this DAG. The reply-to relations between posts are then denoted as direct edges (Links) between nodes in the DAG, and the type of a reply-to relation is defined as Dialogue Act (DA). The Link between two connected posts (i.e. having a reply-to relation) is represented as the distance between the two posts in their chronological ordering. In the annotated version of the example ILIAD thread, as is shown in Figure 1, UserA initiates the thread with a question (Dialogue Act = Question-question: (Kim et al., 2010b)) in the first post, by asking a question. In response, UserB provides an answer (Dialogue Act = Answer-answer). Then, UserA confirms more details about the answer provided (Dialogue Act = Answer-confirmation). UserB responds to UserA to add more information about his/her previous answer (Dialogue Act = Answer-add). Finally, UserC proposes an independent answer again to the original question (Dialogue Act = Answer-answer).

Specifically, we explore features extracted from the thread discourse structure which can be used to help classify the Solvedness of threads. We experiment with both gold-standard and automatically predicted discourse structure, and find that thread discourse structure (which in no way evaluates the correctness of each post) can, indeed, boost thread classification accuracy, achieving state-of-the-art results over the task. We also investigate the correlation between

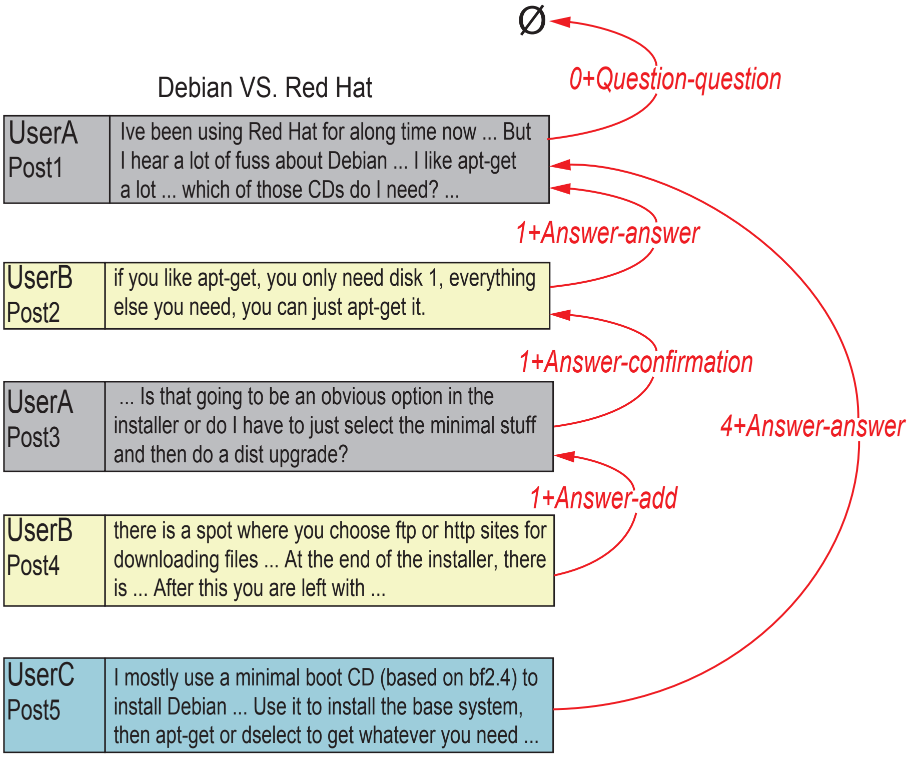

Ø
0+Question-question
Debian VS. Red Hat
UserA
Post1
Ive been using Red Hat for along time now ... But I hear a lot of fuss about Debian ... I like apt-get a lot ... which of those CDs do I need? ...
1+Answer-answer
UserB
Post2
if you like apt-get, you only need disk 1, everything else you need, you can just apt-get it.
1+Answer-confirmation
UserA
Post3
... Is that going to be an obvious option in the installer or do I have to just select the minimal stuff and then do a dist upgrade?
4+Answer-answer
1+Answer-add
UserB
Post4
there is a spot where you choose ftp or http sites for downloading files ... At the end of the installer, there is ... After this you are left with ...
UserC
Post5
I mostly use a minimal boot CD (based on bf2.4) to install Debian ... Use it to install the base system, then apt-get or dselect to get whatever you need ...

Figure 1: A snippeted ILIAD thread with annotated discourse structure.

thread discourse structure prediction F-score and thread Solvedness classification accuracy, and demonstrate a positive correlation. Finally, we show that focusing on improving the F-score over certain dialogue acts is able to boost Solvedness classification.

2 Related Work

As far as we are aware, there is very little NLP work that is specifically targeted at the thread-level analysis of web user forum data. The most closely-related work is that performed by Baldwin et al. (2007), on which this work is directly based. Baldwin et al. (2007) focused on three specific characteristics detected from forum threads, namely *Task orientation* (i.e. whether a thread focuses on solving a problem), *Completeness* (i.e. whether the initial post provides enough information of the problem) and *Solvedness* (i.e. whether the problem is solved). Three classification tasks were identified base on these three characteristics, and experiments were carried out using a range of classification and regression methods. Baldwin et al. (2007) explored not only bag-of-words features, but also another 18 lexical and contextual features from distinct partitions of the thread, namely *initial posts*, *first response post*, *all responses posts* and *final post from the initiator*. 250 threads with various topics from a Linux-related forum and mailing list were annotated for use in the experiments in the paper. While the experimental results illustrated the difficulties in performing the three tasks automatically, the experiments also implied that their approach could be employed to rank threads based on their characteristics.

Research on thread discourse structure analysis and classification over user forums has gained in momentum in recent years. Fortuna et al. (2007) defined 5 post-level dialogue acts to describe the levels of agreement (i.e. agreement, disagreement, insult) and identify questions and answers (i.e. question and answer) in forum posts. Xi et al. (2004) defined 5 prevalent types of post-level dialogue acts in forum threads. This set of dialogue acts was then adapted and extended by Kim et al. (2010b) to describe possible types of posts in troubleshooting-oriented online forums. Specifically, Kim et al. (2010b) devised a post-level dialogue act set and annotated a set of threads taken from CNET.[1] In this work, they proposed a set of novel features, which they applied to the separate tasks of post link classification and dialogue act classification. They later applied the same basic methodology to dialogue act classification over one-on-one live chat data with provided message dependencies (Kim et al., 2010a), demonstrating the generalisability of the original method. In both cases, however, they tackled only a single task, either link classification (optionally given dialogue act tags) or dialogue act classification, but never the two together.

Wang et al. (2011) delved into the task of thread discourse structure parsing further. They used the same features as Kim et al. (2010b), but different parsing approaches. Specifically, Wang et al. (2011) approached thread discourse structure parsing as a joint link and dialogue act classification task, by using CRFSGD (Bottou, 2011) and MaltParser (Nivre et al., 2007). They also demonstrated that the methods they use for thread discourse structure parsing are able to perform equally well over partial threads as complete threads, by experimenting with "in situ" classification of evolving threads.

There is also research focusing on particular types of dialogue acts, such as question–answer pairs in emails (Shrestha and McKeown, 2004) and forum threads (Cong et al., 2008), question–context–answer in forum threads (Cong et al., 2008; Ding et al., 2008; Cao et al., 2009), initiation–response pairs (e.g. question–answer, assessment–agreement, and blame–denial) in forum threads (Wang and Rosé, 2010), as well as request and commitment in emails (Lampert et al., 2007, 2008a,b, 2010).

Thread discourse structure can be used to facilitate different tasks in web user forums. For example, threading information has been shown to enhance retrieval effectiveness for post-level retrieval (Xi et al., 2004; Seo et al., 2009), thread-level retrieval (Seo et al., 2009; Elsas and Carbonell, 2009), sentence-level shallow information extraction (Sondhi et al., 2010), and near-duplicate thread detection (Muthmann et al., 2009). Moreover Wang and Rosé (2010) demonstrated that initiation–response pairs (e.g. question–answer, assessment–agreement, and blame–denial) from online forums have the potential to enhance thread summarisation and automatically generate knowledge bases for Community Question Answering (cQA) services such as Yahoo! Answers. Furthermore, Kim et al. (2006) showed that dialogue acts can be used to classify student online discussions in web-enhanced courses. Specifically, they use dialogue acts to identify discussion threads that may have unanswered questions and need the attention of an instructor.

These previous research efforts suggest that the thread structural representation used in Wang et al. (2011), which includes both linking structure and the dialogue act associated with each link, could potentially provide even greater leverage in these tasks. We specifically target the thread-level task of Solvedness classification as it is conceptually the most difficult of the three classification tasks proposed by Baldwin et al. (2007), and intuitively, it is the task which stands to gain the most from discourse parsing.

[1] http://www.csse.unimelb.edu.au/research/lt/resources/conll2010-thread/

3 Data Description

We use ILIAD (Improved Linux Information Access by Data Mining) data set created by Baldwin et al. (2007), which contains threads crawled from Linuxquestions[2] and Debian mailing lists.[3] The ILIAD data set is made up of 250 threads, which are annotated with three Boolean thread-level labels, namely *Task Orientation*, *Completeness* and *Solvedness*. This paper only explores the classification task of Solvedness, which addresses the question of "is there a documented solution to the original problem described by the thread initiator in the thread (including the possibility of URLs pointing off to solutions elsewhere on that same forum or generally on the web)?" A detailed description of the other tasks and the dataset is presented in Baldwin et al. (2007).

To explore the task of using discourse structure to predict the Solvedness of a thread, we annotated the ILIAD threads for discourse structure based on the dialogue act set proposed by Kim et al. (2010b). The dialogue act set is made up of 5 super-categories: Question, Answer, Resolution, Reproduction and Other. The Question category contains 4 sub-classes: question, add, confirmation and correction. Similarly, the Answer category contains 5 sub-classes: answer, add, confirmation, correction and objection. For example, the label Question-add signifies the Question superclass and add subclass, i.e. addition of extra information to a question. For full details of the original dialogue act tagset, see Kim et al. (2010b).

The original dialogue act set was developed primarily over troubleshooting-oriented threads (i.e. the CNET dataset described in Section 2), however there are non-troubleshooting threads present in the ILIAD dataset (hence the *Task Orientation* thread classification task is addressed in Baldwin et al. (2007)). After manual analysis of the ILIAD data, we identified that the dialogue act tagset was largely transferable in its original state, but needed the addition of the information sub-class to the Question super-category (i.e. Question-information). Question-information is used on posts in threads which are not troubleshooting-oriented and only provide information (e.g. on developer mailing lists to report on a bug fix). We also relaxed the definition of Resolution slightly to accommodate non-troubleshooting threads. For example, in one thread the initiator requests an update to a wiki page, and this update is confirmed later by a non-initiator. In this case, this non-initiator's post is labelled as Resolution. In the original definition, Resolution can only be used on posts from the initiator of the thread.

The modified dialogue acts (DAs) used to annotate the ILIAD dataset for discourse structure are described in Table 1. The annotation was performed by two annotators. The main annotator annotated all 250 threads (containing 1158 posts), and the secondary annotator independently annotated 26 randomly-selected threads (containing 113 posts) for quality assurance purposes. During annotation, annotators first annotate the Links between posts in a thread, and then identify the type of each link (DA). The κ values for agreement between the two annotators are 0.64 for combined Link and DA tagging, 0.79 for just the Links and 0.68 for just the DAs.

While both the ILIAD and CNET datasets are mainly troubleshooting-oriented and technical, they come from different domains. Therefore, we expect the DA and Link distributions of them to be different. However, to our surprise, the distributions of both DAs and Links in the two datasets are remarkably similar, supporting the suggestion that the DA label set has cross-domain applicability.

[2] http://www.linuxquestions.org

[3] http://lists.debian.org/completeindex.html

Super-category	Sub-class	Description
Question	question	the post contains a new and independent question.
	add	the post provides additional information or asks a follow-up question, regarding a previous question.
	confirmation	the post confirms details or errors in a question.
	correction	the post corrects errors in a question.
	information*	the post is in a non-troubleshooting thread, and only provides information.
Answer	answer	the post proposes an answer to a question.
	add	the post provides additional information to an answer.
	confirmation	the post confirms details or errors in an answer.
	correction	the post corrects errors in an answer.
	objection	the post objects to an answer.
Resolution	—	a user confirms that an answer works.*
Reproduction	—	a non-initiator asks a similar question, or confirms that an answer should work.
Other	—	the post does not belong to any of the above classes.

Table 1: The Dialogue Act (DA) set used for annotating ILIAD dataset. ("*" signifies a difference over the original DA proposed by Kim et al. (2010b))

Regarding the Solvedness label for ILIAD dataset, the original thread-level annotations were done by three annotators are on a five-point scale, with 1 indicating high confidence in Solvedness for a given thread and 5 indicating low confidence (Baldwin et al., 2007). These annotations were aggregated by taking the simple mean across the three annotators and discretising into binary classes, with 2.5 as the breakpoint. Out of the 250 threads in the ILIAD dataset, 28 threads had a score of 2.5 and were discarded in the original paper. In the interests of comparability with the original research, we experiment over this reduced dataset (denoted ILIAD_{222}), but question the theoretical soundness of removing these threads from the dataset, so additionally experiment with the full dataset (denoted ILIAD).

4 Discourse Structure Parsing for Thread Solvedness Classification

Baldwin et al. (2007) showed that the task of automatic Solvedness classification on ILIAD_{222} is extremely hard. This is because the annotation was often based on expert knowledge of Linux, and a great deal of information not explicitly mentioned in the thread. Take the thread in Figure 1 for example: although two independent answers are provided in Post2 and Post5, it is almost impossible to identify whether there is a correct solution unless the whole thread is understood at a technical level.

Although predicting Solvedness is challenging, we believe that the use of thread discourse structure should assist in the task. As a first step, we need to do thread discourse structure parsing, which includes predicting both the linkings (Links) between posts and the type (DA) of each link.

Discourse structure parsing, as discussed in Wang et al. (2011), can be addressed in several ways. If a structured classification approach, such as Conditional Random Fields (CRFs: Lafferty et al. (2001)), is used, we can either classify the Link and DA separately and compose them afterwards (denoted as Composition), or we can classify the combined Link and DA (e.g. treat 0+Question-question as a single label) directly (denoted as Combine). Another approach is

Feature Category	Feature Name	Description
DA-only	LastPostDA	The DA of the last post in the thread.
	LastNonInitDA	The DA of the last post from a non-initiator in the thread.
	HasResolution	Whether the thread contains a Resolution post.
LinkDA-based	LastPairDA	The DA pair for the deepest post pair in the thread tree. In the case of ties, the pair containing the latest post is chosen.
	LastSubthreadDA	The sequence of DAs in the longest subthread in the thread tree. In case of ties, the sequence containing the latest post is chosen.

Table 2: Thread discourse structure features used for Solvedness classification.

to treat discourse structure parsing as a dependency parsing problem. Dependency parsing (Kübler et al., 2009) is the task of automatically predicting the dependency structure of a token sequence, in the form of binary asymmetric dependency relations with dependency types. The joint classification task of Link and DA is a natural fit for dependency parsing, in that the task is intrinsically one of inferring labelled dependencies between posts.

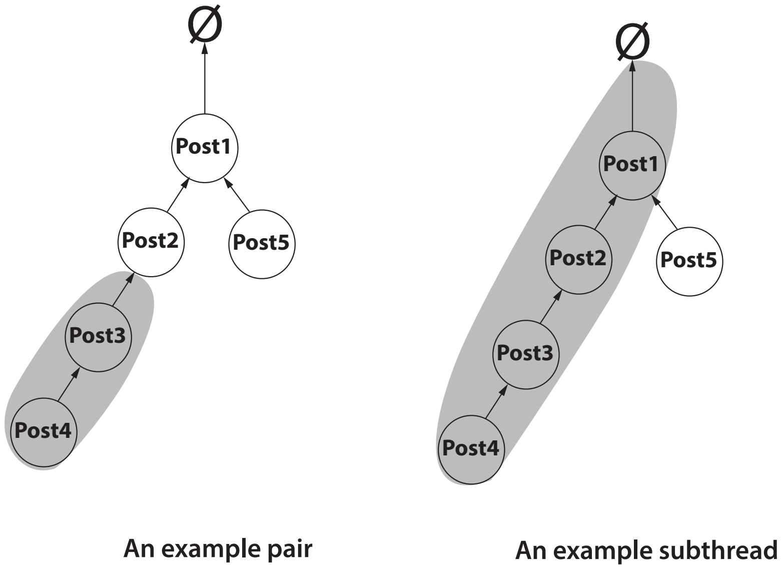
For discourse structure parsing, all experiments were carried out based on 10-fold cross-validation, stratifying at the thread level to ensure that all poss from a given thread occur in a single fold. The results are evaluated using post-level micro-averaged F-score ($\beta = 1$). All three discourse structure parsing methods were tested in our experiments, by using CRFSGD (Bottou, 2011) and MaltParser (Nivre et al., 2007). As for features, we experimented with all the features proposed by Wang et al. (2011), such as Initiator, Position, TitSim, PostSim, Punct and UserProf, as well as many of our own features. We found that using CRFSGD with a simple Initiator (i.e. whether a post's author is the initiator of the thread) feature and the Combine approach achieves the highest Link and DA joint (LinkDA) classification F-score of 0.626. This is significantly better[4] than a strong heuristic baseline which classifies all first posts as 0+Question-question and all subsequent posts as 1+Answer-answer, which achieves a joint classification F-score of 0.420.

We also explored the possibility of domain adaptation, by using threads from the CNET dataset of Kim et al. (2010b) to augment the ILIAD thread discourse structure parsing. However, we were unable to achieve any significant improvements.

When using the thread discourse structure (i.e. Link and DA) for Solvedness prediction, one natural question is "could we simply use Resolution to identify solved thread?" While Resolution is a clear identifier of solved threads with 100% precision, only 8% of the threads contain Resolution posts, and yet 80.4% of the threads are labelled as solved. Therefore, by only using Resolution, a classifier could not do better than a majority class baseline. Instead, we propose to use a combination of discourse structure features to address the Solvedness classification problem. Table 2 displays all the discourse structure features[5] used in this paper, which can be grouped into two categories: (1) those based on only the DAs (DA-only); and (2) those based

[4] All statistical significance results in the paper are based on randomised estimation (Yeh, 2000), at a significance level of $p < 0.05$.

[5] We have experimented with many "discourse structure" features and non-"discourse structure" features, and these are particularly useful and interesting.



An example pair

An example subthread

Figure 2: Examples of a pair and a subthread in a thread.

on Link and DA (LinkDA-based). When using the Link information, we rely on the notions of "pairing" and "subthreading". A pair is defined to be the combination of a post with the parent post it links to (noting that a given post can participate as child in multiple "pairs" as it can link to multiple posts), and a subthread contains all posts in a given path from a leaf node to the root node following the link structure. Figure 2 shows an example of each, based on the sample thread from Figure 1. As the `LastPostDA` is Answer-answer from Post5 (the final post), the `LastNonInitDA` for the thread is Answer-answer, `HasResolution` is 0 (as there are no Resolution posts), `LastPairDA` is Answer-add/Answer-confirmation from the pairing of Post4 and Post3, and `LastSubthreadDA` is Answer-add/Answer-confirmation/Answer-answer/Question-question from the subthread Post4/Post3/Post2/Post1.

5 Solvedness Classification

Baldwin et al. (2007) experimented with various learners from three machine learning software packages, namely LIBSVM (Chang and Lin, 2011), TiMBL (Daelemans et al., 2010) and Weka (Hall et al., 2009), and found that LIBSVM performs superiorly on the Solvedness classification task. Therefore, LIBSVM is used for Solvedness classification in this research.

In our initial experiments, we experimented with different kernel functions for LIBSVM, including linear, polynomial, radial basis function (RBF) and sigmoid kernels, and found the linear kernel to outperform other kernels. Therefore, LIBSVM with linear kernel is used throughout our experiments. We approach the Solvedness classification task by firstly following the procedure of Baldwin et al. (2007), where ILIAD$_{222}$ is used. Subsequently, we carry out experiments over the full 250-thread ILIAD dataset. In both cases, various combinations of the features introduced in Section 4 are used. To generate these features, both the gold-standard LinkDAs and the automatically predicted ones are used.

All our Solvedness classification experiments were carried out based on stratified 10-fold cross-validation. The results are evaluated using classification accuracy (ACC). As our baselines, we use a majority classifier (ZeroR), as well as the best Solvedness classifier provided by Baldwin et al. (2007) (ADCS). As mentioned earlier, randomised estimation (Yeh, 2000) (at a significance level of $p < 0.05$) is used throughout the paper for statistical significance testing.

Feature Category	System/feature(s)	ACC_{gold}	ACC_{auto}
Baseline	ZeroR	.779	
	ADCS	.788	
DA-only	LastPostDA	.833*	.775
	LastNonInitDA	.766	**.792**
	HasResolution	.779	.779
	LastPostDA +LastNonInitDA	.834*	.779
	LastPostDA +HasResolution	**.883***	.775
	LastNonInitDA +HasResolution	.874*	**.792**
	AllDAFeat	**.883***	.779
LinkDA-based	LastPairDA	.851*	**.792**
	LastSubthreadDA	.833*	.779
	AllLinkDAFeat	.833*	**.792**
	AllDAFeat +AllLinkDAFeat	.865*	**.792**

Table 3: Results over ILIAD_{222}, using discourse structure features from the gold-standard and also the discourse parsing model ("*" signifies a significantly better result than both baselines; the best result in each column is indicated in **boldface**).

5.1 Experiments over ILIAD_{222}

Table 3 presents the results from experiments over ILIAD_{222}, using the thread discourse structure features generated from both the gold-standard (ACC_{gold}) LinkDAs, and automatically predicted ones (ACC_{auto}). The automatically predicted discourse structure of the whole ILIAD_{222} dataset is obtained by aggregating the discourse structure predictions from each fold of the 10-fold cross-validation experiments described in Section 4. The combination of all DA-only features (i.e. LastPostDA, LastNonInitDA and HasResolution) is denoted AllDAFeat, and the combination of all LinkDA-based features (i.e. LastPairDA and LastSubthreadDA) is denoted AllLinkDAFeat. Results which are significantly better than both baseline results are signified by "*", and the best result(s) in each column are presented in **boldface**.

Looking first at the ACC_{gold} results in Table 3 we can see that, not surprisingly, HasResolution by itself does not have any effect on the prediction (see our comments in Section 4). Moreover, while LastPostDA leads to a significant improvement, LastNonInitDA does not have a significant effect. More interestingly, the combination of LastPostDA or LastNonInitDA with HasResolution leads to further improvements. This is because the classifiers trained on LastPostDA or LastNonInitDA are aggressive and misclassify many solved threads as unsolved, which HasResolution can correct.

The ACC_{gold} column also shows both the potential and shortcomings of LinkDA-based features — i.e. while both LastPairDA and LastSubthreadDA lead to significantly better results in isolation, combining them does not lead to further improvements. Moreover, combining all the features (i.e. AllDAFeat +AllLinkDAFeat) leads to a drop in results compared to just using AllDAFeat. We hypothesise that there are a number of reasons for this. Firstly, the LastPairDA, LastSubthreadDA and DA-based features have dependencies between each other, in that they all draw on the same set of DAs. While they are closely related, the classifiers do not have any access into the internals of the features to leverage them, causing the learner to overfit the training data. Secondly, while LastPairDA and LastSubthreadDA lead to low results in isolation, this is almost certainly because of the sparse nature (LastPairDA and

Feature Category	System/feature(s)	ACC_{gold}	ACC_{auto}
Baseline	ZeroR	.804	
	ADCS	.804	
DA-only	LastPostDA	.784	.780
	LastNonInitDA	.792	.788
	HasResolution	.804	.804
	LastPostDA +LastNonInitDA	.848*	.776
	LastPostDA +HasResolution	.864*	.780
	LastNonInitDA +HasResolution	.872*	.788
	AllDAFeat	**.884***	.776
LinkDA-based	LastPairDA	.832	**.816**
	LastSubthreadDA	.832	.792
	AllLinkDAFeat	.824	.792
	AllDAFeat +AllLinkDAFeat	.852*	.792

Table 4: Results over ILIAD, using discourse structure features from the gold-standard and also the discourse parsing model ("*" signifies a significantly better result than both baselines; the best result in each column is indicated in **boldface**).

LastSubthreadDA have 72 and 135 distinct values, respectively), much moreso than the DA-based features. When combined with the other features, however, some of these features are found to have utility.

Looking next to the ACC_{auto} results in Table 3, we can see that we surpass the two baselines, delivering on the promise of discourse parsing aiding in Solvedness classification. The results drop appreciably relative to those achieved with the gold-standard labels, and in fact the improvements over the baselines aren't statistically significant. This is perhaps not surprising, however, given than the F-score for discourse parsing was a modest 0.626, meaning that errors will propagate through to the thread-level classification.

While these results are certainly encouraging, and were worthwhile in terms of establishing the superiority of our method when discourse parsing features are used, we always had reservations about the ILIAD$_{222}$ data set, due to the most contentious instances having been removed from the data set. In introducing these instances back into the data set and labelling them as solved, the task becomes both more realistic and more challenging, including the ZeroR baseline rising up further. In the next section, we reapply our methods to the ILIAD data set.

5.2 Experiments over ILIAD

We carry out the same experiments done in Section 5.1 over the whole ILIAD data set, and present the results in Table 4. Again, the results which are significantly better than both baseline results are signified by "*", and the best result in each column is presented in **boldface**.

From Table 4 we can see a similar trend to that in Section 5.1, with our method improving over both baselines when we use either gold-standard or automatically-predicted features. However, there are some notable differences. Looking first at the ACC_{gold} column, firstly, none of LastPostDA, LastNonInitDA and HasResolution led to any improvement in isolation. However, the combination of these three features led to results that are significantly better than the baselines, with AllDAFeat achieving the best result of 0.883. Secondly, neither LastPairDA nor LastSubthreadDA has a significant impact on results, and their combination

DA	LastPostDA	LastNonInitDA	HasResolution	AllDAFeat
Question-add	0.999	0.811	—	0.999
Question-confirmation	0.881	0.991	—	0.961
Question-information	0.918	—	—	0.918
Answer-answer	0.500	0.513	—	0.498
Answer-add	0.461	0.550	—	0.489
Answer-confirmation	0.918	—	—	0.918
Answer-objection	0.918	1.000	—	0.954
Reproduction	1.000	1.000	—	1.000
Resolution	0.332	0.918	0.229	0.237
Other	0.971	0.934	—	0.952

Table 5: Entropy of each DA against the Solvedness class distribution for every DA-only feature and AllDAFeat features.

(i.e. AllLinkDAFeat) also does not outperform the baselines significantly. Looking next to ACC_{auto}, we achieve the best results with LastPairDA once again, surpassing the baselines but not at a level of statistical significance. Overall, while it is clear that the Solvedness classification task becomes harder when we experiment with the full ILIAD dataset, we were able to reproduce the overall results from Section 5.1.

6 Results Analysis and Simulation

Examining the differences between the results for ACC_{gold} and ACC_{auto} in Section 5.1 and Section 5.2 leads us to suspect that if the F-score of the thread discourse parsing could be boosted, we would be able to achieve better Solvedness classification accuracy. Furthermore, because the most effective discourse structure features, i.e. LastPostDA, LastNonInitDA and HasResolution, only make use of a subset of the DAs, we anticipate that if we can improve the F-score over certain DAs, we will be able to significantly boost our Solvedness classification accuracy.

To test these hypotheses, firstly, we examine the entropy (presented in Table 5) of every DA against the Solvedness class distribution for each DA-only feature (i.e. LastPostDA, LastNonInitDA and HasResolution) and the combination of all DA-only features (i.e. AllDAFeat). From Table 5, we can see that Answer-answer, Answer-add and Resolution have relatively low entropy compared to the rest of the DAs. Therefore, it seems that these three DAs can contribute more in Solvedness classification.[6]

Secondly, we conducted simulation experiments to examine the potential relation between DA classification and Solvedness classification. The simulation starts with a seed DA classification result (SeedResults), based on CRFSGD and the Initiator feature. This seed DA classification achieves a F-score of 0.651, significantly better than a strong heuristic baseline (i.e. 0.515) which classifies all first posts as Question-question and all subsequent posts as Answer-answer. Then, an arbitrary higher goal (e.g. 0.8) is set and an artificial classification result (ArtificialResults) is created by randomly correcting errors in the output of the discourse parsing model. The corrections are made evenly across all DA labels, relative to the original error rates for each DA. Next, a simulator is used to predict the labels of each instance, by randomly selecting from the

[6] Note that this entropy analysis can only capture the association between a single DA and the Solvedness class, and we are not able to capture more subtle feature interactions.

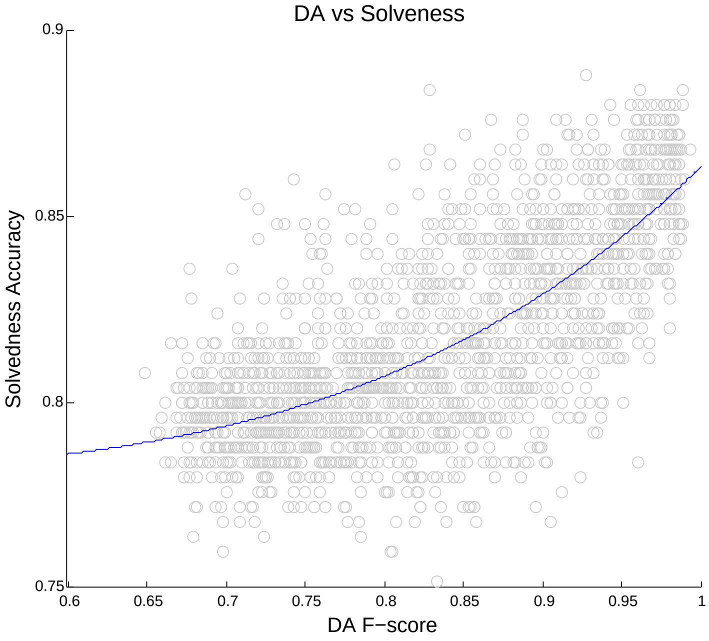


Figure 3: Simulation over all DAs (AllDA).

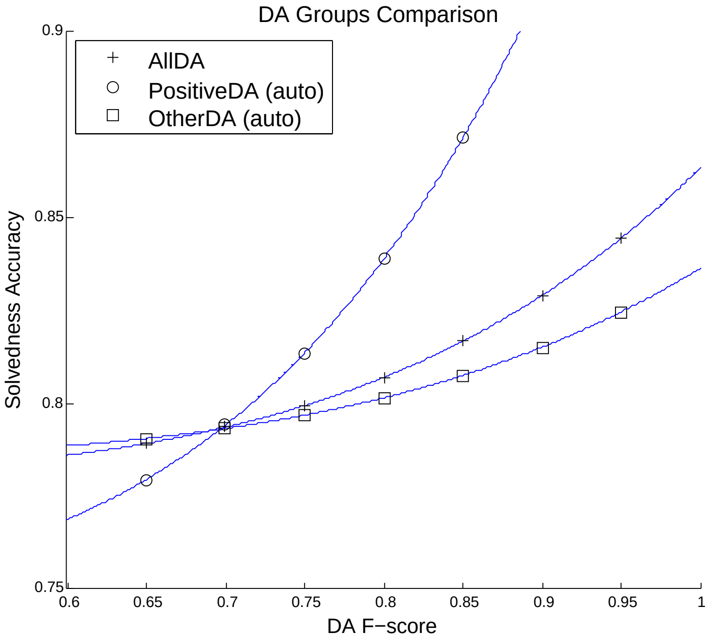


Figure 4: Simulation over automatically-generated DA groups (PositiveDA$_{auto}$ and OtherDA$_{auto}$).

labels returned by SeedResults and ArtificialResults with equal chance. In order to generate enough simulated results, we pick 20 goal F-score figures between 0.651 and 1.0, and run the simulator 100 times for each of these figures. Finally, we use these 2000 simulated discourse structure predictions to classify Solvedness using `AllDAFeat` features, and plot each pair of discourse structure F-score and Solvedness accuracy in a scatter plot. We also try to fit a series of simple polynomial models of the form $y = ax^n + b$ $(n \in \{1, 2, 3, 4, 5\})$[7] to the plot. We find that the model for $y = ax^5 + b$ provides the best fit with the data, although the differences in the range $n \in \{2, 3, 4, 5\}$ are negligible. Figure 3 shows the graph, along with the curve of best fit for the function $y = ax^5 + b$.

From Figure 3 we can see that there is a clear correlation between the F-score of DA classification and the accuracy of Solvedness classification, and that the impact of DA classification on Solvedness classification is, in fact, accentuated for higher F-scores. Theoretically, therefore, by improving the DA classification F-score, the Solvedness classification accuracy will increase accordingly.

The entropy analysis showed that not all DAs have the same utility for the task of Solvedness classification — i.e. some DAs are more important (lower entropy) than others. We select the three DAs (i.e. Answer-answer, Answer-add and Resolution) with lowest entropy values from Table 5, because these DAs seem to be the most effective across the three feature types (i.e. `LastPostDA`, `LastNonInitDA` and `HasResolution`). Then, we carry out an analogous simulation over this set of automatically-selected DAs (PositiveDA$_{auto}$). Additionally, we conducted a simulation over the 8 non-selected DAs (OtherDA$_{auto}$).[8] Once again, a line of best fit for $y = ax^5 + b$ is generated for the resulting simulations. The curves of best fit are shown in Figure 4, along with the original curve of best fit for all DAs (AllDA) from Figure 3.

[7] Choosing $n > 5$ does not result in better fit with the data.

[8] Question-correction and Answer-correction are never used in annotating the discourse structure of ILIAD data set.

DA Group	DA	F-score
PositiveDA$_{auto}$	Answer-answer	0.782
	Answer-add	0.641
	Resolution	0.514
OtherDA$_{auto}$	Question-question	0.992
	Question-add	0.678
	Question-confirmation	0
	Question-information	0
	Answer-confirmation	0
	Answer-objection	0
	Reproduction	0
	Other	0

Table 6: Micro-averaged DA classification F-scores per DA over ILIAD

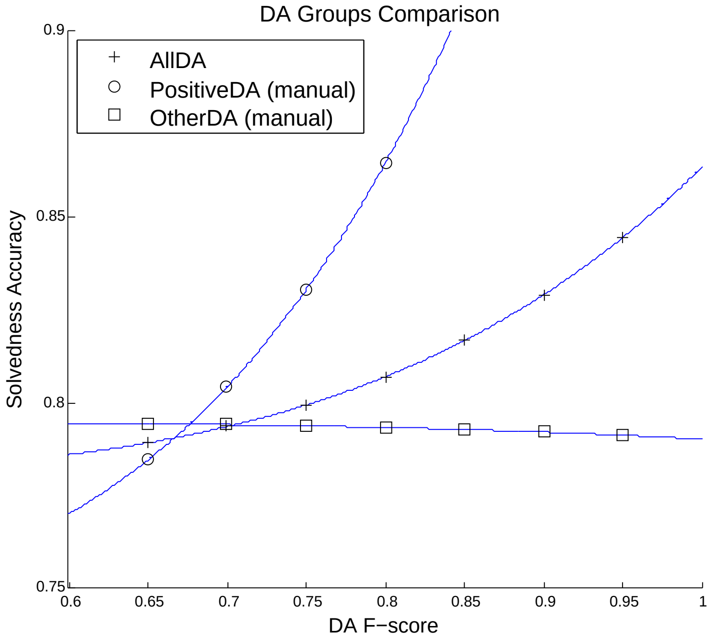


Figure 5: Simulation over manually-created DA groups (PositiveDA$_{manual}$ and OtherDA$_{manual}$).

From Figure 4 we can see that, as suspected, the PositiveDA$_{auto}$ group is much more important than the OtherDA$_{auto}$ group for Solvedness classification. Therefore, to improve Solvedness classification, we should focus our attention on improving the DA classification F-score for DAs such as Answer-answer, Answer-add and Resolution. Table 6 shows the micro-averaged F-scores of DA classification, calculated per DA. When we do a breakdown of the results for the discourse parsing model, we can see that there is definitely room for improvement with Answer-answer, Answer-add and Resolution. Moreover, Answer-answer and Answer-add are the most-frequent and third most-frequent DAs in the ILIAD data set, respectively, appearing 354 and 147 times. Therefore, there appears to be considerable scope for improvement.

While the identification of the more important DAs can be done automatically as shown above, we also attempted to select them in a more ad hoc way, based on our understanding and analysis of the data set. Intuitively, if a thread's last post or the last post from a non-initiator is Question-confirmation, Question-information, Answer-confirmation or Answer-objection, this thread is more likely to be unresolved. At the same time, we can observe that the microaverage F-scores for all these DAs are 0, that is the model never predicts a post to be one of these DA types correctly. To explore the utility of these additional DAs, we conducted an additional simulation experiment including Question-confirmation, Question-information, Answer-confirmation, Answer-objection and Resolution in PositiveDA$_{manual}$, and relegating the other 6 DAs to OtherDA$_{manual}$. The results for these manually-created groupings are shown in Figure 5.

From Figure 5 we can see that the improvements in discourse parsing over the manually-chosen PositiveDA$_{manual}$ will lead to even greater improvements over Solvedness prediction than before, if only we can get the models to make predictions using them. Perhaps even more surprising is that our simulations predict that improvements over OtherDA$_{manual}$ stand to *degrade* Solvedness classification. These findings set the direction for future work on improving the F-score of the discourse parser.

7 Conclusions and Future Work

In this research, we explore the task of Solvedness classification, that is the automatic prediction of whether the information need on the part of the initiator of a thread has been resolved or not, by parsing thread discourse structure in the form of a rooted directed acyclic graph over posts, with edges labelled with dialogue acts. While Solvedness classification has been shown to be very difficult in previous research (Baldwin et al., 2007), we achieve significantly better results using gold-standard discourse structure. We are also able to attain improvements in Solvedness classification accuracy using automatically-predicted thread discourse structure, although not at a level of statistical significance. However, simulations suggest that as we improve the F-score of thread discourse structure parsing, the Solvedness classification accuracy will increase disproportionately. Additionally, we showed that a particular subset of DAs is crucial to Solvedness classification accuracy, and that if we can improve the F-score of our discourse structure predictions over these DAs, we stand to make large gains in Solvedness classification accuracy.

In future work, we plan to firstly investigate ways to improve the discourse parser F-score over the PositiveDA$_{auto}$ and PositiveDA$_{manual}$ sets of DAs. Moreover, we plan to delve further into feature engineering, looking at other means of capturing thread discourse structure. Additionally, although our preliminary experiments on using CNET annotated threads to help ILIAD discourse structure parsing were not positive, it would be interesting to investigate the effect of using CNET threads in predicting specific DAs for the ILIAD data set.

Acknowledgements

NICTA is funded by the Australian government as represented by Department of Broadband, Communication and Digital Economy, and the Australian Research Council through the ICT Centre of Excellence programme.

References

Baldwin, T., Martinez, D., and Penman, R. B. (2007). Automatic thread classification for Linux user forum information access. In *Proceedings of the 12th Australasian Document Computing Symposium (ADCS 2007)*, pages 72–79, Melbourne, Australia.

Bottou, L. (2011). CRFSGD software. `http://leon.bottou.org/projects/sgd`.

Cao, X., Cong, G., Cui, B., Jensen, C. S., and Zhang, C. (2009). The use of categorization information in language models for question retrieval. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM 2009)*, pages 265–274, Hong Kong, China.

Chang, C.-C. and Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27:1–27:27. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Cong, G., Wang, L., Lin, C.-Y., Song, Y.-I., and Sun, Y. (2008). Finding question-answer pairs from online forums. In *Proceedings of 31st International ACM-SIGIR Conference on Research and Development in Information Retrieval (SIGIR'08)*, pages 467–474, Singapore.

Daelemans, W., Zavrel, J., Van der Sloot, K., and Van den Bosch, A. (2010). TiMBL: Tilburg memory based learner, version 6.3, API guide. *ILK Research Group Technical Report Series*, 03(10). Software available at `http://ilk.uvt.nl/timbl/`.

Ding, S., Cong, G., Lin, C.-Y., and Zhu, X. (2008). Using conditional random fields to extract context and answers of questions from online forums. In *Proceedings of the 46th Annual Meeting of the ACL: HLT (ACL 2008)*, pages 710–718, Columbus, USA.

Elsas, J. L. and Carbonell, J. G. (2009). It pays to be picky: An evaluation of thread retrieval in online forums. In *Proceedings of 32nd International ACM-SIGIR Conference on Research and Development in Information Retrieval (SIGIR'09)*, pages 714–715, Boston, USA.

Fortuna, B., Rodrigues, E. M., and Milic-Frayling, N. (2007). Improving the classification of newsgroup messages through social network analysis. In *Proceedings of the 16th ACM Conference on Information and Knowledge Management (CIKM 2007)*, pages 877–880, Lisbon, Portugal.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The WEKA data mining software: An update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18.

Kim, J., Chern, G., Feng, D., Shaw, E., and Hovy, E. (2006). Mining and assessing discussions on the web through speech act analysis. In *Proceedings of the ISWC'06 Workshop on Web Content Mining with Human Language Technologies*, Athens, USA.

Kim, S. N., Cavedon, L., and Baldwin, T. (2010a). Classifying dialogue acts in one-on-one live chats. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP 2010)*, pages 862–871, Boston, USA.

Kim, S. N., Wang, L., and Baldwin, T. (2010b). Tagging and linking web forum posts. In *Proceedings of the 14th Conference on Computational Natural Language Learning (CoNLL-2010)*, pages 192–202, Uppsala, Sweden.

Kübler, S., McDonald, R., and Nivre, J. (2009). Dependency parsing. *Synthesis Lectures on Human Language Technologies*, 2(1):1–127.

Lafferty, J., McCallum, A., and Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning*, pages 282–289, Williamstown, USA.

Lampert, A., Dale, R., and Paris, C. (2008a). The nature of requests and commitments in email messages. In *Proceedings of the AAAI 2008 Workshop on Enhanced Messaging*, pages 42–47, Chicago, USA.

Lampert, A., Dale, R., and Paris, C. (2008b). Requests and commitments in email are more complex than you think: Eight reasons to be cautious. In *Proceedings of the Australasian Language Technology Association Workshop 2008 (ALTA 2008)*, pages 64–72, Hobart, Australia.

Lampert, A., Dale, R., and Paris, C. (2010). Detecting emails containing requests for action. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL HLT 2010)*, pages 984–992, Los Angeles, California.

Lampert, A., Paris, C., and Dale, R. (2007). Can requests-for-action and commitments-to-act be reliably identified in email messages? In *Proceedings of the Twelfth Australasian Document Computing Symposium (ADCS 2007)*, pages 48–55, Melbourne, Australia.

Muthmann, K., Barczyński, W. M., Brauer, F., and Löser, A. (2009). Near-duplicate detection for web-forums. In *Proceedings of the 2009 International Database Engineering & Applications Symposium (IDEAS 2009)*, pages 142–151, Cetraro, Italy.

Nivre, J., Hall, J., Nilsson, J., Chanev, A., Eryigit, G., Kübler, S., Marinov, S., and Marsi, E. (2007). MaltParser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(02):95–135.

Seo, J., Croft, W. B., and Smith, D. A. (2009). Online community search using thread structure. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM 2009)*, pages 1907–1910, Hong Kong, China.

Shrestha, L. and McKeown, K. (2004). Detection of question-answer pairs in email conversations. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004)*, pages 889–895, Geneva, Switzerland.

Sondhi, P., Gupta, M., Zhai, C., and Hockenmaier, J. (2010). Shallow information extraction from medical forum data. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010), Posters Volume*, pages 1158–1166, Beijing, China.

Wang, L., Lui, M., Kim, S. N., Nivre, J., and Baldwin, T. (2011). Predicting thread discourse structure over technical web forums. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 13–25, Edinburgh, UK.

Wang, Y.-C. and Rosé, C. P. (2010). Making conversational structure explicit: identification of initiation-response pairs within online discussions. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL HLT 2010)*, pages 673–676.

Xi, W., Lind, J., and Brill, E. (2004). Learning effective ranking functions for newsgroup search. In *Proceedings of 27th International ACM-SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2004)*, pages 394–401, Sheffield, UK.

Yeh, A. (2000). More accurate tests for the statistical significance of result differences. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING 2000)*, pages 947–953, Saarbrücken, Germany.