Predicting Thread Discourse Structure over Technical Web Forums

Li Wang,[♠♡] **Marco Lui,**[♠♡] **Su Nam Kim,**[♠♡] **Joakim Nivre**[♢] and **Timothy Baldwin**[♠♡]
♠ Dept. of Computer Science and Software Engineering, University of Melbourne
♡ NICTA Victoria Research Laboratory
♢ Dept. of Linguistics and Philology, Uppsala University
li.wang.d@gmail.com, saffsd@gmail.com,
sunamkim@gmail.com, joakim.nivre@lingfil.uu.se, 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. The discourse structure of web forum threads, in the form of labelled dependency relationships between posts, has the potential to greatly improve information access over web forum archives. In this paper, we present the task of parsing user forum threads to determine the labelled dependencies between posts. Three methods, including a dependency parsing approach, are proposed to jointly classify the links (relationships) between posts and the dialogue act (type) of each link. The proposed methods significantly surpass an informed baseline. We also experiment with "in situ" classification of evolving threads, and establish that our best methods are able to perform equivalently well over partial threads as complete threads.

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.

This research aims at enhancing information access and support sharing, by mining the discourse structure of troubleshooting-oriented web user forum threads. Previous research has shown that simple thread structure information (e.g. reply-to structure) can enhance tasks such as forum information retrieval (Seo et al., 2009) and post quality assessment (Lui and Baldwin, 2009). We aim to move beyond simple threading, to predict not only the links between posts, but also show the manner of each link, in the form of the discourse structure of the thread. In doing so, we hope to be able to perform richer visualisation of thread structure (e.g. highlighting the key posts which appear to have led to a successful resolution to a problem), and more fine-grained weighting of posts in threads for search purposes.

To illustrate the task, we use an example thread, made up of 5 posts from 4 distinct participants, from the CNET forum dataset of Kim et al. (2010b), as shown in Figure 1. The discourse structure of the thread is modelled as a rooted directed acyclic graph

Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pages 13–25,
Edinburgh, Scotland, UK, July 27–31, 2011. ©2011 Association for Computational Linguistics

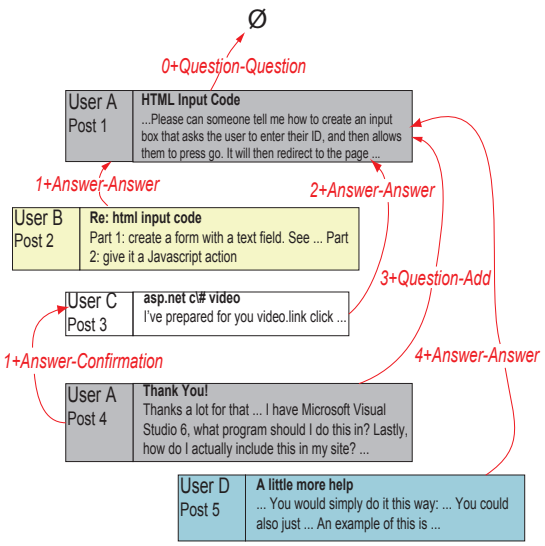

Figure 1: A snippeted and annotated CNET thread

(DAG) with a dialogue act label associated with each edge of the graph. In this example, UserA initiates the thread with a question (dialogue act = Question-Question) in the first post, by asking how to create an interactive input box on a webpage. In response, UserB and UserC provide independent answers (dialogue act = Answer-Answer). UserA responds to UserC to confirm the details of the solution (dialogue act = Answer-Confirmation), and at the same time, adds extra information to his/her original question (dialogue act = Question-Add); i.e., this one post has two distinct dependency links associated with it. Finally, UserD proposes a different solution again to the original question.

To predict thread discourse structure of this type, we jointly classify the links and dialogue acts between posts, experimenting with a variety of supervised classification methods, namely dependency parsing and linear-chain conditional random fields. In this, we build on the earlier work of Kim et al. (2010b) who first proposed the task of thread discourse analysis, but only carried out experiments on post linking and post dialogue act classification as separate tasks. In addition to achieving state-of-the-art accuracy over the task, we carry out in-depth analysis of classification effectiveness at different thread depths, and establish that the accuracy of our method over partial threads is equivalent to that over full threads, indicating that the method is applicable to in-situ thread classification. Finally, we investigate the role of user-level features in discourse structure analysis.

2 Related Work

This work builds directly on earlier work of a subset of the authors (Kim et al., 2010b), whereby a novel post-level dialogue act set was proposed, and used as the basis for annotation of a set of threads taken from CNET. In the original work, we proposed a set of novel features, which we applied to the separate tasks of post link classification and dialogue act classification. We 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, we tackled only a single task, either link classification (optionally given dialogue act tags) or dialogue act classification, but never the two together. In this paper, we take the obvious step of exploring joint classification of post link and dialogue act tags, to generate full thread discourse structures.

Discourse disentanglement (i.e. link classification) and dialogue act tagging have been studied largely as independent tasks. Discourse disentanglement is the task of dividing a conversation thread (Elsner and Charniak, 2008; Lemon et al., 2002) or document thread (Wolf and Gibson, 2005) into a set of distinct sub-discourses. The disentangled discourse is sometimes assumed to take the form of a tree structure (Grosz and Sidner, 1986; Lemon et al., 2002; Seo et al., 2009), an acyclic graph structure (Rosé et al., 1995; Schuth et al., 2007; Elsner and Charniak, 2008; Wang et al., 2008; Lin et al., 2009), or a more general cyclic chain graph structure (Wolf and Gibson, 2005). Dialogue acts are used to describe the function or role of an utterance in a discourse, and have been applied to the analysis of mediums of communication including conversational speech (Stolcke et al., 2000; Shriberg et al., 2004; Murray et al., 2006), email (Cohen et al., 2004; Carvalho and Cohen, 2005; Lampert et al., 2008), instant messaging (Ivanovic, 2008; Kim et al., 2010a), edited documents (Soricut and Marcu, 2003; Sagae, 2009) and online forums (Xi et al.,

2004; Weinberger and Fischer, 2006; Wang et al., 2007; Fortuna et al., 2007; Kim et al., 2010b). For a more complete review of models for discourse disentanglement and dialogue act tagging, see Kim et al. (2010b).

Joint classification has been applied in a number of different contexts, based on the intuition that it should be possible to harness interactions between different sub-tasks to the mutual benefit of both. Warnke et al. (1997) jointly performed segmentation and dialogue act classification over a German spontaneous speech corpus. In their approach, the predictions of a multi-layer perceptron classifier on dialogue act boundaries were fed into an n-gram language model, which was used for the joint segmentation and classification of dialogue acts. Sutton and McCallum (2005) performed joint parsing and semantic role labelling (SRL), using the results of a probabilistic SRL system to improve the accuracy of a probabilistic parser. Finkel and Manning (2009) built a joint, discriminative model for parsing and named entity recognition (NER), addressing the problem of inconsistent annotations across the two tasks, and demonstrating that NER benefited considerably from the interaction with parsing. Dahlmeier et al. (2009) proposed a joint probabilistic model for word sense disambiguation (WSD) of prepositions and SRL of prepositional phrases (PPs), and achieved state-of-the-art results over both tasks.

There has been a recent growth in user-level research over forums. Lui and Baldwin (2009) explored a range of user-level features, including replies-to and co-participation graph analysis, for post quality classification. Lui and Baldwin (2010) introduced a novel user classification task where each user is classified against four attributes: clarity, proficiency, positivity and effort. User communication roles in web forums have also been studied (Chan and Hayes, 2010; Chan et al., 2010).

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). These results suggest that the thread structural representation used in this research, which includes both linking structure and the dialogue act associated with each link, could potentially provide even greater leverage in these retrieval tasks.

Another related research area is post-level classification, such as general post quality classification (Weimer et al., 2007; Weimer and Gurevych, 2007; Wanas et al., 2008; Lui and Baldwin, 2009), and post descriptiveness in particular domains (e.g. medical forums: Leaman et al. (2010)). It has been demonstrated (Wanas et al., 2008; Lui and Baldwin, 2009) that thread discourse structure can significantly improve the classification accuracy for post-level tasks.

Initiation–response pairs (e.g. question–answer, assessment–agreement, and blame–denial) from online forums have the potential to enhance thread summarisation or automatically generate knowledge bases for Community Question Answering (cQA) services such as Yahoo! Answers. While initiation–response pair identification has been explored as a pairwise ranking problem (Wang and Rosé, 2010), question–answer pair identification has been approached via the two separate sub-tasks of question classification and answer detection (Cong et al., 2008; Ding et al., 2008; Cao et al., 2009). Our thread discourse structure prediction task includes joint classification of post roles (i.e. dialogue acts) and links, and could potentially be performed at the sub-post sentence level to extract initiation–response pairs.

3 Task Description and Data Set

The main task performed in this research is joint classification of inter-post links (Link) and dialogue acts (DA) within forum threads. In this, we assume that a post can only link to an earlier post (or a virtual root node), and that dialogue acts are labels on edges. It is possible for there to be multiple edges from a given post, e.g. if a post both confirms the validity of an answer and adds extra information to the original question (as happens in Post4 in Figure 1).

We experiment with two different approaches to joint classification: (1) a linear-chain CRF over combined Link/DA post labels; and (2) a dependency parser. The joint classification task is a natural fit for dependency parsing, in that the task is intrinsically one of inferring labelled dependencies

between posts, but it has a number of special properties that distinguish it from standard dependency parsing:

strict reverse-chronological directionality: the head always precedes the dependent, in terms of the chronological sequencing of posts.

non-projective dependencies: threads can contain non-projective dependencies, e.g. in a 4-post thread, posts 2 and 3 may be dependent on post 1, and post 4 dependent on post 2; around 2% of the threads in our dataset contain non-projective dependencies.

multi-headedness: it is possible for a given post to have multiple heads, including the possibility of multiple dependency links to the same post (e.g. adding extra information to a question [Question-Add] as well as retracting information from the original question [Question-Correction]); around 6% of the threads in our dataset contain multi-headed dependencies.

disconnected sub-graphs: it is possible for there to be disconnected sub-graphs, e.g. in instances where a user hijacks a thread to ask their own unrelated question, or submit an unrelated spam post; around 2% of the threads in our dataset contain disconnected sub-graphs.

The first constraint potentially simplifies dependency parsing, and non-projective dependencies are relatively well understood in the dependency parsing community (Tapanainen and Jarvinen, 1997; McDonald et al., 2005). Multi-headedness and disconnected sub-graphs pose greater challenges to dependency parsing, although there has been research done on both (McDonald and Pereira, 2006; Sagae and Tsujii, 2008; Eisner and Smith, 2005). The combination of non-projectivity, multi-headedness and disconnected sub-graphs in a single dataset, however, poses a challenge for dependency parsing.

In addition to performing evaluation in batch mode over complete threads, we consider the task of "in situ thread classification", whereby we predict the discourse structure of a thread after each post. This is intended to simulate the more realistic setting of incrementally crawling/updating thread data, but needing to predict discourse structure for partial threads. We are interested in determining the relative degradation in accuracy for in situ classification vs. batch classification.

As our dataset, we use the CNET forum dataset of Kim et al. (2010b),[1] which contains 1332 annotated posts spanning 315 threads, collected from the Operating System, Software, Hardware and Web Development sub-forums of cnet.[2] Each post is labelled with one or more links (including the possibility of null-links, where the post doesn't link to any other post), and each link is labelled with a dialogue act. The dialogue act set is made up of 5 super-categories: Question, Answer, Resolution (confirmation of the question being resolved), Reproduction (external confirmation of a proposed solution working) 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 dialogue act tagset, see Kim et al. (2010b).

Dependency links are represented by their relative position in the chronologically-sorted list of posts, e.g. 1 indicates a link back to the preceding post, and 2 indicates a link back two posts.

Unless otherwise noted, evaluation is over the combined link and dialogue act tag, including the combination of superclass and subclass for the Question and Answer dialogue acts. For example, 1+Answer-Answer indicates a dependency link back one post, which is an answer to a question. The most common label in the dataset is 1+Answer-answer (28.4%).

4 Learners and Features

4.1 Learners

To predict thread discourse structure, we use a structured classification approach — based on the findings of Kim et al. (2010b) and Kim et al. (2010a) — and a dependency parser. The structured classification approach we experiment with is a linear-

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

[2] http://forums.cnet.com/

chain conditional random field learner (CRF: Lafferty et al. (2001)), within which we explore two simple approaches to joint classification, as is explained in Section 5.1. 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.

Standardly, CRFs have been applied to tasks such as part-of-speech tagging, named entity recognition, semantic role labelling and supertagging, where the individual tokens are single words. Similarly, dependency parsing is conventionally applied to sentences, with single-word tokens. In our case, our tokens are thread posts, with much greater scope for feature engineering than single words, and technical challenges in scaling the underlying implementations to handle potentially much larger feature sets.

As our learners, we deployed CRFSGD (Bottou, 2011) to learn the CRF, and MaltParser (Nivre et al., 2007) as our dependency parser. CRFSGD uses stochastic gradient descent to efficiently solve the convex optimisation problem, and scales well to large feature sets. We used the default parameter settings for CRFSGD, with feature templates including all unigram features of the current token as well as bigram features combining the previous output token with the current token.

MaltParser implements transition-based parsing, where no formal grammar is considered, and a transition system, or state machine, is learned to map a sentence onto its dependency graph. One feature of MaltParser that makes it well suited to our task is that it is possible to define feature models of arbitrary complexity for each token. In presenting the thread data to MaltParser, we represent the null-link from the initial post of each thread, as well as any disconnected posts, as the root.

To the best of our knowledge, there is no past work on using dependency parsing to learn thread discourse structure. Based on extensive experimentation, we determined that the MaltParser configuration that obtains the best results for our task is the Nivre algorithm in arc-standard mode (Nivre, 2003; Nivre, 2004), using LIBSVM (Chang and Lin, 2011) with a linear kernel as the learner, and a feature model with exhaustive combinations of features relating to the features and predictions of the first/top three tokens from both "Input" and "Stack".[3] As such, MaltParser is actually unable to predict any non-projective structures, as experiments with algorithms supporting non-projective structures invariably led to lower results. In our choice of parsing algorithm, we are also unable to detect posts with multiple heads, but can potentially detect disconnected sub-graphs.

4.2 Features

The features used in our classifiers are as follows:

Structural Features:

Initiator a binary feature indicating whether the current post's author is the thread initiator.

Position the relative position of the current post, as a ratio over the total number of posts in the thread.

Semantic Features:

TitSim the relative location of the post which has the most similar title (based on unweighted cosine similarity) to the current post.

PostSim the relative location of the post which has the most similar content (based on unweighted cosine similarity) to the current post.

Punct the number of question marks (QuCount), exclamation marks (ExCount) and URLs (UrlCount) in the current post.

UserProf the class distribution (in the training thread) of the author of the current post.

These features are drawn largely from the work of Kim et al. (2010b), with two major differences: (1) we do not use post context features because our learners (i.e. CRFSGD and MaltParser) inherently capture Markov chains; and (2) our UserProf features are customised to the class set associated with the task at hand, e.g. the UserProf features for the standalone linking task take the form of the link labels (and not dialogue act labels) of the posts by the relevant author in the training data. Table 1 shows the feature representation of the third post in a thread

Feature	Value	Explanation
Initiator	1.0	post from the initiator
ExCount	4.0	4 exclamation marks
QuCount	0.0	0 question marks
UrlCount	0.0	0 URLs
Position	0.25	$\frac{i-1}{n} = \frac{3-1}{8}$
PostSim	2.0	most similar to post 1
TitSim	2.0	most similar to post 1
UserProf	$\vec{x}$	counts for posts of each class from the same author in the training data

Table 1: The feature presentation of the third post in a thread of length 8

of length 8. The values of each feature are scaled to the range $[0, 1]$ before being fed into the learners.

We also experimented with other features, including raw bag-of-words lexical features, dimensionality-reduced lexical features (using principal components analysis), and different post similarity measures such as longest common subsequence (LCS) match. While we were able to obtain gains in isolation, when combined with the other features, these features had no impact, and are thus not included in the results presented in this paper.

5 Classification Methodology

All our experiments were carried out based on stratified 10-fold cross-validation, stratifying at the thread level to ensure that all posts from a given thread occur in a single fold. The results are primarily evaluated using post-level micro-averaged F-score (F_μ: $\beta = 1$), and additionally with thread-level F-score/classification accuracy (i.e. the proportion of threads where all posts have been correctly classified[4]), where space allows. Statistical significance is tested using randomised estimation (Yeh, 2000) with $p < 0.05$. Initial experiments showed it is hard for learners to discover which posts have multiple links, largely due to the sparsity of multi-headed posts (which account for less than 5% of the total posts). Therefore, only the the most recent link for each multi-headed post was included in training, but evaluation still considers all links.

5.1 Joint classification

In our experiments, we test two basic approaches to joint classification for the CRF: (1) classifying the Link and DA separately, and composing the predictions to form the joint classification (Composition); and (2) combining the Link and DA labels into a single class, and applying the learner over the posts with the combined class (Combine). Note that Composition has the potential for mismatches in the number of Link and DA predictions it generates, causing complications in the class composition. Even if the same number of labels is predicted for both Link and DA, if multiple tags are predicted in both cases, we are left with the problem of determining which link label to combine with which dialogue act label. As such, we have our reservations about Composition, but as the CRF performs strict 1-of-n labelling, these are not issues in the experiments reported herein.

MaltParser natively handles the combination of Link and DA in its dependency parsing formulation.

5.2 In Situ Thread Classification

One of the biggest challenges in classifying the discourse structure of a forum thread is that threads evolve over time, as new posts are posted. In order to capture this phenomenon, and compare the accuracy of different models when applied to partial thread data (artificially cutting off a thread at post N) vs. complete threads.[5] This is done in the following way: classification over the first two posts only ($[1, 2]$), the first four posts ($[1, 4]$), the first six posts ($[1, 6]$), the first eight posts ($[1, 8]$), and all posts ($[all]$). In each case, we limit the test data only, meaning that the only variable in play is the extent of thread context used to learn the thread discourse structure for the given set of posts. We break down the results in each case into the indicated subthreads, e.g. we take the predictions for $[all]$, and break them down into the results for $[1, 2]$, $[1, 4]$, $[1, 6]$, $[1, 8]$ and $[all]$, for direct comparison with the predictions over the respective sub-thread data.

[3] `http://maltparser.org/userguide.html#parsingalg`

[4] Classification accuracy = F-score at the thread-level, as each thread is assigned a single label of correct or incorrect.

[5] In practice, completeness is defined at a given point in time, when the crawl was done, and it is highly likely that some of the "complete" threads had extra posts after the crawl.

Method	Link	DA
Kim et al. (2010b)	.863 / .676	.751 / .543
CRFSGD	.891 / .727	.795 / .609

Table 2: Post/thread-level component-wise classification F-scores for Link and DA classes

6 Experiments and Analysis

6.1 Joint classification

As our baseline for the task, we first use a simple majority class classifier in the form of the single joint class of 1+Answer-Answer for all posts, which has a post-level F-score of 0.284. A stronger baseline is to classify all first posts as 0+Question-Question and all subsequent posts as 1+Answer-answer, which achieves a post-level F-score of 0.515 (labelled as Heuristic).

As described in Section 5.1, one approach to joint classification with CRFSGD is to firstly conduct component-wise classification over Link and DA separately, and compose the predictions. The results for the separate Link and DA classification tasks are presented in Table 2, along with the best results for Link and DA classification from Kim et al. (2010b). At the component-wise tasks, our method is superior to Kim et al. (2010b), based on a different learner and slightly different feature set.

Next, we compose the component-wise classifications for the CRF into joint classifications (Composition). We contrast this with the combined class approach for CRFSGD and MaltParser (jointly presented as Joint in Table 3). With the combined class results, we additionally ablate each of the feature types from Section 4.2, and also present results for a dummy model, where no features are provided and the prediction is based simply on sequential priors (Dummy). The results are presented in Table 3, along with the Heuristic baseline result.

Several interesting things can be observed from the post-level F-score results in Table 3. First, with no features (Dummy), while CRFSGD performs slightly worse than the Heuristic baseline, MaltParser significantly surpasses the baseline. This is due to the richer sequential context model of MaltParser. Second, the single feature with the greatest impact on results is UserProf, i.e. user profile features extracted from the training data; CRFSGD in particular benefits from this feature. We return to explore this effect in Section 6.4. Third, although the Initiator feature does not have much effect on CRFSGD, it affects the performance of MaltParser significantly. Further experiments shown that the combination of Initiator and UserProf is sufficient to achieve a competitive result (i.e. 0.731). It therefore seems that MaltParser is more robust than CRFSGD, whose performance relies crucially on user-level features which must be learned from the training data (i.e. UserProf).

Method	CRFSGD	MaltParser
Heuristic	.515*/ .311*	
Dummy	.508*/ .394*	.533*/ .356*
Composition	.728*/ .553*	—
Joint +ALL	.756 / .578	.738 / .578
−Initiator	.745 / .569	.708*/ .534*
−Position	.750 / .565	.736 / .568
−PostSim	.753 / .578	.737 / .568
−TitSim	.760 / .587	.734 / .571
−Punct	.745 / .571	.735 / .578
−UserProf	.672*/ .527*	.701*/ .536*

Table 3: Post/thread-level Link-DA joint classification F-scores ("*" signifies a significantly worse result than that for the same learner with ALL features)

Looking to the thread-level F-scores, we observe some interesting divergences from the post-level F-score results. First, with no features (Dummy), CRFSGD significantly outperforms both the baseline and MaltParser. This appears to be because CRFSGD performs particularly well over short threads (e.g. of length 3 and 4), but worse over longer threads. Second, the best thread-level F-scores from CRFSGD (i.e. 0.587) and MaltParser (i.e. 0.578) are not significantly different, despite the discrepancy in post-level F-score (where CRFSGD is markedly superior in this case). With the extra features, the performance of MaltParser on short threads appears to pick up noticeably, and the difference in post-level predictions is over longer threads.

If we evaluate the two models over DA superclasses only (ignoring mismatches at the subclass level for Question and Answer), the post-level F-scores for joint classification with ALL features for CRFSGD and MaltParser are 0.803 and 0.787, respectively.

Approaches	Link	DA
Component-wise	.891 / .727*	**.795 / .609**
CRFSGD decomp	**.893 / .749**	.785 / .603
MaltParser decomp	.870*/ .730*	.766*/ .571*

Table 4: Post/thread-level Link and DA F-scores from component-wise classification, and from Link-DA classification decomposition ("*" signifies a significantly *worse* result than the **best** result in that column)

Looking at the performance of CRFSGD (in Combine mode) and MaltParser on disconnected sub-graphs, while both models did predict a small number of non-initial posts with null-links (including MaltParser predicting 5 out of 6 posts in a single thread as having null-links), none were correct, and neither model was able to correctly predict any of the 6 actual non-initial instances of null-links in the dataset.

Finally, we took the joint classification results from CRFSGD and MaltParser using ALL features, and decomposed the predictions into Link and DA. The results are presented in Table 4, along with the results for component-wise classification from Table 2. Somewhat surprisingly, the decomposed predictions are mostly slightly worse than the results for the component-wise classification, despite achieving higher F-score for the joint classification task. This is simply due to the combined method tending to get both labels correct or both labels wrong, for a given post.

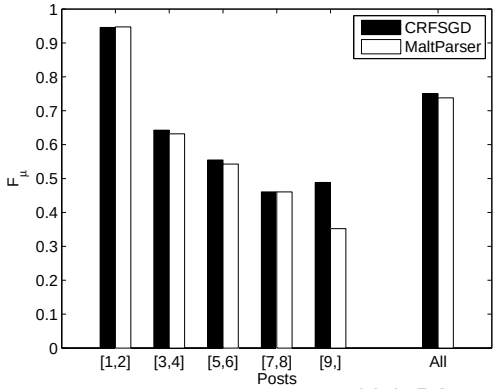

Figure 2: Breakdown of post-level Link-DA results for CRFSGD and MaltParser based on post position

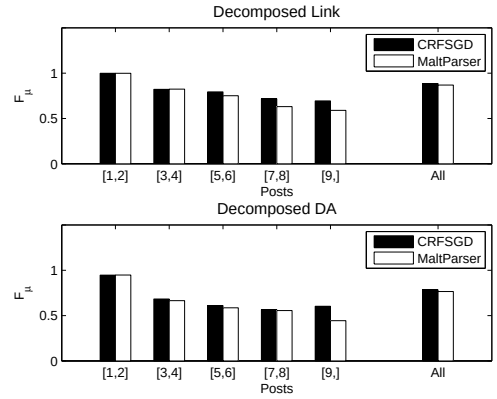

Figure 3: Breakdown of post-level Link and DA F-score based on the decomposition of CRFSGD and MaltParser classifications

6.2 Post Position-based Result Breakdown

One question in thread discourse structure classification is how accurate the predictions are at different depths in a thread (e.g. the first two posts vs. the second two posts). A breakdown of results across posts at different positions is presented in Figure 2.

The overall trend for both CRFSGD and MaltParser is that it becomes increasingly hard to classify posts as we continue through a thread, due to greater variability in discourse structure and greater sparsity in the data. However, it is interesting to note that the results for CRFSGD actually improve from posts 7 and 8 ($[7, 8]$) to posts 9 and onwards ($[9,]$). To further investigate this effect, we performed class decomposition over the joint classification predictions, and performed a similar breakdown of posts for Link and DA; the results are presented in Figure 3. It is clear that the anomaly for CRFSGD comes from the DA component, due to there being greater predictability in the dialogue for final posts in a thread (users tend to confirm a successful resolution of the problem, or report on successful external reproduction of the solution). MaltParser seems less adept at identifying that a post is at the end of a thread, and predicting the dialogue act accordingly. This observation is congruous with the findings of McDonald and Nivre (2007) that errors propagate, due to MaltParser's greedy inference strategy. The higher results for Link are to be expected, as throughout the thread, most posts tend to link locally.

Test \ B/down	$[1,2]$	$[1,4]$	$[1,6]$	$[1,8]$	$[All]$
$[1,2]$	.947/.947	—	—	—	—
$[1,4]$	.946/.947	.836/.841	—	—	
$[1,6]$	.946/.947	.840/.841	.800/.794	—	—
$[1,8]$	.946/.947	.840/.841	.800/.794	.780/.769	—
$[All]$	.946/.946	.840/.838	.800/.791	.776/.767	.756/.738

Table 5: Post-level Link-DA F-score for CRFSGD/MaltParser, based on in situ classification over sub-threads of different lengths (indicated in the rows), broken down over different post extents (indicated in the columns)

6.3 In Situ Structure Prediction

As described in Section 5.2, we simulate in situ thread discourse structure prediction by removing differing numbers of posts from the tail of the thread, and applying the trained model over the resultant sub-threads. The results for in situ classification are presented in Table 5, with the rows indicating the size of the test sub-thread, and the columns being a breakdown of results over different portions of the classified thread. The reason that we do not provide numbers for all cells in the table is that the size of the test sub-thread determines the post extents we can breakdown the results into, e.g. we cannot return results for posts 1–4 ($[1,4]$) when the size of the test thread was only two posts ($[1,2]$).

From the results, we can see that both CRFSGD and MaltParser are very robust when applied to partial threads, to the extent that we actually achieve higher results over shortened versions of the thread than over the complete thread in some instances, although the only difference that is statistically significant is over $[1,8]$ for CRFSGD, where the prediction over the partial thread is actually superior to that over the complete thread. From this, we can conclude that it is possible to apply our method to partial threads without any reduction in effectiveness relative to classification over complete threads. As such, our method is shown to be robust when applied to real-time analysis of dynamically evolving threads.

6.4 User profile feature analysis

In our experiments, we noticed that the user profile feature (UserProf) is the most effective feature for both CRFSGD and MaltParser. To gain a deeper insight into the behaviour of the feature, we binned the posts according to the number of times the author had posted in the training data, evaluated based on a user score ($uscore$) for each user:

Bin	$uscore$	Posts per user	Total users	Total posts
High	224.6	251	1	251
Medium	1~41.7	4~48	45	395
Low	0	2~4	157	377
Very Low	0	1	309	309

Table 6: Statistics for the 4 groups of users

$$uscore_i = \frac{\sum_{j=1}^{n_i} s_{p_{i,j}}}{n_i}$$

where n_i is the number of posts by user i, and $s_{p_{i,j}}$ is the number of posts by user i that occur as training instances for other posts by the same author. $uscore$ reflects the average training–test post ratio per user in cross-validation. Note that as we include all posts from a given thread in a single partition during cross-validation, it is possible for an author to have posted 4 times, but have a $uscore$ of 0 due to those posts all occurring in the same thread.

We ranked the users in the dataset in descending order of $uscore$, sub-ranking on n_i in cases of a tie in $uscore$. The users were binned into 4 groups of roughly equal post size. The detailed statistics are shown in Table 6, noting that the high-frequency bin ("High") contains posts from a single user. We present the post-level micro-averaged F-score for posts in each bin based on CRFSGD, with and without user profile features, in Figure 4.

Contrary to expectation, the UserProf features have the greatest impact for users with fewer posts. In fact, a statistically significant difference was observed only for users with no posts in the training data ($uscore = 0$), where the F-score jumped over 10% in absolute terms for both the Low and Very Low bins. Our explanation for this effect is that the

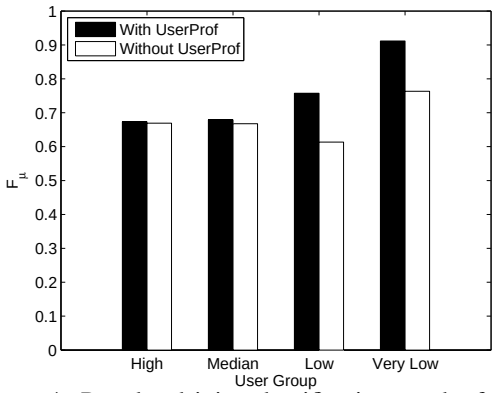

Figure 4: Post-level joint classification results for users binned by $uscore$, based on CRFSGD with and without UserProf features)

lack of user profile information is predictive of the sort of posts we can expect from a user (i.e. they tend to be newbie users, asking questions).

7 Conclusions and Future Work

In this research, we explored the joint classification of web user forum thread discourse structure, in the form of a rooted directed acyclic graph over posts, with edges labelled with dialogue acts. Three classification approaches were proposed: separately predicting Link and DA labels, and composing them into a joint class; predicting a combined Link-DA class using a structured classifier; and applying dependency parsing to the problem. We found the combined approach based on CRFSGD to perform best over the task, closely followed by dependency parsing with MaltParser.

We also examined the task of in situ classification of dialogue structure, in the form of predicting the discourse structure of partial threads, as contrasted with classifying only complete threads. We found that there was no drop in F-score over different subextents of the thread in classifying partial threads, despite the relative lack of thread context.

In future work, we plan to delve further into dependency parsing, looking specifically at the implications of multi-headedness and disconnected subgraphs on dependency parsing. We also intend to carry out meta-classification, combining the predictions of CRFSGD and MaltParser.

Our user profile features were found to be the pick of our features, but counter-intuitively, to benefit users with no posts in the training data, rather than prolific users. We wish to explore this effect further, including incorporating unsupervised user-level features into our classifiers.

Acknowledgements

The authors wish to acknowledge the development efforts of Johan Hall in configuring MaltParser to handle numeric features, and be able to parse thread structures. 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

Léon Bottou. 2011. CRFSGD software. `http://leon.bottou.org/projects/sgd`.

Xin Cao, Gao Cong, Bin Cui, Christian S. Jensen, and Ce Zhang. 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.

Vitor R. Carvalho and William W. Cohen. 2005. On the collective classification of email "speech acts". In *Proceedings of 28th International ACM-SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2005)*, pages 345–352.

Jeffrey Chan and Conor Hayes. 2010. Decomposing discussion forums using user roles. In *Proceedings of the WebSci10: Extending the Frontiers of Society On-Line (WebSci10)*, pages 1–8, Raleigh, USA.

Jeffrey Chan, Conor Hayes, and Elizabeth M. Daly. 2010. Decomposing discussion forums using user roles. In *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media (ICWSM 2010)*, pages 215–8, Washington, USA.

Chih-Chung Chang and Chih-Jen Lin. 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`.

William W. Cohen, Vitor R. Carvalho, and Tom M. Mitchell. 2004. Learning to classify email into "speech acts". In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)*, pages 309–316, Barcelona, Spain.

Gao Cong, Long Wang, Chin-Yew Lin, Young-In Song, and Yueheng Sun. 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.

Daniel Dahlmeier, Hwee Tou Ng, and Tanja Schultz. 2009. Joint learning of preposition senses and semantic roles of prepositional phrases. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP 2009)*, pages 450–458, Singapore. Association for Computational Linguistics.

Shilin Ding, Gao Cong, Chin-Yew Lin, and Xiaoyan Zhu. 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.

Jason Eisner and Noah A. Smith. 2005. Parsing with soft and hard constraints on dependency length. In *Proceedings of the Ninth International Workshop on Parsing Technology*, pages 30–41, Vancouver, Canada.

Jonathan L. Elsas and Jaime G. Carbonell. 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.

Micha Elsner and Eugene Charniak. 2008. You talking to me? a corpus and algorithm for conversation disentanglement. In *Proceedings of the 46th Annual Meeting of the ACL: HLT (ACL 2008)*, pages 834–842, Columbus, USA.

Jenny Rose Finkel and Christopher D. Manning. 2009. Joint parsing and named entity recognition. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL HLT 2009)*, pages 326–334, Boulder, Colorado. Association for Computational Linguistics.

Blaz Fortuna, Eduarda Mendes Rodrigues, and Natasa Milic-Frayling. 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.

Barbara J. Grosz and Candace L. Sidner. 1986. Attention, intention and the structure of discourse. *Computational Linguistics*, 12(3):175–204.

Edward Ivanovic. 2008. Automatic instant messaging dialogue using statistical models and dialogue acts. Master's thesis, University of Melbourne.

Su Nam Kim, Lawrence Cavedon, and Timothy Baldwin. 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.

Su Nam Kim, Li Wang, and Timothy Baldwin. 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.

Sandra Kübler, Ryan McDonald, and Joakim Nivre. 2009. Dependency parsing. *Synthesis Lectures on Human Language Technologies*, 2(1):1–127.

John Lafferty, Andrew McCallum, and Fernando Pereira. 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.

Andrew Lampert, Robert Dale, and Cécile Paris. 2008. The nature of requests and commitments in email messages. In *Proceedings of the AAAI 2008 Workshop on Enhanced Messaging*, pages 42–47, Chicago, USA.

Robert Leaman, Laura Wojtulewicz, Ryan Sullivan, Annie Skariah, Jian Yang, and Graciela Gonzalez. 2010. Towards internet-age pharmacovigilance: Extracting adverse drug reactions from user posts in health-related social networks. In *Proceedings of the 2010 Workshop on Biomedical Natural Language Processing (ACL 2010)*, pages 117–125, Uppsala, Sweden.

Oliver Lemon, Alex Gruenstein, and Stanley Peters. 2002. Collaborative activities and multi-tasking in dialogue systems. *Traitement Automatique des Langues (TAL), Special Issue on Dialogue*, 43(2):131–154.

Chen Lin, Jiang-Ming Yang, Rui Cai, Xin-Jing Wang, Wei Wang, and Lei Zhang. 2009. Modeling semantics and structure of discussion threads. In *Proceedings of the 18th International Conference on the World Wide Web (WWW 2009)*, pages 1103–1104, Madrid, Spain.

Marco Lui and Timothy Baldwin. 2009. You are what you post: User-level features in threaded discourse. In *Proceedings of the 14th Australasian Document Computing Symposium (ADCS 2009)*, Sydney, Australia.

Marco Lui and Timothy Baldwin. 2010. Classifying user forum participants: Separating the gurus from the hacks, and other tales of the internet. In *Proceedings of the 2010 Australasian Language Technology Workshop (ALTW 2010)*, pages 49–57, Melbourne, Australia.

Ryan McDonald and Joakim Nivre. 2007. Characterizing the errors of data-driven dependency parsing models. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2007)*, pages 122–131, Prague, Czech Republic.

Ryan McDonald and Fernando Pereira. 2006. Online learning of approximate dependency parsing algorithms. In *Proceedings of the 11th Conference of*

the European Chapter of the Association for Computational Linguistics (EACL 2006), pages 81–88, Trento, Italy.

Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajic. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 523–530, Vancouver, Canada.

Gabriel Murray, Steve Renals, Jean Carletta, and Johanna Moore. 2006. Incorporating speaker and discourse features into speech summarization. In *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 367–374.

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

Joakim Nivre, Johan Hall, Jens Nilsson, Atanas Chanev, Gülsen Eryigit, Sandra Kübler, Svetoslav Marinov, and Erwin Marsi. 2007. MaltParser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(02):95–135.

Joakim Nivre. 2003. An efficient algorithm for projective dependency parsing. In *Proceedings of the 8th International Workshop on Parsing Technologies (IWPT 03)*, pages 149–160, Nancy, France.

Joakim Nivre. 2004. Incrementality in deterministic dependency parsing. In *Proceedings of the ACL Workshop Incremental Parsing: Bringing Engineering and Cognition Together (ACL-2004)*, pages 50–57, Barcelona, Spain.

Carolyn Penstein Rosé, Barbara Di Eugenio, Lori S. Levin, and Carol Van Ess-Dykema. 1995. Discourse processing of dialogues with multiple threads. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pages 31–38, Cambridge, USA.

Kenji Sagae and Jun'ichi Tsujii. 2008. Shift-reduce dependency DAG parsing. In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING 2008)*, pages 753–760, Manchester, UK.

Kenji Sagae. 2009. Analysis of discourse structure with syntactic dependencies and data-driven shift-reduce parsing. In *Proceedings of the 11th International Conference on Parsing Technologies (IWPT-09)*, pages 81–84, Paris, France.

Anne Schuth, Maarten Marx, and Maarten de Rijke. 2007. Extracting the discussion structure in comments on news-articles. In *Proceedings of the 9th Annual ACM International Workshop on Web Information and Data Management*, pages 97–104, Lisboa, Portugal.

Jangwon Seo, W. Bruce Croft, and David A. Smith. 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.

Elinzabeth Shriberg, Raj Dhillon, Sonali Bhagat, Jeremy Ang, and Hannah Carvey. 2004. The ICSI meeting recorder dialog act (MRDA) corpus. In *Proceedings of the 5th SIGdial Workshop on Discourse and Dialogue*, pages 97–100, Cambridge, USA.

Parikshit Sondhi, Manish Gupta, ChengXiang Zhai, and Julia Hockenmaier. 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.

Radu Soricut and Daniel Marcu. 2003. Sentence level discourse parsing using syntactic and lexical information. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2003)*, pages 149–156, Edmonton, Canada.

Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Pail Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26(3):339–373.

Charles Sutton and Andrew McCallum. 2005. Joint parsing and semantic role labeling. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, pages 225–228, Ann Arbor, Michigan. Association for Computational Linguistics.

Pasi Tapanainen and Timo Jarvinen. 1997. A non-projective dependency parser. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, pages 64–71, Washington, USA.

Nayer Wanas, Motaz El-Saban, Heba Ashour, and Waleed Ammar. 2008. Automatic scoring of online discussion posts. In *Proceeding of the 2nd ACM workshop on Information credibility on the web (WICOW '08)*, pages 19–26, Napa Valley, USA.

Yi-Chia Wang and Carolyn P. Rosé. 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.

Yi-Chia Wang, Mahesh Joshi, and Carolyn Rosé. 2007. A feature based approach to leveraging context for classifying newsgroup style discussion segments. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions (ACL 2007)*, pages 73–76, Prague, Czech Republic.

Yi-Chia Wang, Mahesh Joshi, William W. Cohen, and Carolyn Rosé. 2008. Recovering implicit thread structure in newsgroup style conversations. In *Proceedings of the Second International Conference on Weblogs and Social Media (ICWSM 2008)*, pages 152–160, Seattle, USA.

V. Warnke, R. Kompe, H. Niemann, and E. Nöth. 1997. Integrated dialog act segmentation and classification using prosodic features and language models. In *Proc. Eurospeech*, volume 1, pages 207–210.

Markus Weimer and Iryna Gurevych. 2007. Predicting the perceived quality of web forum posts. In *Proceedings of the 2007 International Conference on Recent Advances in Natural Language Processing (RANLP 2007)*, pages 643–648, Borovets, Bulgaria.

Markus Weimer, Iryna Gurevych, and Max Mühlhäuser. 2007. Automatically assessing the post quality in online discussions on software. In *Proceedings of the 45th Annual Meeting of the ACL: Interactive Poster and Demonstration Sessions*, pages 125–128, Prague, Czech Republic.

Armin Weinberger and Frank Fischer. 2006. A framework to analyze argumentative knowledge construction in computer-supported collaborative learning. *Computers & Education*, 46:71–95, January.

Florian Wolf and Edward Gibson. 2005. Representing discourse coherence: A corpus-based study. *Computational Linguistics*, 31(2):249–287.

Wensi Xi, Jesper Lind, and Eric Brill. 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.

Alexander Yeh. 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.