You Are What You Post: User-level Features in Threaded Discourse

Marco Lui and Timothy Baldwin
University of Melbourne
VIC 3010 Australia
saffsd@gmail.com, tb@ldwin.net

Abstract *We develop methods for describing users based on their posts to an online discussion forum. These methods build on existing techniques to describe other aspects of online discussions communities, but the application of these techniques to describing users is novel. We demonstrate the utility of our proposed methods by showing that they are superior to existing methods over a post-level classification task over a published real-world dataset.*

Keywords Document Management, Information Retrieval, Web Documents

1 Introduction

People like to talk. In particular, people like to talk to other people that share their interests, resulting in everything from hobby groups to clubs to professional associations. The internet gives people the ability to talk to each other on an unprecedented scale, and this has fostered the growth of publicly-accessible communities around a gamut of topics, from technology (Slashdot[1]) to knitting (Ravelry[2]), to social interaction for its own sake (Facebook[3]).

The most natural form of communication is through dialogue, and in the internet age this manifests itself via modalities such as forums and mailing lists. What these systems have in common is that they are a textual representation of a *threaded discourse*. The Internet is full of publicly-accessible communities which engage in innumerable discourses, generating massive quantities of data in the process. This data is rich in information, and with the help of computers we are able to archive it, index it, query it and retrieve it. In theory, this would allow people to take a question to an online community, search its archives for the same or similar questions, follow up on the contents of prior discussion and find an answer. However, anyone with any experience in searching for an answer to a technical question online would agree that the situation is seldom that simple.

[1] http://www.slashdot.org
[2] http://www.ravelry.com
[3] http://www.facebook.com

Proceedings of the 14th Australasian Document Computing Symposium, Sydney, Australia, 4 December 2009.

One problem with current approaches to accessing threaded discourse data is that they do not take into account the structure of the discourse itself. The bag-of-words (BOW) model standardly used in text classification and information retrieval (IR) discards all contextual information. However, even in IR it has long been known that much more information than simple term occurrence is available. In the modern era of web search, for example, extensive use is made of link structure, anchor text, document zones, and a plethora of other document (and query, click stream and user) features [15].

The natural question to ask at this point is, "What additional structure can we extract from threaded discourse?" Previous work has been done in extracting useful information from various implicit relationships between chunks of data in threaded discourse; we describe this in more detail in Section 2. However, one dimension that has not yet been explored is how we can use information about the identity of the participants to extract useful information from the structure of the discourse. In this paper we will examine how we can extract such *user-level* features, and how we can use them to improve performance over established tasks.

We use the term *threaded discourse* to describe online data that represents a record of messages exchanged between a group of participants. The two most common examples of this are forums and mailing lists. In this paper, the data that we examine in Section 5 originates from a site which bridges both. Indeed, the techniques we describe should generalize to any data which can be mapped into a similar structure.

There are several dimensions to the structure of threaded discourse that can be useful. For this paper, we will focus on the relationships between participants, which we refer to as the *user-level* structure. However, most instances of threaded discourse do not encode relationships between users explicitly. Therefore, we must infer the user-level relationships from relationships in other dimensions of the data. In particular, we focus on the following levels of threaded discourse structure:

Post-level: The individual unit contributions submitted by participants

Thread-level: Groupings of posts into a discussion on a particular topic

Our contribution in this paper is to develop methods for describing users based on their posts to an online discussion forum. We demonstrate the utility of our proposed methods by showing that they are superior to existing methods over a post-level classification task over a dataset from Nabble.[4]

The research presented in this paper forms a component of a larger research agenda on the utility of user-level characteristics in a variety of user forum tasks [**?**].

2 Related Work

This section provides a first-gloss overview of related work on thread- and post-level text classification, and feature-based approaches to capturing user characteristics. We return to present the aspects of this work that are most relevant to our research in greater detail, as detailed below.

Wanas et al. [16] detail a set of post-level features extracted based on a more structured approach. They evaluate their feature set over a classification task involving post and rating data derived from Slashdot. Their task involves classifying discrete posts into one of three quality levels (High, Medium or Low) where the gold-standard is provided by annotations from the community itself. We implement part of their feature set for experiments conducted in this paper; more detail is provided in Section 4.

Agrawal et al. [1] describe a technique for partitioning the users in an online community based on their opinion on a given topic. They find that basic text classification techniques are unable to do better than the majority-class baseline for this particular task. They then describe a technique based on modeling the community as a *reply-to* network, with users as individual nodes, and edges indicating that a user has replied to a post by another user. They find that using this representation, they are able to do much better than the baseline. Fortuna et al. [5] build on the work done by [1], defining additional classes of networks that represent some of the relationships present in an online community. We describe these networks in detail in Section 4, and adapt them to generate user-level features.

Weiner et al. [17, 18] propose a set of heuristic post-level features to predict the perceived quality of posts using a supervised machine learning approach. The data they evaluate over is extracted from Nabble, and they use the ratings provided by users as the gold-standard for a correct classification. They conclude that post-level classification using their feature set provides a substantial improvement over the majority-class baseline. We describe the dataset in greater detail Section 5.1, and use it as the basis of our evaluation.

In work on thread classification, Baldwin et al. [2] attempted to classify forum threads scraped from Linux-related newsgroups according to three qualities:

Task Orientation: Is the thread about a specific problem?

Completeness: Is the problem described in adequate detail?

Solvedness: Has a solution been provided?

They manually annotated a set of 250 threads for these qualities, and extracted a set of features to describe each thread based on the aggregation of features from posts in different sections of the thread. We apply a similar idea, but instead of aggregating over sections of the thread, we aggregate posts from a given user. The results from [2] were inconclusive, but we have found that their feature set can be effective when aggregated by user. Full details of the feature set are presented in Section 5.3.

3 Applications

While our experiments in this paper focus exclusively on a post-level classification task, this research has potential impact in a much wider range of settings, as outlined in this section.

3.1 Information Access

A key application of this paper is to support improved information access over internet forums, building on the work of Baldwin et al. [2]. The underlying intuition here is that not all contributions on a forum are equal in their usefulness, and that we often find that certain users are consistently outstanding in their contribution. Note that this is not the same as the user being an expert — other qualities come into play, such as how clear their explanations are, as well as how much effort they put into responding. Indeed, a relatively inexperienced user may post a detailed description of how he or she tackled a particular problem, which could be extremely valuable to a similar inexperienced user tacking a similar problem.

3.2 User Profiling

In some situations, we may wish to identify users with particular characteristics. For example, Kim et al. [9] use Speech Act Analysis to classify student contributions according to Speech Act categories, thereby identifying roles that participants play, using this information to identify when participants require assistance. This approach can be enhanced with user-level features.

3.3 User Karma

Karma is formalization of the notion of how influential a user is in an online community. It is the subject of much discussion in web communities as it is critical to the self-organizing structure of some communities, such as Reddit.[5] It is even more influential in other

[4] `http://www.nabble.com`

[5] `http://www.reddit.com`

Feature name	*Description*	*Type*
distribution	Mention of the name of a Linux distribution	Boolean
beginner	Mention of terms suggesting the posted is inexperienced	Boolean
emoticons	Presence of "smiley faces"	Boolean
version numbers	Presence of version numbers	Boolean
URLs	Presence of hyperlinks	Boolean
words	Number of words in post	Integer
sentence	Number of sentences in post	Integer
question sentence	Number of questions in post (sentences ending in '?')	Integer
exclaim sentence	Number of exclamations in post (sentences ending in '!')	Integer
period sentence	Number of sentences ending in a period	Integer
other sentence	Number of sentences not falling into the above three categories	Integer

Table 1: The ILIAD feature set

Feature name	*Description*	*Type*
onThreadTopic	Post's relevance to the topic of a thread	Float
overlapPrevious	Post's largest overlap to a previous post	Float
overlapDistance	How far away the previous overlapping post is	Integer
timeliness	Ratio of time interval from previous post to average inter-post interval in thread	Float
lengthiness	Ratio of length of post to average length of post in thread	Float
formatEmoticons	How often emoticons are used with respect to number of sentences	Float
formatCapitals	How often capitals are used with respect to number of sentences	Float
weblinks	How often weblinks are used with respect to number of sentences	Float

Table 2: The WANAS feature set

communities, where it is used to give incremental moderation powers to users (e.g. Stack Overflow[6]). There is no body of formal research associated with it, and sometimes the exact mechanism is a closely-guarded secret. User-level features are relevant to this because they can be used to more fully describe a user, which in turn can be used to compute a karma score that takes into account more aspects of the user's participation.

3.4 Automatic Grading

Lui et al. [12] use a text classification approach to perform content analysis. This task involves automatically grading participation by students in an online learning community. They make use of a fairly simplistic model of the content. It may be possible to improve their approach by extracting more detailed structural information from participants' contributions.

4 User-Level Features

In Section 2, we outlined existing methods for extracting features to describe posts and threads. In this section, we present methods for extracting features for describing *users*.

4.1 Aggregate

The first type of user-level feature we consider are features derived from aggregation over features describing individual posts. We implement two post-level feature sets. The first, which is henceforth referred to as ILIAD, is derived from [2] and is described in Table 1. The second, which is henceforth referred to as WANAS, is derived from [16], and is described in Table 2.

From each of ILIAD and WANAS we derive a user-level feature set by finding the mean of each feature value over all of the user's posts. These feature sets are referred to as $\text{ILIAD}_{\text{AGG}}$ and $\text{WANAS}_{\text{AGG}}$, respectively.

4.2 Network-Based

Fortuna et al. [5] present a method of describing forum data using Social Network Analysis. The network is a graph representation of relationships within the forum, reminiscent of algorithms such as PageRank [4]. In the case of PageRank, each node represents a webpage and each edge represents a hyperlink. In [5], the authors define 3 *author networks*, where each node represents an author, and 2 *thread networks*, in which each node represents a thread. The meaning of an edge varies for each network, and each edge may be directed or undirected according to the network.

The authors then use each of these networks to extract features on a per-post basis. We briefly summarize the method here; more detail is provided in [5].

For Author Networks, each post is assigned a feature vector v of length N, where N is the total number of nodes, or equivalently, the total number of authors in the network. v has at least one feature set to 1, which corresponds to the author of the post. Authors directly

[6] `http://www.stackoverflow.com`

connected to the post author in the network receive a feature value of 1, and authors that are second-level neighbours of the post author are set to a feature value of 0.5. All other values in v are set to 0. Since each post has a unique author, this network can be used to describe authors without modification.

For Thread Networks, the method for computing a feature vector is similar to that for Author Networks. The key difference is that in this instance, the vector v is of length T, where T is the total number of threads in the forum. Therefore, each value v_T in the vector describes a relationship to a particular thread. In [5], the authors are interested in the relationship between posts, so they assign to each post the feature vector of the thread it belongs to. However, in our case we do not wish to describe a post directly; instead, we are interested in describing the author. To do this, we consider every thread that the author has posted in. For each of these threads, we set the feature corresponding to the thread to 1. We then set all the immediate neighbours of the threads to 1 as well, and the second-level neighbours thereafter to 0.5.

In our work, we consider two Author Networks and one Thread Network:

POSTAFTER *(Author Network)*

POSTAFTER is modeled on the *reply-to* network described in [5]. Our data does not contain exact information about the reply structure in a thread, so we approximate this information by the temporal relationship between posts. Effectively, we have made the assumption that within a thread, each post replies to the post immediately preceding it in terms of the time-of-posting. We expect that this will generally be the case, but in the context of the original work by [1] on partitioning users by opinion, it is possible that, given three posts A, B and C, B and C both reply in objection to A, therefore defining a different network from ours. Nonetheless, our results will show that our approximation is admissible in that it can be used to augment a BOW feature set to exceed a benchmark result; we will present evidence of this in Section 5.3.

POSTAFTER is parameterized with two values: $dist$ and $count$. Being an *Author Network*, the nodes represent authors. Two authors $A1$ and $A2$ have a directed edge from $A1$ to $A2$ if and only if $A1$ submits a post to a thread that is within $dist$ posts of a post in the same thread by $A2$ on at least $count$ occasions. For our experiments, we used $dist = 1$ and $count = 3$.

THREADPARTICIPATION *Author Network*

THREADPARTICIPATION is implemented as described in [5]. In this network, nodes are again authors, and each undirected edge indicates that two authors have posted in the same thread on at least k occasions. In the original work, the authors set $k = 5$, but in out case, we use $k = 2$ as the network is too sparse for higher settings of k.

COMMONAUTHORS *Thread Network*

COMMONAUTHORS is implemented as described in [5]. In this network, nodes are threads, and each undirected edge indicates that two threads have at least m authors in common. We followed the original research in setting $m = 3$.

5 Evaluation

We evaluate the effectiveness of the features described in Section 4 by utilizing them for a classification task. In this paper, we focus exclusively on a post-level classification task, which allows us to assess the usefulness of user-level features in describing post-level data.

5.1 Dataset

The data set we are using is based on that from Weimer and Gurevych [17]. The data consists of 16562 posts across 2956 different threads. Separately, there are 4508 annotations spanning 4291 distinct posts, rating the quality of the post. Each annotation consists of an ordinal rating from 1 to 5 stars, with more stars indicating better quality. We filtered the annotated posts by removing all posts with an empty body. We also removed all posts that had an average rating of exactly 3.0. This eliminated posts that were rated 3 once, as well as posts that received contradictory ratings, such as a post rated 1 by one user and 5 by another, leaving 4094 rated posts. We divided posts into two groups, corresponding to posts with an average rating > 3.0, which we consider GOOD, and posts with an average rating ≤ 3.0, which we consider BAD. In the 4094 rated posts, there were 2060 GOOD posts and 2034 BAD posts. Our approach to filtering the data is generally consistent with that in [17]. Differences in our use of the dataset are discussed in Section 6.

5.2 Methodology

For each post, we extracted the feature sets described in Section 4, as summarized in Table 3. For user-level feature sets, we use the features corresponding to the post's author to describe the post. We evaluate various combinations of these feature sets by carrying out 10-fold cross-validation [10], as follows:

1. Divide the data randomly into 10 *partitions*
2. For each partition, train a classifier on the other 9 partitions
3. Use the trained classifier to predict the categories of the instances in the selected partition
4. Pool together the predictions from the 10 iterations and evaluate

The partitioning is performed once and re-used for each pairing of learner and feature set. We repeat this procedure using a number of different learners. The learners used, along with their parameter settings, are as follows:

Label	*Type*
BOW	Post
ILIAD	Post
WANAS	Post
$\text{ILIAD}_{\text{AGG}}$	User
$\text{WANAS}_{\text{AGG}}$	User
POSTAFTER	Author Network
THREADPARTICIPATION	Author Network
COMMONAUTHORS	Thread Network

Table 3: Feature sets used in classification

`SVM`: Support vector machines [8] as implemented in `bsvm` [6], using the package default values which correspond to an RBF kernel.

`SkewAM`: Nearest-prototype skew divergence, as implemented in `hydrat` [13]. This is a Rocchio-style approach [7], where a centroid is computed for each class by finding the arithmetic mean of all the instances of the class. Classification is then carried out by assigning the class of the single nearest neighbour. The distance metric we use is skew divergence [11], with a mixing parameter $\alpha = 0.99$.

`Maxent`: Maximum entropy modeling [3] as implemented in the `Maximum Entropy Toolkit` [19]. We use L-BFGS for parameter estimation [14], with 10 iterations of the training algorithm.

For each cross-validated result, we report the overall classification accuracy (Acc), which is the proportion of correct predictions made by the classifier; a larger number is, naturally, better. When comparing a result to a benchmark value, we also provide the p-value for a two-tailed paired t-test. We can conduct a paired t-test because for each result, the partitions used have been kept constant and thus the performance over them is directly comparable. The null hypothesis is always that the difference in the mean accuracy over all 10 partitions is identical for both results being compared. Therefore, a low p-value indicates that it is highly improbable that the two combinations of feature sets being considered have led to the same results. To facilitate discussion of statistical significance, we will consider a p-value < 0.05 to be statistically significant. This corresponds to the 5% significance level that is commonly reported. In tables, p-values that are statistically significant at the 5% significance level are shown in **bold**.

Our experiments were performed using `hydrat` [13], an open-source framework for comparing classification systems. `hydrat` provides facilities for managing and combining feature sets, setting up cross-validation tasks and automatically computing corresponding results.

5.3 Results

The baseline for this task is a majority-class (`ZeroR`) result of 0.489. Although this is a binary task, the majority-class result is less than 0.5 because the majority class varies across partitions. In 8 of the 10 partitions, it was the overall majority class (GOOD), whereas in 2 of the 10 partitions, it was the majority class in the training data but overall minority class (BAD).

Learner	*Accuracy*
`SVM`	0.780
`SkewAM`	0.812
`Maxent`	0.820
`ZeroR`	0.489

Table 4: Accuracy for each learner when utilizing only the BOW feature set

Feature Set	*Acc*	p
BOW	0.780	—
ILIAD	0.723	2.1×10^{-6}
WANAS	0.751	7.3×10^{-4}
$\text{ILIAD}_{\text{AGG}}$	**0.831**	2.4×10^{-6}
$\text{WANAS}_{\text{AGG}}$	**0.829**	2.7×10^{-4}
POSTAFTER	0.636	5.1×10^{-13}
THREADPARTICIPATION	0.670	1.1×10^{-10}
COMMONAUTHORS	0.671	4.2×10^{-11}

Table 5: Accuracy for each feature set over `SVM` (results higher than the baseline are highlighted in **bold**; p is the probability that the result differs from the benchmark only due to chance)

We establish benchmark results for this task using only the BOW feature set. The overall accuracy for each learner is summarized in Table 4. Immediately, it is apparent that the benchmark result is significantly better than the baseline. The best result over only the BOW feature set is attained by `Maxent`, with an accuracy of 0.820.

Next, we consider each learner over each individual feature set. For `Maxent` and `SkewAM`, this always leads to results that are below the BOW benchmark. For `SVM`, however, the aggregate features $\text{ILIAD}_{\text{AGG}}$ and $\text{WANAS}_{\text{AGG}}$ do better than the BOW benchmark, attaining an accuracy of 0.831 and 0.829, respectively. These are different from the BOW result with $p = 2.4\times10^{-6}$ and $p = 2.7\times10^{-4}$ respectively. Both results are statistically significant. We report results for each feature set in Table 5.

We then investigate the use of the various feature sets to augment BOW, as presented in Table 6. A fairly consistent picture emerges from this: the ILIAD and $\text{ILIAD}_{\text{AGG}}$ feature sets cause performance to drop when combined with the BOW feature set, whereas all other feature sets cause performance to rise with respect to BOW.

We also experimented with *feature ablation*, by examining the result of removing one feature set at a time from the full set of features. The results for this are reported in Table 7. Surprisingly, removing a particular

Learner		*Feature Sets Present*	*Acc*	*p*
SVM	BoW		0.780	—
	BoW	ILIAD	0.746	**0.001**
	BoW	WANAS	**0.790**	0.202
	BoW	$\text{ILIAD}_{\text{AGG}}$	0.768	0.136
	BoW	$\text{WANAS}_{\text{AGG}}$	**0.797**	**0.041**
	BoW	POSTAFTER	**0.780**	0.978
	BoW	THREADPARTICIPATION	**0.790**	0.243
	BoW	COMMONAUTHORS	**0.786**	0.492
SkewAM	BoW		0.812	—
	BoW	ILIAD	0.799	**0.041**
	BoW	WANAS	**0.827**	**0.005**
	BoW	$\text{ILIAD}_{\text{AGG}}$	0.805	0.236
	BoW	$\text{WANAS}_{\text{AGG}}$	**0.830**	**0.001**
	BoW	POSTAFTER	**0.825**	**0.019**
	BoW	THREADPARTICIPATION	**0.827**	**0.008**
	BoW	COMMONAUTHORS	**0.829**	**0.005**
Maxent	BoW		0.820	—
	BoW	ILIAD	0.624	**0.000**
	BoW	WANAS	**0.843**	**0.025**
	BoW	$\text{ILIAD}_{\text{AGG}}$	0.564	**0.000**
	BoW	$\text{WANAS}_{\text{AGG}}$	**0.849**	**0.002**
	BoW	POSTAFTER	**0.834**	0.127
	BoW	THREADPARTICIPATION	**0.836**	0.088
	BoW	COMMONAUTHORS	**0.840**	**0.043**

Table 6: Accuracy for each learner when combining each feature set with BoW (results better than the BoW benchmark for each learner are highlighted in **bold**; p is the probability that a result differs from the benchmark only due to chance, and p-values significant at the 5% level are highlighted in **bold**)

feature set can result in a statistically significant performance increase for both SVM and SkewAM. For SVM, the feature set in question is BoW, whereas for SkewAM, removing THREADPARTICIPATION or COMMONAUTHORS leads to a statistically significant increase in results. Maxent is the only learner where there is no significant increase resulting from removing a single feature set.

Finally, we proceed to test other combinations of feature sets. We exhaustively tested all possible combinations of two and three feature sets, as well as all feature sets, all feature sets minus one, and all feature sets minus ILIAD and $\text{ILIAD}_{\text{AGG}}$. The best combination that we found was using BoW, WANAS and COMMONAUTHORS, with Maxent as the learner. This produced an accuracy of 0.854. However, the top 10 combinations of features and learners all produced very similar results, so we cannot conclude that this is the undisputed best combination overall. We also found that the best combination of feature sets for SVM was different from that for Maxent, but was still extremely close to the best result. The top 10 combinations that we found over the classifiers considered are reported in Table 8.

6 Discussion

As noted in Section 5.1, our dataset is based on data originally used in [17]. Our task is most similar to the ALL task of [17], in that we do not divide the data on the basis of the Nabble sub-forum it originates from. We have also filtered the data slightly differently. The

Learner	*Feature Set*	*Acc*	*p*
SVM	ALL	0.775	—
	−BoW	**0.796**	**0.005**
	−ILIAD	0.775	1.000
	−WANAS	0.775	0.949
	$-\text{ILIAD}_{\text{AGG}}$	0.770	0.508
	$-\text{WANAS}_{\text{AGG}}$	**0.776**	0.897
	−POSTAFTER	0.775	0.949
	−THREADPARTICIPATION	**0.778**	0.731
	−COMMONAUTHORS	**0.777**	0.834
SkewAM	ALL	0.776	—
	−BoW	0.689	**0.000**
	−ILIAD	**0.778**	0.750
	−WANAS	0.769	0.248
	$-\text{ILIAD}_{\text{AGG}}$	**0.788**	0.099
	$-\text{WANAS}_{\text{AGG}}$	0.764	**0.024**
	−POSTAFTER	**0.785**	0.140
	−THREADPARTICIPATION	**0.811**	**0.000**
	−COMMONAUTHORS	**0.812**	**0.000**
Maxent	ALL	0.741	—
	−BoW	0.687	**0.003**
	−ILIAD	0.648	**0.000**
	−WANAS	0.730	0.503
	$-\text{ILIAD}_{\text{AGG}}$	0.697	0.082
	$-\text{WANAS}_{\text{AGG}}$	0.714	0.129
	−POSTAFTER	0.741	0.975
	−THREADPARTICIPATION	0.737	0.768
	−COMMONAUTHORS	0.738	0.825

Table 7: Accuracy for feature ablation over the full feature set for each learner (results better than the BoW benchmark for each learner are highlighted in **bold**; p is the probability that a result differs from the benchmark only due to chance, and p-values significant at the 5% level are highlighted in **bold**)

original authors made use of 3418 posts, whereas we use 4094 posts. The bulk of the difference is due to the original authors eliminating posts which they determined to be non-English. We did not do this because some of our methods do not make use of any language-specific information, so we were still able utilize the non-English data. According to [17], there are 668 non-English posts.

The remaining difference results from the original authors opting to eliminate any posts with 'contradictory ratings', in that the post received ratings both > 3 and ≤ 3, whereas we only eliminated posts where the average rating was $= 3.0$. In practice, the difference is negligible as it only accounts for 8 out of 4291 posts.

The original authors report a maximum accuracy of 0.775 over their ALL task. Their values are not directly comparable to ours because the two tasks are not identical, as we have described above. However, they are very similar, so our best accuracy of 0.854 suggests that our technique would yield an improvement if applied directly to the original task.

We found that, even in isolation, user-level features can outperform a benchmark based on the conventional IR bag-of-words approach, to a high level of statistical significance. This is important because it justifies the use of user-level features for post-level classification tasks. Furthermore, we showed that most of the user level feature sets can be added to the basic bag-of-words model to improve its performance, and that

Learner	*Feature Sets*								*Acc*	*p*
	BoW	ILIAD	WANAS	$\text{ILIAD}_{\text{AGG}}$	$\text{WANAS}_{\text{AGG}}$	POSTAFTER	THREADPARTICIPATION	COMMONAUTHORS		
Maxent	✓		✓					✓	0.854	**0.002**
Maxent	✓		✓		✓	✓	✓	✓	0.850	**0.002**
Maxent	✓				✓	✓			0.850	**0.002**
Maxent	✓				✓				0.849	**0.002**
SVM			✓	✓	✓				0.848	**0.006**
SVM				✓	✓				0.847	**0.004**
Maxent	✓				✓		✓		0.847	**0.006**
Maxent	✓				✓			✓	0.845	**0.012**
Maxent	✓		✓						0.843	**0.025**
Maxent	✓					✓		✓	0.842	**0.027**

Table 8: Top 10 Results over different combinations of learner and feature sets (p is the probability that the result differs from the Maxent BoW benchmark only due to chance; p-values significant at the 5% level are highlighted in **bold**)

this behaviour is consistent across a range of different learners.

Table 8 suggests that the ILIAD feature set is generally ineffective, which may explain the poor results reported in [2]. However, the SVM learner is able to make effective use of the user-level aggregates of ILIAD, $\text{ILIAD}_{\text{AGG}}$, whereas the Maxent learner is not. This is reflected in both the single-featureset experiment reported in Table 5, as well as the overall results in Table 8. The reason for this is not immediately obvious, and further investigation may yield insight into how to reconcile the two.

Another obvious difference between the results from the Maxent and SVM learners is that Maxent performs best in the presence of the BoW features, whereas SVM performs better without the BoW features. This trend is clearly visible in Table 7, where for SVM, removing the BoW features leads to a statistically significant increase in the results, whereas for Maxent and SkewAM, it causes a significant drop. This trend is also visible in Table 8, where we see that the top results for Maxent include BoW, whereas the top results for SVM exclude it. Again the reason for this is not immediately clear. What is clear is that each learner is effective over different sets of features, so there may be scope for further work in terms of applying meta-classification techniques such as stacking in order to further improve results.

It is important to consider the implications of using user-level features for performing a classification task over the 'quality' of a post. The fact that user-level features in isolation can perform better than the baseline is a strong case for the argument that users are consistently good or consistently bad, indicating that the quality of a user's previous posts is a good predictor for the quality of future posts. However, we also expect that the quality of an individual post can vary; it therefore makes sense that the best results we have obtained use a mixture of features, some reflecting purely the content of the post, and some reflecting the overall posting trends of the user.

7 Further Work

Previous studies have either used only a single classification method [5, 16, 17, 18], or have not found significant differences between the relative performance of learners with respect to a given feature set [2]. However, we have seen that over the data being examined in this paper, learners respond better to particular feature sets. We intend to investigate this further by applying the technique to a wider variety of tasks over a greater number of datasets.

We have also adopted a relatively simplistic approach to aggregating post-level features at a user level, simply computing the arithmetic mean of the feature values. Further work would involve taking more information into account, for example the variance and skew in each post-level feature when examined at a user-aggregate level. Another dimension to be taken into account is that a user's knowledge and attitude evolve over time, so we may need to introduce some kind of temporal weighting to the post-level features we aggregate to produce the user-level profile.

Finally, it is also important to note that the gold-standard labels are provided by anonymous internet users, and that each post often has only a single annotation. It is therefore difficult to establish exactly how well the annotation reflects the opinion of the entire community with respect to the post annotated. Further work in this respect would involve establishing datasets where there are a number of annotations for each post, so as to be able to judge inter-annotator agreement and have a feeling for the upper bound in terms of possible classifier performance on the task.

8 Conclusion

In this paper, we have shown that user-level features can improve performance over classification tasks involving posts. We started by defining threaded discourse as an umbrella term for online discussions, and deriving several sets of features for describing users based on techniques for describing other aspects of the threaded discourse.

We evaluated our features over a dataset that has been used in previous research, defining a task similar to that previously investigated. We established a majority-class baseline for the task, as well as a benchmark result based on a conventional bag-of-words model for each post. We investigated our feature sets in isolation, as well as their interactions, across three different off-the-shelf learners. We found that in general, user-level features performed significantly better than simple BoW features on the given task, and that different learners seemed to prefer a different combination of feature sets.

We succeeded in our primary goal of showing that user-level features are effective in classifying posts according to quality, and we expect that the use of these features will generalize well to tasks over other aspects of threaded discourse, for example in profiling users or in classifying threads.

References

[1] Rakesh Agrawal, Sridhar Rajagopalan, Ramakrishnan Srikant and Yirong Xu. Mining newsgroups using networks arising from social behavior. In *Proceedings of the Twelfth International World Wide Web Conference (WWW'03)*, pages 529–535, Budapest, Hungary, 2003.

[2] Timothy Baldwin, David Martinez and Richard Baron Penman. Automatic thread classification for linux user forum information access. In *Proceedings of the Twelfth Australasian Document Computing Symposium (ADCS 2007)*, pages 72–9, Melbourne, Australia, 2007.

[3] Adam L. Berger, Stephen A. Della Pietra and Vincent J. Della Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, Volume 22, Number 1, pages 39–71, 1996.

[4] Sergei Brin and Larry Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, Volume 30, Number 1-7, pages 107–117, 1998.

[5] Blaz Fortuna, Eduarda Mendes Rodrigues and Natasa Milic-Frayling. Improving the classification of newsgroup messages through social network analysis. In *Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management (CIKM '07)*, pages 877–880, Lisboa, Portugal, 2007.

[6] Chih-Wei Hsu and Chih-Jen Lin. BSVM-2.06. `http://www.csie.ntu.edu.tw/cjlin/bsvm/`, 2006. Retrieved on 15/09/2009.

[7] David Hull. Improving text retrieval for the routing problem using latent semantic indexing. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '94)*, pages 282–291, Dublin, Ireland, 1994.

[8] Thorsten Joachims. Text categorization with support vector machines: learning with many relevant features. In *Proceedings of the 10th European Conference on Machine Learning*, pages 137–142, Chemnitz, Germany, 1998.

[9] Jihie Kim, Grace Chern, Donghui Feng, Erin Shaw and Eduard Hovy. 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, 2006.

[10] Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI-95)*, pages 1137–1145, Montréal, Canada, 1995.

[11] Lillian Lee. Measures of distributional similarity. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 25–32, College Park, USA, 1999.

[12] Andrew Kwok-Fai Lui, Siu Cheung Li and Sheung On Choy. An evaluation of automatic text categorization in online discussion analysis. In *Proceedings of the Seventh IEEE International Conference on Advanced Learning Technologies (ICALT 2007)*, pages 205–209, Niigata, Japan, 2007.

[13] Marco Lui and Timothy Baldwin. hydrat. `http://hydrat.googlecode.com`, 2009. Retrieved on 15/09/2009.

[14] Robert Malouf. A comparison of algorithms for maximum entropy parameter estimation. In *Proceedings of the 6th Conference on Natural Language Learning (CoNLL-2002)*, pages 49–55, Taipei, Taiwan, 2002.

[15] Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, UK, 2008.

[16] Nayer Wanas, Motaz El-Saban, Heba Ashour and Waleed Ammar. Automatic scoring of online discussion posts. In *Proceedings of the 2nd ACM Workshop on Information Credibility on the web (WICOW '08)*, Napa Valley, USA, 2008.

[17] Markus Weimer and Iryna Gurevych. Predicting the perceived quality of web forum posts. In *Proceedings of the 2007 Conference on Recent Advances in Natural Language Processing (RANLP-07)*, Borovets, Bulgaria, 2007.

[18] Markus Weimer, Iryna Gurevych and Max Mühlhäuser. 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, 2007.

[19] Le Zhang. Maximum entropy toolkit. `http://homepages.inf.ed.ac.uk/lzhang10/maxent_toolkit.html`, 2004. Retrieved on 15/09/2009.