Multi-Label Classification of Short Text: A Study on Wikipedia Barnstars

Hitesh Sajnani **Sara Javanmardi**
University of California Irvine
hsajnani@uci.edu, sjavanma@ics.uci.edu

David W. McDonald
University of Washington
dwmc@uw.edu

Cristina V. Lopes
University of California Irvine
lopes@ics.uci.edu

Abstract

A content analysis of Wikipedia barnstars personalized tokens of appreciation given to participants reveals a wide range of valued work extending beyond simple editing to include social support, administrative actions, and types of articulation work. Barnstars are examples of short semi-structured text characterized by informal grammar and language. We propose a method to classify these barnstars which contain items of interest into various work type categories. We evaluate several multi-label text categorization classifiers and show that significant performance can be achieved by simple classifiers using features which carry context extracted from barnstars. Although this study focused specifically on work categorization via barnstar content for Wikipedia, we believe that the findings are applicable to other similar collaborative systems.

Introduction

A critical challenge for large online communities is supporting users in understanding the activities of others. When a community is small one can read, skim or view much of the content or interactions among the community members. However, as a community grows this is no longer practical. The application of text processing and machine learning techniques show promise in assisting users to understand the community and the various roles that members fill. Large communities are based on user generated content (UGC), with some of the most popular relying on very short text contributions. Unlike traditional web documents, these text and web segments are usually noisier, less topic–focused, and much shorter. They consist of from a dozen words to a few sentences. As a result, traditional machine learning methods struggle to achieve desired accuracy due to the data sparseness.

In this paper, we study the classification of Wikipedia barnstars; micro-text that is exchanged on users personal pages. Barnstars are generally used to acknowledge and thank a Wikipedian for some valuable work they performed. We extract hundreds of features that capture the topic and the context of the barnstar text and use those features to classify the text into seven different broad types of work. Using state of the art multi-label classification techniques, we show that in spite of characteristics of the short text which make them difficult to classify, significant performance can be achieved by simple classifiers using features which carry context. This paper makes two contributions. We provide a preliminary analysis of multi-label classification of barnstar micro-text despite the dynamic characteristics that poses classification challenges. Our analysis can guide the further development of multi-label classification of micro-text data. Second, we explain and illustrate the value of contextual features in achieving better classification results even with simple classifiers.

The next section describes the background and highlights the related work in the domain and our motivation behind this work. Next we describe the corpus used in this study and the preprocessing performed on it. After describing the corpus, we outline the techniques, classifiers and performance evaluation measures used for our experiments. A subsequent section details the experiment setup and the outcomes. We close with some implications of our findings and future work.

Background

Most text analysis research to date has been on well-formed text documents. Researchers try to discover and thread together topically related material in these documents. However short text mediums are becoming popular in todays societies and provide quite useful information about peoples current interests, habits, social behavior and tendencies (Khan et al. 2002; Köse, Özyurt, and Amanmyradov 2007; Rosa and Ellen 2009). A number of classification research studies with various interests like detecting general emotion cues, empathy, verbal irony, certainty (or confidence) have been done (Glazer and Courtney 2002). These studies focus on binary classification. Other studies with interest to cluster items based on the short text are also done (Banerjee, Ramanathan, and Gupta 2007; OConnor, Krieger, and Ahn 2010). Some research has also been done in the direction to improve the classification using personalized features, for example, ChatTrack (Bengel et al. 2004), a text classification system, creates a concept based profile that represents a summary of the topics discussed in a chat room or by an individual participant. Use of domain specific features extracted from authors profile for short text classification in

twitter is also explored to a certain extent (Sriram et al. 2010) . Researchers have also defined the concept of event in the social text streams as a set of relations between social actors on a specific topic over a certain time period. They employ some diverse features such as social networks and inherent links (Zhao and Mitra 2007)

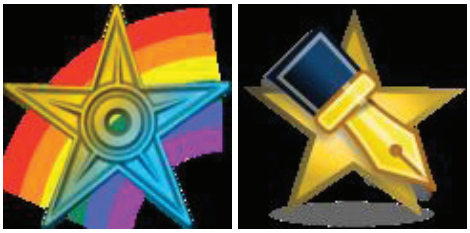
These provide inspiration for our work, demonstrating that traditional classification algorithm approaches such as bag of words and term-frequency have limitations when applied to short text. We address some limitations of prior approaches by extracting additional features which carry context. Using these features, we do multi-label classification of micro-text, which has not been well explored, as multi-label classification itself is in its infancy.

Corpus

For this study, our initial labeled training dataset comes from some prior work studying barnstars (Kriplean, Beschastnikh, and McDonald 2008). Barnstars were extracted from the November 2006 English Wikipedia database dump using a hand-tuned parser to generously identify candidate barnstars. The parsing extracted 14,573 barnstars given to 4,880 unique users. A labeling codebook was developed through an initial open coding of a random sample of 200 barnstars. The codebook was validated and refined based on a second random sample of 200 barnstars. The codebook was then used to iteratively code a random sample of 2,400 barnstars, excluding the prior 400 barnstars. The barnstars were divided randomly into six bins. Pairs of coders from our research group were then assigned to independently code the barnstars in each bin. After the initial independent coding, one coder reviewed the codes and noted all discrepancies. Of the 2,400 multi-labeled barnstars, 274 were determined to be clear parsing errors and were removed from the set. The remaining 2,126 barnstars were used as the training set for all of the following experiments. A second random sample of 586 barnstars (omitting all previously selected barnstars) was selected and independently coded to create a test set of labeled barnstars. Of those, a total of 108 were determined to be parsing errors and were removed from the set, yielding 478 coded barnstars for our test set.

Barnstar Description

In its simplest form, a barnstar is an image accompanied by a short personalized statement of appreciation for the work of another Wikipedia editor. Two example barnstars are shown in Figure 1, along with their accompanying text. Barnstars were invented to allow individuals to recognize the valuable contributions of others. Anyone can create, copy, customize and award these tokens to any other Wikipedia editor. Givers typically post barnstars to the recipients user or user talk page. Barnstars seem to carry relatively high value given their prominence on some users' pages. Some users will move their barnstars to a "gallery" of Wikipedia related achievements. While barnstars usually acknowledge some form of work, they can serve multiple purposes. They can serve as an apology, recognize otherwise unappreciated work, encourage new editors, foster competition, or even to antagonize a recipient. While we focus on Wikipedia barnstars, there are other wiki based communities that have adopted barnstars as a form of user recognition.



(a) Barnstar-1 (b) Barnstar-2

Figure 1: Barnstar Images

Barnstar-1: The Special Barnstar. This is in recognition of the many times since my first edit in March 2005 that you've answered my questions, cleaned up after me with your mop and bucket, and generally made me a better Wikipedian. anonymous user: 09:24, 14 August 2008 (UTC)
Barnstar-2: Congratulation on reaching 100000 edits. You have achieved a milestone that very few editors have been able to accomplish. The Wikipedia Community thanks you for your continuing efforts. Keep up the good work! anonymous user: 06:01, 7 April 2011 (UTC)

Feature Extraction and Normalization

Two types of features are extracted from our corpus: textual features and context specific features. Textual features are the words that users would see when reading the text of a barnstar. We first normalize the text by converting it to lower–case, removing stop words, HTML and wikitext markup, and timestamps. The cleaned text is then tokenized to collect unigrams (individual words) and calculate their frequencies across the entire corpus. This results in 54,320 unique unigrams. We condense this feature set by only considering unigrams with a frequency greater than 6, which results in 714 unigram features. This processing of textual features is similar to that of most prior work. However, processing context specific features is specific to the domain of Wikipedia.

Appropriately handling context specific features is easier when one knows the domain. In the case of barnstars, the Wikipedia context begins to scope how the text is used, where it is applied, and the types of things that are likely to be mentioned. This is different from some very open ended social media systems like twitter or other news feeds. In the case of barnstars we can apply what we know about the social practices and norms of Wikipedia to help identify likely features, extract those features, and then apply ML techniques to see if those features improve overall classification performance.

Contextual features relate to a range of normative activity in Wikipedia. One contextual feature, which builds off of textual features are phrases, names and titles. An easy illustrative example is the way barnstars are named. While any individual can create a new type of barnstar there are

still many barnstars that are well recognized and specifically named. For example, ”The Resilient Barnstar”, ”The da Vinci Barnstar” or ”The Original Barnstar” are all well know titles. Certain names, titles and other phrases form critical n-gram features that are well recognized by Wikipedians and frequently used. These are potentially important features which need to be extracted, but which may or may not always be important to a classification.

But contextual features can go beyond single terms or phrases. For example, Wikipedia has a rich set of policies that are used to facilitate the creation of the encyclopedia and to help resolve disputes about content and behavior. Policies can be mentioned as a textual feature, but are potentially more important when embedded as a link. Thus, wikitext and other aspects of the markup are potentially important features that need to be extracted. Links to policy pages, article pages, other user pages, the use of particular templates, links to images or the embedding of images, are all potentially important. In this case we extracted the names and links of all Wikipedia policies and guidelines. We also extracted links and image file names that are often embedded in a barnstar. For example, the image ”barnstar of reversion2.png” is a common image used by barnstars that recognize effort to repel vandalism and ”wmbarnstar.png” is a common image in barnstars that recognize sustained effort in editing articles.

In total, we extracted 286 contextual features. Table 1 shows some common features for each activity category. For example, ”support me for adminship” or ”new mop” are n-gram features for detecting barnstars in Administrative category, whereas ”[[wp:ppol — policy]]” is a common contextual feature relating to barnstars that acknowledge another’s collaborative action. In general, each barnstar (represented by barnstar id) has a corresponding class vector, consisting of a series of boolean values signifying class membership for that barnstar based on extracted features.

Dimension of Activity	Features
Administrative (A)	Admin, sysoup, new mop, username block, supervision, my, requests for adminship
Border Patrol (B)	revert, vfd, rfcu, copyright, wp:cp, patrol, personal user awards, candidates for speedy deletion, rc patrol
Collaborative Action (C)	consensus, survey, humor, reconciliation, rationality, npov, [[wp:ppol — policy]], templates for deletion
Editing (E)	make this article, numerous, contributions, typo, minor edit, categorization, wikifying, restructuring, reference, templates for deletion
Meta-Content (M)	tagging speedy deletes, infobox, logo, article assessment, format, css class, user categories, how to copyedit
Social and Community (S)	commitment, persistence, up to date, for founding, esperanza, nomination for wikimedals
Undifferentiated (U)	anniversary, cake, birthday, promotion, count of

Table 1: Examples of contextual features

Multi-Label Classification

Classifiers

We used k-Nearest Neighbor (k-NN), Naïve Bayes, Sequential Minimal Optimization (SMO), and Multilayer Perceptron (MLP), classification algorithms (Manning, Raghavan, and Schutze 2008) with two types of Problem Transformation (PT) techniques – Binary Relevance (BR) and Label Powerset (LP) – along with Algorithm Adaptation (AA) technique. All classifiers are well known for text categorization, except for MLP, which we chose because of the multi-label nature of classification problem. For k-NN, the values of k ranged from 1 to 20. The best value of k was chosen by taking the average of the ROC-AUC values over all the folds for different values of k and choosing the value of k which performed best on our training data for the class in consideration. We obtained results by tuning various parameters in the classifiers using 3 and 5 fold cross validation as mentioned in the Experiment section of the paper. All the experiments were performed three times. Once with total set of features, second with only textual features selected using collection frequency greater than or equal to 6 and finally using only contextual features.

Performance Measures

Multi-label classification requires different evaluation measures than traditional single-label classification. Taxonomy of multi-label classification evaluation measures is given in (Tsoumakas, Katakis, and Vlahavas 2010; Zhang and Zhou 2007). They consider three main categories: example-based, label-based measures, and ranking based. As in our case, the probabilistic estimate of the labels makes more sense than the ranking of the labels for each test instance; we decided to look at label based measures for our evaluation. As AUC is a label-based measure (i.e. a measure that looks at each label independently), we present both the micro-averaged and macro-averaged values of AUC for our experiments. The micro-averaged calculations (Micro-AUC) give equal weight to each data instance (barnstar), while the macro-averaged values (Macro-AUC) give equal weight to each label.

Problem Transformation Methods

This section presents different Problem Transformation methods (Tsoumakas, Katakis, and Vlahavas 2010) that we use for multi-label classification of Wikipedia barnstars.

Binary Relevance (BR) Using this technique (Tsoumakas, Katakis, and Vlahavas 2010) we learn —L— binary classifiers H1: X true, false, one for each different label l in L. It transforms the original data set into —L— data sets Dl that contain all examples of the original data set, labeled as true if the labels of the original example

Classifier	Micro-AUC	Macro-AUC	Features
BR-kNN	0.8831	0.8421	CF
	0.8071	0.7195	TF
	0.8418	0.7897	CF+TF
Naïve Bayes	0.8708	0.8434	CF
	0.7957	0.7304	TF
	0.8329	0.8043	CF+TF
MLP	0.8449	0.7807	CF
	0.7655	0.7360	TF
	0.7547	0.7323	CF+TF
SMO	0.7070	0.6374	CF
	0.5184	0.5293	TF
	0.7680	0.7350	CF+TF

Table 2: Performance summary for Binary Relevance
CF: contextual features, TF: textual featuresresults

Classifier	Micro-AUC	Macro-AUC	Features
BR-kNN	0.7798	0.7565	CF
	0.6474	0.6540	TF
	0.7585	0.7429	CF+TF
Naïve Bayes	0.8230	0.8023	CF
	0.7233	0.7156	TF
	0.6930	0.7141	CF+TF
MLP	0.6504	0.6521	CF
	0.4105	0.4909	TF
	0.3463	0.4081	CF+TF
SMO	0.2482	0.3160	CF
	0.3869	0.4630	TF
	0.7307	0.6440	CF+TF

Table 3: Performance summary for Label Powerset
CF: contextual features, TF: textual featuresresults

contained 1 and as false otherwise. It is the same solution used in order to deal with a single-label multiclass problem using a binary classifier. For the classification of a new instance x this method outputs as a set of labels the union of the labels that are output by the —L— classifiers.

Table 2 shows the experimental results with contextual features, textual features, and combined feature set including both textual and contextual features for each classifier. We have 286 contextual features and 714 textual features. So totally we have 1000 features in the combined feature set.

According to the results in Table 2, most of the classifiers achieve higher performance when they are learned only based on contextual features.

For example, the BR-kNN classifier outperforms the rest with respect to Micro-AUC measure. Similarly, Naïve Bayes outperforms the rest with respect to Macro-AUC measure. Surprisingly, SMO produces lower classification performance compared to BR-kNN and Naïve Bayes.

Label Powerset (LP) BR ignores label correlation as it looks at each class independently. LP technique considers the notion of label correlation by treating each distinct occurrence of a label pattern as a new class, more formally LP considers each different set of labels that exist in the multi-label data set as a single label. It learns one single-label classifier H: X P (L), where P (L) is the power set of L (Tsoumakas, Katakis, and Vlahavas 2010). Table 3 shows the experimental results with Contextual features, textual features, and combined feature set including both textual and contextual features for each classifier.

The Naïve Bayes classifier using 286 contextual features performed the best in the experiment, with respect to both Micro-AUC and Macro-AUC measure score. k-NN showed up as the second best classifier. The difference between the results of these two classifiers is significant and cannot be ignored. SMO performed relatively well with combined feature set. However it produced very poor results with only contextual and textual features. Unlike BR, the observation of contextual features always performing better than any other setting doesnt hold with LP. However in all cases, either contextual feature alone or the combinations of contextual features with textual features have produced the best results. This signifies that contextual features play important role in achieving better classification. There is a decline of 15% in the best performance value for k-NN and Naïve Bayes and about 25% compare the values of SMO and MLP. In general, the results of LP technique are much lower than that obtained in the BR technique. LP considers label correlation using data instances as it treats every distinct label pattern as a single label. It means that it is ignorant about any label combination which does not exist in the training set. In addition there might be very few data cases for some label combination. Hence this method performs badly if either there are lots of different combinations of labels in the test set which did not exist in train set or there are very few examples for such cases for the classifier to learn them correctly. It turned out that our data set had 7 such combinations in test which did not occur at all in the training. There were 16 cases which had less than 3 examples in the train set. Hence, since the dataset is lightly multi-labeled, LP was not the right method for this data set.

Algorithm Adaptation Method

The algorithm adaptation method extends machine learning algorithms in order to handle multi-label data directly (Tsoumakas, Katakis, and Vlahavas 2010). This is different from Problem Transformation technique as it does not modify the data set. Table 4 describes our results for the above listed classifiers using this technique.

The Multi-Label k-NN (MLkNN) classifier, which is an extension of the k-NN for multi-label data as described in by (Zhang and Zhou 2007), performed the best in this experiment, with respect to both Micro-AUC and Macro-AUC measure followed by Back Propagation Multi-Label Learning (BPMLL) classifier (Tsoumakas et al. 2010). The difference between the results of these two classifiers is significant. Surprisingly, Naïve Bayes did not perform well in this setting. Like BR the order of performance for the classifiers is determined by the type of features, with contextual features giving the best results and textual features giving relatively poorer results.

Classifier	Micro-AUC	Macro-AUC	Features
MLkNN	0.8710	0.8264	CF
	0.7960	0.7014	TF
	0.8200	0.7584	CF+TF
Naïve Bayes	0.6959	0.5785	CF
	0.6762	0.5300	TF
	0.7118	0.6705	CF+TF
BPMLL	0.7292	0.5287	CF
	0.7119	0.5328	TF
	0.6743	0.4667	CF+TF
SMO	0.5000	0.5201	CF
	0.5118	0.5106	TF
	0.5823	0.5541	CF+TF

Table 4: Performance summary for Algorithm Adaptation CF: contextual features, TF: textual featuresresults

Comparison of Methods

We made local extensions and customization in Java, within the framework of Mulan (Tsoumakas et al. 2010) and Weka (Hall et al. 2009) to set up the experiments. We experimented with two PT methods and Algorithm Adaptation methods in conjunction with various classifiers. Figure 2 shows the comparative analysis of the performance of these methods with contextual features.

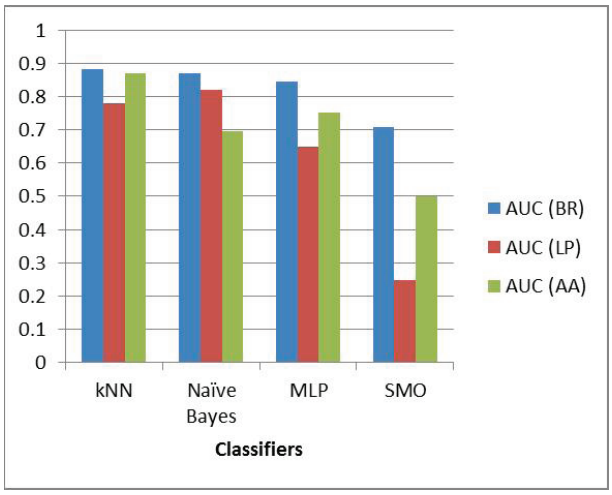


Figure 2: Comparative analysis of classifiers based on Multi-label classification techniques

The following observations can be made from the experiments.

- BR technique performed better than LP technique and Algorithm Adaptation technique, irrespective of the choice of classifier
- k-NN and Naïve Bayes produced the best results irrespective of the choice of technique used

Figure 3 shows the comparative analysis of classifiers with respect to the type of the features. The performance of the all the classifiers, except SMO, is the best when using contextual features. In case of SMO, combined features produce

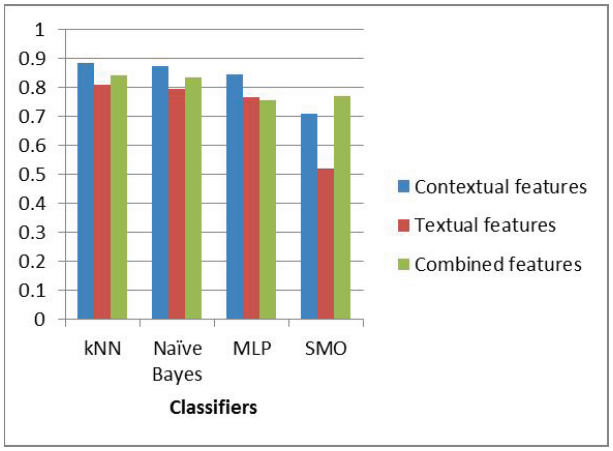


Figure 3: Comparative analysis of classifiers based on feature types

the best results. However, in either case, the influence of contextual feature is visible in boosting the classification performance. In aggregate, the combination of k-NN algorithm together with BR gives the best results. In case of LP, Naïve Bayes classifier gives the best results. SMO in combination with any method produced the worst results. However since SMO requires careful tuning to work well with a dataset, optimizations might give better results.

These are interesting observations as SMO is a very popular classifier for text classification. Also, LP is the most popular method in the literature and BR is not as popular as LP (Tsoumakas, Katakis, and Vlahavas 2010).

Discussions and Implications

BR performed the best on our data set. However this may vary based on the properties of the data set. Tsoumakas et. al. [10] showed label cardinality and label density to be the parameters that influence the performance of the different multi-label methods. Let D be a multi-label data set consisting of —D— multi-label examples, then label cardinality of D is the average number of labels of the examples in D and label density of D is the average number of labels of the examples in D divided by —L—. Table 5 presents the values of these metrics for our data set.

Type	Label Cardinality	Label Density	#Distinct LabelSets
Train Set	1.246	0.178	51
Test Set	1.162	0.166	24

Table 5: Multi-label Metrics for the data set

The low value of label cardinality and label density justifies the better performance of BR technique compared to other techniques. Figure 4 shows the graph of occurrences of the number of labels in the data cases.

The x-axis is a logarithmic scale representing the number of data instances, whereas y-axis represents the number of labels that appear together for the data instances. Out of

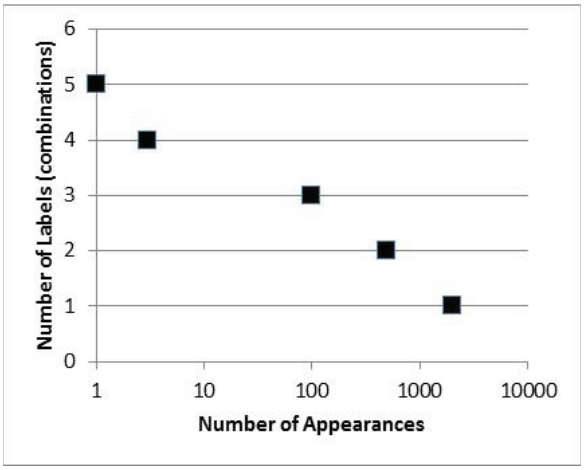


Figure 4: Label occurrences (frequency) in the data set

2,547 data instances, 1,999 data instances had single label value, only one data instance had 5 labels, and none had all the 7 labels associated with it. These observations are true reflection of the dynamics of an online collaborative community and our perspective toward it. Most of the time, we associate a contributor with a single activity.

Conclusion

In this paper we discussed the multi-label short text classification and its application in classifying Wikipedia barnstars. Unlike traditional classification of short text which is mostly based on textual features such as word frequency or TF-IDF, in this work we introduced contextual features, which are domain specific. We used different multi-label classification techniques and classifiers, and showed that k-NN and Naïve Bayes outperforms the rest.

In addition, we showed that the role of contextual features is significant in boosting the classification performance, as well as reducing the dimension of the feature space.

Although these results are based on our study on Wikipedia barnstars, we believe that the findings can be extended to any acknowledgement based system, as they share similar culture, and hence similar text characteristics. However, as each data set has some influence of domain and community, we welcome further studies exploring how classification results differ across various data sets with respect to changes in domain and community. Nonetheless, our experiments can serve as a pilot for multi-label short text classification and the use of contextual features.

Acknowledgement Authors would like to thank Gregory Tsoumakas for his valuable feedback during the course of this work. Portions of this material are based on work supported by NSF Grant IIS-0811210.

References

Banerjee, S.; Ramanathan, K.; and Gupta, A. 2007. Clustering short texts using wikipedia. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR'07, 787–788. ACM.

Bengel, J.; Gauch, S.; Mittur, E.; and Vijayaraghavan, R. 2004. Chat room topic detection using classification. In *In proceeding of 2nd Symposium on Intelligence and Security Informatics*, 266–277.

Glazer, and Courtney. 2002. Playing nice with others: The communication of emotion in an online classroom. In *9th Annual Distance Education Conference*.

Hall, M.; Frank, E.; Holmes, G.; Pfahringer, B.; Reutemann, P.; and Witten, I. H. 2009. The weka data mining software: An update. *SIGKDD Explorations* 11.

Khan, F.; Fisher, T.; Shuler, L.; Tianhao, W.; and Pottenger, W. 2002. Mining chat-room conversations for social and semantic interactions. In *LU-CSE-02-011*.

Köse, C.; Özyurt, O.; and Amanmyradov, G. 2007. Mining chat conversations for sex identification. In *Proceedings of the 2007 international conference on Emerging technologies in knowledge discovery and data mining*, PAKDD'07, 45–55. Berlin, Heidelberg: Springer-Verlag.

Kriplean; Beschastnikh, I.; and McDonald, D. 2008. The dimensions of wikiwork: Uncovering valued work in wikipedia through barnstars. In *Proceedings of the 2008 ACM Conference on Computer Supported Cooperative Work*, CSCW '08.

Manning; Raghavan, P.; and Schutze, H. 2008. Introduction to information retrieval. 1263 – 1284.

OConnor, B.; Krieger, M.; and Ahn, D. 2010. Tweetmotif: Exploratory search and topic summarization for twitter. In *Proceeding of the International AAAI Conference on Weblogs and Social Media*, ICWSM '10.

Rosa, K. D., and Ellen, J. 2009. Text classification methodologies applied to micro-text in military chat. In *International Conference on Machine Learning and Applications*, ICMLA '09, 710–714.

Sriram, B.; Fuhry, D.; Demir, E.; Ferhatosmanoglu, H.; and Demirbas, M. 2010. Short text classification in twitter to improve information filtering. In *Proceedings of the 33rd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR' 10.

Tsoumakas, G.; Vilcek, J.; Spyromitros, E.; and Vlahavas, I. 2010. Mulan: A java library for multi-label learning. *Journal of Machine Learning Research*.

Tsoumakas, G.; Katakis, I.; and Vlahavas, I. 2010. Mining multi-label data. *Data Mining and Knowledge Discovery Handbook* 667–685.

Zhang, M.-L., and Zhou, Z.-H. 2007. Ml-knn: A lazy learning approach to multi-label learning, pattern recognition. *Pattern Recognition* 40(7):2038–2048.

Zhao, Q., and Mitra, P. 2007. Study of static classification of social spam profiles in myspace. In *Proceeding of the Int'l AAAI Conference on Weblogs and Social Media*, ICWSM '07, 82–89.