Detecting User Emotions in Twitter through Collective Classification

İbrahim İleri and Pinar Karagoz
Department of Computer Engineering, Middle East Technical University, 06800, Ankara, Turkey

Keywords: Social Networks, Emotion Analysis, Sentiment Analysis, Collective Classification.

Abstract: The explosion in the use of social networks has generated a big amount of data including user opinions about varying subjects. For classifying the sentiment of user postings, many text-based techniques have been proposed in the literature. As a continuation of sentiment analysis, there are also studies on the emotion analysis. Due to the fact that many different emotions are needed to be dealt with at this point, the problem gets more complicated as the number of emotions to be detected increases. In this study, a different user-centric approach for emotion detection is considered such that connected users may be more likely to hold similar emotions; therefore, leveraging relationship information can complement emotion inference task in social networks. Employing Twitter as a source for experimental data and working with the proposed collective classification algorithm, emotions of the users are predicted in a collaborative setting.

1 INTRODUCTION

Recent advances in social networks increase the ways of explaining ideas on diverse subjects. Moreover users can share their opinions with their online friends in a collaborative manner. All that rich information sources make the social networks a suitable working base for researchers.

Effective methodologies and techniques are required to extract various kinds of information from social networks automatically. Among them, identifying users' sentiments on a product or service has turned into a valid indicator of marketing success. Apart from the classical sentiment analysis algorithms, networked data include valuable relationship information that can contribute to this analysis process beside textual contexts that are produced by the users. Such data is can be useful in emotion analysis, in which, instead of detection of sentimentality, the type of sentimentality is further detected in terms of different kinds of emotions.

In this study, collective classification algorithms, which constitute a sub-field of link mining field, are applied within the context of emotion analysis in microblogs. As the microblog, Twitter is used as the data source. In our setting, Twitter users are nodes and their relationships are edges, which are extracted from retweets or user mentions (@) in tweets. Giving graph structure as input to collective classification framework, unknown emotion labels for users are predicted by utilizing their labeled neighbors. The performance of relational classifiers are experimented under different configurations.

Since the collected tweets are in Turkish, in addition to tokenization, Turkish morphological analysis and stemming are applied as well. However, apart from this, all of the remaining methods are equally applicable to the texts in other languages as well. With the aim of applying collective classification techniques on the context of emotion analysis in social networks, to the best of our knowledge, this is the first work in the literature. The contribution of the study can be summarized as follows:

- Collective classification algorithms are applied on the context of emotion analysis in social networks. The performance under various settings are investigated as well.
- A new Twitter dataset (EmoDS) is gathered with its uniquely generated relationship information.

The paper is organized as follows. Section 2 presents a summary of the related studies in the literature, which includes emotion analysis and link-based sentiment analysis methodologies. Section 3 provides background about employed collective classification methods. Section 4 presents proposed method for emotion detection with collective classification and applied steps for data gathering, preprocessing, and relationship construction. Section 5 shows the related experiments and their results. Lastly, Section 6 con-

İleri, İ. and Karagoz, P.
Detecting User Emotions in Twitter through Collective Classification.
DOI: 10.5220/0006037502050212
In *Proceedings of the 8th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2016) - Volume 1: KDIR*, pages 205-212
ISBN: 978-989-758-203-5

cludes with final remarks about this study and gives future directions.

2 RELATED WORK

In this section, we summarize the related studies on emotion analysis and link-based classification.

2.1 Emotion Analysis

Basically, there are two main approaches in the literature for emotion detection on texts. The first one is the text classification based methods that build classifiers from labeled text data as in traditional supervised learning. The second method that detects the emotional states especially in tweets is the lexicon-based approach. For this approach, an emotion lexicon is constructed.

Regarding the field of psychology, Ekman (Ekman, 1999) defined 7 emotions that are categorized by observable human facial expressions. Kozareva et al. (Kozareva et al., 2007) classified news headlines using these emotion classes. They averaged different web search engines hit counts' on the emotional classes and news headlines as query words.

Alm et al. (Alm et al., 2005) presented empirical results of applying supervised machine learning techniques to categorize English fairy tale sentences into different emotions. They proposed their own text-based classifier algorithm (SNoW) and it achieved significant accuracy results. Go et al. (Go et al., 2009) applied supervised learning methods to classify collected Twitter data into binary sentiments as positive or negative.

Boynukaln (Boynukalin, 2012) worked on two data sets. One of them is the Turkish translation of ISEAR [1] data and the other is the manually labeled Turkish fairy tales. Emotion levels are predicted using different n-gram feature constructions and weighted log likelihood algorithm (Nigam et al., 2000) is utilized to determine most significant features.

Akba et al. (Akba et al., 2014) investigated the feature selection methods on Turkish movie reviews. They labeled their corpus by dividing emotions into three categories as positive, negative and neutral. In their experiments, supervised methods had been employed for the classification of movie reviews into two or three categories. Tocoglu et al. (Tocoglu and Alpkocak, 2014) proposed an emotion extraction system from Turkish texts, which is based on text classification approach. Applying Naive Bayes classifier in Weka achieved promising accuracy result.

[1] http://www.affective-sciences.org/researchmaterial

Demirci (Demirci, 2014) classified Turkish tweets into six emotion categories (anger, surprised, fear, sadness, joy and disgust) with supervised learning. Beside the classical text preprocessing operations, morphological analysis is applied as well. Finally, several supervised classification methods are compared with the baseline algorithm of Boynukaln (Boynukalin, 2012).

2.2 Link-based Classification

In this work, a subfield of link mining, which is called collective classification, is used. Briefly, it aims to predict the labels of objects by using relationships among them. The main challenge is to design an algorithm for collective classification that uses associations between object classes and jointly infer their labels in the graph.

Chakrabarti et al. (Chakrabarti et al., 1998) work on the problem of categorizing related news objects in the Reuters dataset. They are the first to leverage class labels of related instances and also their attributes. Although using class labels improves classification accuracy, the same thing does not apply for considering attributes.

Neville and Jensen (Neville and Jensen, 2000) propose a simple link based classification method that classifies corporate datasets involving heterogeneous graphs with different set of features.

Lu and Getoor (Lu and Getoor, 2003) aim to enhance traditional machine learning algorithm by introducing new features that are built out of correlations between objects. As a result, a new link based classification algorithm that uses probability terms such as Markov blanket of related class labels, is developed.

Pang and Lee (Pang and Lee, 2004) seek to determine sentiment polarities of movie reviews by extracting subjective portions of the sentences. For this purpose, they use a graph-based technique that finds the minimum cuts. By this way, contextual information is added in polarity classification process and significant accuracy improvement is achieved.

In the literature, the most similar study to our work is the one by Rabelo et al. (Rabelo et al., 2012), which proposes an user centric approach on the context of sentiment analysis. They have classified Twitter user's political opinions into binary classes by using collective classification. Their algorithm takes a partially labeled graph, applies a graph pruning process and runs the collective classification. Preliminary experiments have shown promising results.

3 COLLECTIVE CLASSIFICATION

It is an important issue to investigate how objects influence each other in network, such as how user's emotions are affected by his/her relationships in Twitter. This can be generalized to the problem of finding the labels of the entities in the network.

Collective classification can be seen as a relational optimization task for networked data. According to algorithm's nature, different relational objective functions are optimized within collective inference techniques. Constructed relational features can be used for the inference task. However, most of the local classifiers use only fixed-size feature vectors, whereas neighbor counts vary considerably in networked data. For instance, a Twitter user can have many followers. Although it is not a preferred method, by considering limited (equal) number of connections for each user, fixed-size feature vectors could be constructed.

A desirable solution is to apply aggregation techniques to summarize the node's neighborhood information. For example, the number of neighbors that have different class labels could be counted and added as a new feature to node. Class labels may be replaced or supported with local attributes. For numerical attributes, it is also possible to use statistical methods such as minimum, maximum, median, mode, ratio.

On the other hand, for each pair of neighboring nodes, similarities of their local attributes can be considered exactly. In this study, a similar method is discussed but not only implemented as an aggregation method but also used as a weight in relational probability calculations. Perlich and Provost (Perlich and Provost, 2003) discuss aggregation-based feature construction as the relational concept in more detail.

We can divide collective classification into three models. These models are described as follows:

1. **Local (Non-relational) Model.** This model is learned for target (class) variable by using the local attributes of the nodes in the network. Alternatively, classical supervised learning methods such as naive Bayes or decision trees can be employed. In this study, the priors are estimated by using Bayesian approach, which uses available local attributes of the nodes to estimate its class-probabilities.
2. **Relational Model.** Relational features and links among entities come into prominence for this component. It builds different objective functions to estimate node's target attribute probabilities with its neighborhood. It is also possible to benefit from local attributes of the neighboring nodes.
3. **Collective Inference.** Created relational objective functions are generally the joint probability distributions which are based on Markov Random Fields. For example computing relational objective functions needs collective inference methods such as iterative classification and relaxation labeling. As a result, it is aimed to find out how a node's classification is influenced from its neighbors classification in a collaborative setting.

Netkit-SRL (or Netkit) (Macskassy and Provost, 2007), is an open source Network Learning Toolkit for Statistical Relational Learning. It is coded in Java and it can be integrated with Weka (Hall et al., 2009) data mining tool. It allows to combine different types of components for relational classification on networked data as well as to design new classifier components and use them with different configurations.

Within the scope of this work, we use the following relational models. Weighted-vote relational neighbor classifier (*wvrn*) produces a weighted mean of class membership probability estimations from node's neighbors. Probabilistic relational neighbor classifier (*prn*) estimates a particular node's class label probability by multiplying each neighboring node's class prior probability values. The class distribution relational neighbor classifier (*cdrn-norm-cos*) creates an average class vector for each class of node and then estimates a label for a new node by calculating how near that new node is to each of these class reference vectors. Network-only Bayes classifier (*no-bayes*) counts the class labels of node's each neighbors. Then, this value is multiplied with prior class distributions. Estimation needs product of each neighbors observed class value probabilities conditioned on given nodes class values and getting powered with edges weights. Network-only link-based relational classifier (*nolb-lr-distrib*) firstly creates normalized feature vector of the training node via aggregating its neighbor's class attributes. Then, it uses logistic regression for relational modeling.

The main goal of collective inference algorithms is to infer the unknown class labels of nodes by maximizing the marginal probability distribution which is represented by learned objective functions from relational classifiers. In *Null inference* setting, the local classifier is applied, and then the relational classifier is applied only once. *Iterative classification* classifies the node's unknown class labels by updating current state of the graph in each iteration until every node's label is stabilized or maximum iteration count is reached. *Relaxation labeling* uses direct class estimations from learned models rather than constant labeling (e.g. as *null*). By this way, it does not miss the previously estimated probabilities in each step of the inference. Instead of updating the graph's state instantly, it stands, holds the estimations from previous iteration then use these values on the next iteration. As a result, inference is carried out simultaneously.

4 PROPOSED METHOD

This section presents the method proposed in this study. Our main objective is to collectively classify users' emotions reflected in the social network postings with the help of their relationship information. As the basic difference, by the proposed relational classifier, textual content features in the postings are also taken into consideration.

Within this study, we used the dataset that we gathered from Twitter. As the next step, tweets are preprocessed in order to construct feature vectors. Finally, collective classification algorithms are applied on our data set.

4.1 Data Gathering

Similar to the work in (Mohammad, 2012), for gathering our emotion data labeled instances, different emotion related hashtags are queried by using Twitter API. These are Turkish keywords corresponding to six emotions such that "öfke" ("anger"), "korku" ("fear"), "mutluluk" ("joy"), "üzüntü" ("sadness"), "iğrenme" ("disgust"), "şaşkınlık" ("surprise"). If the word does not return enough number of results, its derived versions are employed. As a result, 1200 labeled retweets are collected for each emotion category. Then, all of six categories retweets are merged together to obtain 7200 instances in total.

However, since our approach is user-centric, duplicate usernames are eliminated from the whole data. Finally, there are 6841 instances (unique usernames are regarded as key field) having username, (Re) Tweet and Label information. Class distributions can be seen in Table 1.

Additionally, our collected emotion dataset's text features are needed to have detailed pre-processing

Table 1: Dataset Class Label Distribution due to Instance Counts.

Class Labels	Instance Counts
anger	1118
disgust	1140
fear	1145
joy	1191
sadness	1121
surprise	1126

for the latter feature vector construction steps. Several preprocessing steps are applied on the textual content such as cleaning noise and removing hashtag and punctuation.

4.2 Feature Vector Construction and Feature Selection

After preprocessing steps, textual content of each posting is represented as a feature vector, where each one is a word extracted from the tweets. Stemming on the tokens is performed by using Zemberek (Akın and Akın, 2007), which is a Turkish morphological analysis tool.

Each individual token is also inspected for language and spell checking. Tokens that are not in Turkish are removed. In addition, for the misspellings, first corrected suggestion of Zemberek is used.

After stop-word removal, all reliable tokens are added into a common pool (*bag of words*). Elimination of the stop-words is performed by using a Turkish stopwords list [2], retrieved from a publicly accessible project. This final token pool (dictionary) constitutes the feature space for the data. On the total, the pool contains 1862 features. Features are weighted by their term counts directly.

In order to select significant features, information gain method (Kent, 1983) on Weka (Hall et al., 2009) is applied and the large feature space is reduced into 800 features (the number of best features is obtained as 800 as it is shown to provide the best accuracy in (Demirci, 2014)). Finally, the constructed dataset is named as EmoDS. A sample from EmoDS including 4 instances is shown in Table 2.

4.3 Proposed Relational Classifier for Collective Classification

In this section, we present our motivation for proposing a new relational classifier and the details of the proposed method.

McDowell and Aha (McDowell and Aha, 2013) investigate the neighboring attributes' contribution while building relational models and estimating the label with collective inferencing on partially labeled networks. They propose a probabilistic relational model for bringing neighbor attributes and labels together. Results show that using both neighbor attributes and labels on building relational model, often produces the best accuracy.

However, this study is tested under some small sparsely-labeled networked datasets. Considering

[2]https://code.google.com/p/stop-words/

Table 2: Short View of EmoDS.

Usernames	Feature Vector Values	Labels
@gioselyn_4	$< 1,1,1,1,1,0,0,1,1,\ldots >$	sadness
@Ersiyn	$< 0,0,0,1,0,0,1,1,0,\ldots >$	surprise
@Mukremin1973	$< 0,0,0,0,1,0,0,2,0,\ldots >$	fear
@Feneristcom	$< 0,0,0,0,0,0,0,1,0,\ldots >$	joy

that the approach has potential to improve accuracy for emotion detection in fully-labeled social networks and inspired by this idea, Netkit's network only bayes relational classifier (*nobayes*) is extended by adding neighbor features information into process and implemented in Netkit environment.

The proposed algorithm starts with a simple probabilistic assumption that each neighbor's features are conditionally independent. Since attributes are represented as feature vectors, cosine similarities between a node and each of its neighbors are computed for prior probability calculations, rather than simply counting the neighbors. Then, these scores are used in the objective function given in Equation 1 as a new variable.

$$P(N_i|c) = \frac{1}{Z} \prod_{v_j \in N_i} P(c_j = c_j^{'}|c_i = c)^{weight_{i,j}} \times P(simscore_j|c_i = c)^{weight_{i,j}} \quad (1)$$

In Equation 1, let c be the estimated class label value for node i, $c_j^{'}$ is the class label observed at neighbor node j and $simscore_j$ is the cosine similarity score observed at node j. $weight_{i,j}$ represents the edge weight between node i and node j (simply equal to 1 for this study). N_i is the immediate neighbors of node i and Z is standard normalization variable, which smooths the summed values on the range 0 and 1.

Relational classification based on Equation 1 is called as Network-Only Bayes-VectorSimilarity classifier, *no-bayes-vecsim* for short. Pseudo-code *nobayes-vecsim* relational classifier is shown on Algorithm 1. In the algorithm, T denotes the training set of labeled users and U is the set of unknown labeled users. The proposed algorithm expands *no-bayes* relational classifier with adding neighbor features information into process based on a simple probabilistic assumption that each neighbor's features are conditionally independent. The cosine similarity values between a node and its neighbors are computed for prior probability calculations. Hence, while estimating the proposed objective function with collective inferencing method, neighbor's features also become valuable. On the other hand, if neighbor feature vectors are disparate with unknown labeled user's feature vector, it is expected not to contribute much to label inference.

Algorithm 1: Pseudo-code of Proposed Nobayes-Vecsim Relational Classifier.

function InduceRelationalModel
for all $v_i \in T$ **do**
  // find $P(c)$
  $c\_prior\_prob\_vec \leftarrow v_i$'s class value counts
  for all $v_j \in N_i$ **do**
    // find $P(c_j = c_j^{'}|c_i = c)^{weight_{i,j}}$
    $c\_nbor\_prob\_vec \leftarrow v_j$'s class value counts powered with edge weights
    // find $P(simscore_j|c_i = c)^{weight_{i,j}}$
    $simscore\_nbor\_prob\_vec \leftarrow v_j$'s cosine similarity scores powered with edge weights
  end for
end for
Normalize each vector
end function
function ApplyEstimation
// for the inference phase, estimate Equation 1
$estimation\_prob\_vec := \{\}$
for all $v_i \in U$ **do**
  $known\_prob\_vec \leftarrow c\_prior\_prob\_vec$
  for all $v_j \in N_i$ **do**
    $known\_prob\_vec \mathrel{*}= c\_nbor\_prob\_vec * simscore\_nbor\_prob\_vec$
  end for
end for
$estimation\_prob\_vec \leftarrow known\_prob\_vec$
return $estimation\_prob\_vec$ as v_i's class estimations
end function

Another version of this relational classifier is named as *nobayes-avgsim*. It takes into account the case that a user can have different number of *retweets* under different emotion classes. However, Netkit's configuration does not support using the same username (seen as *key* attribute) under different classes.

For this reason, a different implementation strategy is employed as follows: cosine similarities between each neighbor user's feature vectors are calculated and the average is taken at the end. Then, averaged similarity score for each node is fed into relational classifier's objective function externally.

By this way, not only class labels of their friends but also their text-based attributes are taken into consideration.

Table 3: Collective Classification with No Relationships Accuracy Results (aggr. -All).

	wvrn	**prn**	**cdrn-norm-cos**	**no-bayes**	***no-bayes-vecsim***	**nolb-lr-distrib**
Null Inference	0.165	0.173	0.174	0.174	0.174	0.174
Iterative Classification	0.171	0.170	0.174	0.174	0.174	0.174
Relaxation Labeling	0.165	0.170	0.174	0.174	0.174	0.174

Table 4: Collective Classification with No Relationships Running Time (sec) Results (aggr. -All).

	wvrn	**prn**	**cdrn-norm-cos**	**no-bayes**	***no-bayes-vecsim***	**nolb-lr-distrib**
Null Inference	2	1	14	2	1	46
Iterative Classification	3	17	15	1	1	47
Relaxation Labeling	1	3	162	1	1	125

5 EXPERIMENTAL RESULTS

In this section, conducted experimental analysis and the obtained results are presented. In order to construct a network for EmoDS, by following the interaction means that are described in (Kivran-Swaine and Naaman, 2011), a set of realistic friendship relations are extracted from retweets that contain RT flags and @ mentions. Self-edges caused by self-retweets and relations that are not unique (i.e. either one of the related usernames are not located as an instance in EmoDS) are discarded. As a result, 606 realistic relationships are generated.

Experiments are run on Netkit environment under 15 different collective classification configurations. Each of these configuration's components are as explained briefly in Section 3. Results are evaluated under 10-fold cross validation.

We have compared the accuracy performance of the proposed relational classifiers, *nobayes-vecsim* and *nobayes-avgsim* with the previous collective inference-relational classifier methods.

As described in Section 4.3, according to the proposed relational classifier's nature, the algorithm checks the similarity of each user's vector to each of its neighbors' feature vector and uses it as a new variable in probability function calculation.

In the first set of experiments, we analyze the effect of *No Relationship* setting, in which the effect of network is neglected, on *nobayes-vecsim* in comparison to other methods. In Table 3, accuracy results are presented. In Table 4, time efficiency performance is shown.

In the second set of experiments, we use *All Relationships* setting to compare the performance of *nobayes-vecsim*. In Table 5, accuracy results are given, whereas in Table 6, time efficiency performance is given. In Table 7 and Table 8, performance of *nobayes-avgsim* is compared against other methods.

As seen in the results, proposed relational classifier provides small performance gain (˜5%) in collective classification accuracies in comparison to those classifiers that can execute in reasonable amount of time. One possible reason for this limited increase in accuracy is that generated sparse user friendship relations are yielded in different ranges among users such that some users do not share the same emotions with their friends, as opposite to homophily principle.

Proposed relational classifier (*nobayes-vecsim*) leads to a small increase in collective classification accuracy. However, it consumes considerable amount of time. This situation is improved with *nobayes-avgsim* relational classifier. In *nobayes-avgsim*, by calculating vector similarities before the classification, almost the same accuracy results are obtained in less time.

6 CONCLUSION AND FUTURE WORK

In this study, user emotions in social networks are aimed to be predicted in a collaborative setting. To this aim, we have collected a data set from Twitter and constructed a network through RT and mention activities among users.

We propose a new relational classifier and a vari-

Table 5: Collective Classification with All Relationships Accuracy Results (aggr. -All).

	wvrn	**prn**	**cdrn-norm-cos**	**no-bayes**	***no-bayes-vecsim***	**nolb-lr-distrib**
Null Inference	0.187	0.192	0.190	0.207	0.206	0.220
Iterative Classification	0.187	0.195	0.189	0.207	0.211	0.223
Relaxation Labeling	0.192	0.186	0.190	0.208	0.208	0.220

Table 6: Collective Classification with All Relationships Running Time (sec) Results (aggr. -All).

	wvrn	**prn**	**cdrn-norm-cos**	**no-bayes**	***no-bayes-vecsim***	**nolb-lr-distrib**
Null Inference	2	2	14	2	11	498
Iterative Classification	3	20	15	1	11	525
Relaxation Labeling	1	3	173	1	11	711

Table 7: Collective Classification with All Relationships Accuracy Results-2 (aggr. -All).

	wvrn	**prn**	**cdrn-norm-cos**	**no-bayes**	***no-bayes-avgsim***	**nolb-lr-distrib**
Null Inference	0.190	0.188	0.193	0.213	0.211	0.237
Iterative Classification	0.194	0.195	0.190	0.217	0.210	0.231
Relaxation Labeling	0.195	0.189	0.195	0.214	0.208	0.235

Table 8: Collective Classification with All Relationships Running Time (sec) Results-2 (aggr. -All).

	wvrn	**prn**	**cdrn-norm-cos**	**no-bayes**	***no-bayes-avgsim***	**nolb-lr-distrib**
Null Inference	1	1	15	1	1	659
Iterative Classification	3	20	16	1	1	580
Relaxation Labeling	1	2	181	1	1	880

ation of it. We have investigated the performance of the proposed methods in comparison to five different relational classifier setting with three different collective inference model within NetKit. The proposed models' accuracy values are close to the best accuracy obtained by the existing models. Furthermore, the proposed models improve the execution time considerably.

It is possible to extend current study in several directions. Edge weights are assumed to be equal (with value 1) in the experimental setting. For the networks that include weighted edges, different link mining techniques can be considered such as edge selection or handling heterogeneous links. Edge selection can propose techniques analogous to those used in traditional feature selection. Proposed and existing methods can be extended to handle time-dependent emotional data that have changes in user emotions along the time.

REFERENCES

Akba, F., Uçan, A., Sezer, E. A., and Sever, H. (2014). Assessment of feature selection metrics for sentiment analyses: Turkish movie reviews. In *8th European Conference on Data Mining 2014*.

Akın, A. A. and Akın, M. D. (2007). Zemberek, an open source nlp framework for turkic languages. *Structure*, 10:1–5.

Alm, C. O., Roth, D., and Sproat, R. (2005). Emotions from text: machine learning for text-based emotion prediction. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 579–586. Association for Computational Linguistics.

Boynukalin, Z. (2012). Emotion analysis of turkish texts by using machine learning methods. Master's thesis, Middle East Technical University.

Chakrabarti, S., Dom, B., and Indyk, P. (1998). Enhanced hypertext categorization using hyperlinks. In *ACM SIGMOD Record*, volume 27, pages 307–318. ACM.

Demirci, S. (2014). Emotion analysis on turkish tweets. Master's thesis, Middle East Technical University.

Ekman, P. (1999). Facial expressions. *Handbook of cognition and emotion*, 16:301–320.

Go, A., Bhayani, R., and Huang, L. (2009). Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, 1:12.

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

Kent, J. T. (1983). Information gain and a general measure of correlation. *Biometrika*, 70(1):163–173.

Kivran-Swaine, F. and Naaman, M. (2011). Network properties and social sharing of emotions in social awareness streams. In *Proceedings of the ACM 2011 conference on Computer supported cooperative work*, pages 379–382. ACM.

Kozareva, Z., Navarro, B., Vázquez, S., and Montoyo, A. (2007). Ua-zbsa: a headline emotion classification through web information. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 334–337. Association for Computational Linguistics.

Lu, Q. and Getoor, L. (2003). Link-based classification. In *ICML*, volume 3, pages 496–503.

Macskassy, S. A. and Provost, F. (2007). Classification in networked data: A toolkit and a univariate case study. *The Journal of Machine Learning Research*, 8:935–983.

McDowell, L. K. and Aha, D. W. (2013). Labels or attributes?: rethinking the neighbors for collective classification in sparsely-labeled networks. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 847–852. ACM.

Mohammad, S. M. (2012). # emotional tweets. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 246–255. Association for Computational Linguistics.

Neville, J. and Jensen, D. (2000). Iterative classification in relational data. In *Proc. AAAI-2000 Workshop on Learning Statistical Models from Relational Data*, pages 13–20.

Nigam, K., McCallum, A. K., Thrun, S., and Mitchell, T. (2000). Text classification from labeled and unlabeled documents using em. *Machine learning*, 39(2-3):103–134.

Pang, B. and Lee, L. (2004). A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd annual meeting on Association for Computational Linguistics*, page 271. Association for Computational Linguistics.

Perlich, C. and Provost, F. (2003). Aggregation-based feature invention and relational concept classes. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 167–176. ACM.

Rabelo, J. C., Prudêncio, R. B., Barros, F., et al. (2012). Using link structure to infer opinions in social networks. In *Systems, Man, and Cybernetics (SMC), 2012 IEEE International Conference on*, pages 681–685. IEEE.

Tocoglu, M. A. and Alpkocak, A. (2014). Emotion extraction from turkish text. In *Network Intelligence Conference (ENIC), 2014 European*, pages 130–133. IEEE.