An Ensemble Model for Cross-Domain Polarity Classification on Twitter

Conference Paper · October 2014

DOI: 10.1007/978-3-319-11746-1_12

Adam Tsakalidis
Queen Mary, University of London

Symeon Papadopoulos
The Centre for Research and Technology, Hellas

Ioannis (Yiannis) Kompatsiaris
The Centre for Research and Technology, Hellas

An Ensemble Model for Cross-Domain Polarity Classification on Twitter

Adam Tsakalidis, Symeon Papadopoulos, and Ioannis Kompatsiaris

Centre of Research and Technology (CERTH),
57001, Thessaloniki, Greece
{atsak,papadop,ikom}@iti.gr

Abstract. Polarity analysis of Social Media content is of significant importance for various applications. Most current approaches treat this task as a classification problem, demanding a labeled corpus for training purposes. However, if the learned model is applied on a different domain, the performance drops significantly and, given that it is impractical to have labeled corpora for every domain, this becomes a challenging task. In the current work, we address this problem, by proposing an ensemble classifier that is trained on a general domain and and adapts, without the need for additional ground truth, on the desired (test) domain before classifying a document. Our experiments are performed on three different datasets and the obtained results are compared with various baselines and state-of-the-art methods; we demonstrate that our model is outperforming all out-of-domain trained baseline algorithms, and that it is even comparable with different in-domain classifiers.

Keywords: sentiment analysis, polarity detection, ensemble classifier

1 Introduction

Twitter[1] is a microblogging platform that has seen increasing popularity during the latest years. The content produced in this network reflects its users' thoughts on different topics and has proven beneficial for various applications such as modeling public behavior [5], summarisation of events [14] or predicting election results [17]. Sentiment analysis is of particular importance in Twitter: brands can learn what people think of their products, politicians can learn the users' opinions on them, people can aggregate opinions on topics of their interest and so on. Given the huge amounts of broadcasted content, automating the process of opinion mining becomes a rather crucial task for creating real-time insights.

The two mostly studied sentiment analysis tasks are *subjectivity* (given a set of documents, find the subjective ones) and *polarity* detection (discrimination of positive/negative documents). The most common approach to deal with these tasks is to train a classifier on a labeled corpus and apply the learned model on the desired test set. However, accuracy drops significantly when the applied

[1] https://twitter.com/

model is trained on a different type of document, domain or time [12]. In order to overcome this problem, lexicon-based approaches are often employed, using a predefined dictionary of words along with their corresponding prior polarity weight. For a given document, they detect its polarity based on the majority class sum-of-weights of its keywords; however, those fail to perform comparably to in-domain learned models. Furthermore, the task of sentiment analysis becomes more difficult when dealing with short, noisy content and well-known approaches applied in well-formed documents seem ineffective for content of such type.

In the current work, we tackle the problem of the domain-dependent nature of the polarity detection task in Twitter. We train different classifiers on various sets of features and combine them in an ensemble model that achieves an average accuracy boost of 10.22% over our main baseline model (text-based learning) when trained on a different domain. We compare our method with out-of-domain state-of-the-art approaches on public datasets achieving better results; we compare different lexicons and show that our method outperforms them by 12.5% on average; most importantly though, we show that the accuracy of our approach is highly competitive against traditional in-domain training methods, following by only a 3.86% the best in-domain algorithm.

2 Background

Representation Forms: Given a set of documents to classify, the first step is to create a vector space representation of them (usually as n-gram features), often using the $tf \cdot idf$ formula to emphasize characteristic words of a document:

$$tf \cdot idf_{i,j} = \ln(1 + tf_{i,j}) \cdot \ln(|D|/df_i) \tag{1}$$

Here i is a term occurring in document j, tf and df its frequency on j and the total number of documents this term appears in respectively and $|D|$ is the total number of documents in the corpus. Other common preprocessing steps include stop-word removal and stemming; however, these were found to decrease [13] or offer no increase [3] in accuracy in various sentiment analysis tasks.

Part-Of-Speech (POS) tags are also used as features ([13], [2], [19], [11]), along with n-grams, since several tags can reveal the presence of subjectivity in a document or help in word sense disambiguation. Saif et al. [13] demonstrated a boost in accuracy ranging from 0.9% to 8.1% when POS tags were used along with unigrams, whereas a slight decrease was found in Go et al.'s work [8]. The role of several other features has been explored, such as punctuation [2], semantic entities [13] and consecutive letters in a word [6], with results varying.

Learning Methods: Sentiment analysis approaches can be separated into *supervised* and *unsupervised* methods. Supervised approaches require a labeled corpus of documents to learn a model from and apply it to a test set. On the contrary, unsupervised approaches apply a predefined list of rules (usually given by a lexicon) in the test set, overcoming the training step of supervised approaches.

Despite the high accuracy reported by many supervised approaches in microblogs, their algorithms are tested on the same domain that they are trained on. However, one cannot expect to find a labeled corpus for training for all different types of problems. Even worse, classifiers are not only domain-dependent but also topic-, document- and time-dependent ([12], [1]), making it impossible to be applied in real-life problems achieving the same accuracy. This is probably the reason that online sentiment analysis services tend to disagree in their outputs. For example, three different online sentiment analysis services were used in [2], revealing a low kappa statistic ranging from 0.4 to 0.6, whereas the average pair-wise agreement of eight different methods ranged from 48% to 72% in [9].

Using a Naïve Bayes (NB) and Support Vector Machine (SVM), Read revealed that both algorithms perform significantly better when tested on the same dataset that they were trained on, in almost all cases [12], arguing that a more general training set should be constructed. A common approach for this task (e.g. [8], [12]) is to collect a large number of documents (tweets) containing happy/sad emoticons and assign to them the corresponding label (positive/negative), whereas another way to tackle the problem is by applying an unsupervised method.

Most unsupervised methods use a sentiment lexicon, e.g., SentiWordNet [7] (“SWN”; about 150,000 synsets with double values indicating their sentiments), Subjectivity Lexicon [18] or Bing Liu’s Opinion Lexicon [10] (“OL”; about 6,800 words marked as “positive”/“negative”). Compared to in-domain supervised methods, these approaches perform worse, but achieve comparable or better results than out-of-domain supervised algorithms (e.g., a lexicon-based method achieved an average accuracy of 60.6% on sentences compared to 67.9% and 57.2% of an SVM algorithm trained in- and out-of-domain respectively in [1]).

In [9] the authors studied different methods and combined them in a unique system that failed to perform better than their best individual model. A boost of about 1% for a 4-class sentiment task is reported in [18] when keywords are combined with their prior polarity, whereas a gain of about 5% for the polarity task is reported in [11] when lexicon features are used along with content ones using an in-domain classifier. A weighted classifier was developed in [1] that combined a supervised and a lexicon-based approach based on their precision on every class and revealed a significant increase in accuracy of 13.65% on average for the polarity task. However, their approach assumes that every algorithm should perform fairly well on one class and vice versa.

In the current research we try to overcome the domain-dependence problem by creating an ensemble classifier. Instead of learning one model to apply to our test data, we combine different algorithms based on different document representation forms and highlight their role in the polarity detection task.

3 Methodology

Using the Twitter API[2] over a two-day period in mid-March 2014, we gathered 250,000 tweets written in English containing happy/sad emoticons (“:)”, “”:(”;

[2] `https://dev.twitter.com/`

equally balanced), removed all retweets ($7,469$) and used the rest as a training set ("Emoticons Dataset", "ED"). Working on ED, we created four different tweet representations and trained one classifier on each one of them, trying to find the parameters that achieve the highest accuracy [3].

Text-Based Representation (TBR): We used three representations of the tweets using binary, term frequency (tf) and $tf \cdot idf$ n-grams. We set $n = 1$, 2, 3, resulting into nine representations in total. Stop-word removal and stemming were ignored, as suggested by previous works ([13], [3]).

Feature-Based Representation (FBR): We represented every tweet as a set of binary values indicating the presence of several features. These included consecutive dots, exclamation marks, mentions, URLs and negations; hashtags were added by removing the "#" sign; words written in upper-case were lower-cased and added by inserting the word "very" upfront (e.g., "very big" for "BIG"); words containing more than two consecutive letters were also added, by replacing the repetitions with two consecutive ones (e.g., "biig" for "biiig").

Lexicon-Based Representation (LBR): We used two lexicons (SentiWordNet and Opinion Lexicon). Instead of assigning the majority class label on a tweet, we counted the sum of nouns, verbs, adjectives, adverbs and the overall sum of the words as indicated by SWN and the overall sum of words as indicated by OL. In the case of presence of negation, the polarity score of the term that follows was inverted. We use these six features to learn a model from and we compare our results with the simple counting methods of both lexicons.

Combined Representation (CR): We used a combination of TBR with POS tags, using the Stanford POS Tagger [16] and we tested the same parameters as in the case of TBR. Finally, in TBR, FBR and CR, features that appeared only once in the training set were eliminated to achieve noise reduction.

Ensemble Classifier: The main idea behind our ensemble model is to combine the different algorithms' outputs in a weighted scheme in order to classify a tweet. We have separated our classifiers in two categories, based on their domain-(in)dependent nature: the *hybrid* classifier (HC) and the *lexicon-based* one (LC), which is the algorithm that was tested on LBR achieving the highest accuracy. The HC assigns one value per class to a document based on the outputs of the individual (probabilistic) classifiers that are trained on TBR, FBR and CR:

$$hval_c(i) = \sum_r w_r \cdot p_r(i, c). \tag{2}$$

Here i corresponds to the tweet, r to the representation model, w is the model's weight and $p(i, c)$ the probability assigned by the classifier on the respective tweet and class. The weight was set equal to the difference of every classifier's accuracy compared to the random classifier (50%), based on the ED. Finally, the HC assigns the polarity class with the highest *hval* to every tweet.

[3] All features along with the learned models can be found at `https://github.com/socialsensor/sentiment-analysis`

The predictions HC and LC on a test set are combined by the ensemble classifier and the documents for which they agree on are automatically assigned the corresponding label, under the assumption that they are most likely to have been classified correctly. Then, those "agreed" documents are considered as a new training set, whereas the remaining ones of the test set comprise the new test set. At the final stage, a model is learned on the "agreed" documents and applied on the remaining ones. This technique alleviates us from the domain-dependence problem; however, the final training stage is highly dependent on the accuracy of the ensemble classifier achieved on the "agreed" documents.

4 Experimental Study

4.1 Twitter Test Datasets

We used three datasets for testing our approach, focusing on the tweets written in the English language. These datasets will be used in section 5, while the current one will focus on the results based on ED.

Stanford Twitter Dataset Test Set (STS): We have used the non-neutral part of two versions of this dataset [8]; the first one (referred here as "STS-1") consists of 177 positive and 184 negative tweets (see [13]); the second one ("STS-2") consists of 108 positive and 75 negative tweets (see [15]).

Obama Healthcare Reform (HCR): This dataset contains tweets related to the healthcare reform introduced by Barrack Obama in 2010 ([15]). This set is split into three parts. We focused on the positive and negative tweets contained in the "dev" and "test" set separately. In the first one we found 135 positive and 328 negative tweets, whereas in the second one 116 and 383 respectively.

Obama-McCain Debate (OMD): This dataset contains 3,269 annotated tweets related to the 2008 Obama vs McCain debate ([13], [15]). We used the "positive" and "negative" tweets that have been annotated by at least three people for which the annotators' agreement was above 50% on one of our examined classes. This resulted in 1,897 tweets (707 positive and 1,190 negative ones).

4.2 Model Building

Some common pre-processing steps on ED include the replacements of all user mentions with "usrmntn", URLs with "urlink" and hashtags with the actual hashtag by removing the "#" sign. Negations and common abbreviations were transformed into their reference form (e.g., "I've" to "I have", "isn't" to "is not"). We expanded a list of some commonly used abbreviations[4] to transform them into their proper form, removed all emoticons from the training set and replaced all emoticons with their latent meaning in the test sets only (e.g., ":)" to "feeling happy"). Finally, words containing more than two consecutive letters (e.g. "hiiii") were shortened so that they contain only two repetitions ("hii").

[4] http://www.englishclub.com/esl-chat/abbreviations.htm

We used Multinomial Naïve Bayes (MNB) and Support Vector Machines (SVM) on ED, as provided by Weka[5]. We randomized ED and used a 66%/33% split for training/evaluation purposes before working on any test set. We chose MNB to be applied on TBR, FBR and CR, as in previous works ([3], [4]) and SVM for LBR due to their capability of dealing with double-valued attributes.

TBR: Table 1 presents the accuracy for all examined TBR forms. Bigrams outperformed unigrams and trigrams. Following previous claims regarding the appropriate weighting scheme [4], we expected to see some differences as moving from binary to $tf \cdot idf$ forms; however, we find no such differences. This may be because we have not yet moved to another domain, in which case words that appear much more frequently than in our training set could affect results.

FBR: We extracted 29,269 features (mainly hashtags, repetitions and uppercase words) and achieved a relatively low accuracy (61.95%). One explanation of this can be found in the recall (0.84/0.4 for positive/negative class respectively), revealing a bias towards the positive class. Nevertheless, we decided to apply the learned model on our test sets to test the impact of FBR on a different domain.

LBR: Table 2 presents the results obtained by SVM compared to the "counting" methods using both lexicons individually and each POS tag from SWN, revealing the superiority of SVM. Our findings consistently support that adjectives carry more sentimental weight [10]. OL achieved better results than SWN, despite that there are far less words documented in this lexicon and are only marked as "positive" or "negative", whereas every synset is carrying a double-valued polarity weight in SWN. What is important from the results presented here though is that a learning algorithm over some lexicon features can boost traditional lexicon-based approaches by an average of 3.9% in accuracy.

CR: The results on the ED support previous findings ([13], [8]) on the use of POS tags along with unigrams, revealing an average boost of 4.73% in accuracy across different weighting models (see Table 1). However, we notice a slight decrease on bigrams and trigrams, most likely because the sparsity of these representations increases along with the increase of the "n" value in n-grams much faster than in TBR, resulting into information loss. This is explained by the number of extracted features: there exist about 33% more features in the case of unigrams for CR than for TBR (28,491 vs 37,796), while there are fewer trigrams (150,398 vs 145,178), because of the tf threshold we applied.

Table 1. Accuracies achieved on the 33% of the ED for TBR and CR.

Represent.	**binary**			**tf**			**tfidf**		
n-gram	1	2	3	1	2	3	1	2	3
TBR	74.59	**81.07**	78.07	74.62	80.92	78.07	73.46	81.02	78.88
CR	79.64	80.86	77.42	79.61	80.62	77.15	77.60	**80.95**	78.13

[5] http://www.cs.waikato.ac.nz/ml/weka/

Table 2. Accuracy using lexicon counting methods and an SVM applied on LBR.

	Adv	**Noun**	**Vrb**	**Adj**	**SWN**	**OL**	**LC**
Acc.(%)	49.86	51.23	54.85	57.41	59.88	62.44	65.06

Table 3. Accuracies of all classifiers in DEV. LBR and LC refer to the same classifier.

Method	**TBR**	**FBR**	**LBR**	**CR**	**HC**	**LC**	**Agreed**	**Disagr.**	**All**
Acc.(%)	82.16	65.73	79.96	80.56	82.54	79.96	91.13	69.88	85.72

Ensemble Classifier: We created a new development set ("DEV") of $10,000$ equally-balanced tweets aggregated from Twitter in the same way that the ED was created; we removed all retweets and used the rest for tuning the parameters of the ensemble classifier, while using the full ED for training. The parameters for HC in Equation 2 were set to 31.07, 11.95 and 30.95 for TBR, FBR and CR respectively, as indicated by the best achieved accuracy of every model on ED. Table 3 summarizes the results achieved by all individual classifiers on the DEV. The results are similar to the ones achieved in the ED set, with the exception of LBR, which achieved significantly better results in the DEV because of the replacement of emoticons with their latent meaning.

The column "HC" shows the accuracy of the hybrid classifier, whereas the "LC" is copied from LBR for clarity. HC achieved slightly better results, compared to its individual components. In total the two classifiers agreed on 74.55% of the DEV set and they achieve a very high accuracy on these "agreed" tweets, revealing that these can serve as a slightly noisy new training set. In the final stage, we trained a MNB using TBR based on the "agreed" tweets, since this representation achieved the best results in both ED and DEV sets. We notice that the accuracy falls down to 69.88% on the remaining 25.45% "disagreed" tweets, yielding an overall 85.72%. This reduction is because 8.87% of the training set was wrongly classified upfront and the learned model was partially based on these tweets. Nevertheless, our approach managed to outperform the best individual algorithm (MNB on TBR) by a 3.56%; this difference may be greater when we apply our model to a specific domain, in which case TBRs usually lead to poor results.

5 Results

We compare our results with several baselines and state-of-the-art approaches: the majority class classifier (MC); the four methods that comprise our ensemble classifier; three different lexicon (counting) approaches; and the results obtained by our four models (TBR, FBR, LBR, CR) with 10-fold cross validation. All "averages" presented here are calculated by first averaging the accuracies on every test set (e.g., the average of HCR-dev and HCR-test) and then calculating the overall average.

On average, our model outperformed all out-of-domain models presented in Table 4. The average boost in accuracy is 11.65% compared to the MC, 10.22%

Table 4. Comparison of results. The "training" column refers to whether the training was based in the same domain with the test set or not. Results copied from other words are cited. The best in- and out-of-domain classifier is highlighted on every dataset.

Feature	**Training**	**STS-1**	**STS-2**	**HCR-dev**	**HCR-test**	**OMD**	**Average**
Majority Class	Out	50.14	58.33	**70.75**	**76.65**	62.72	63.55
TBR	Out	77.52	76.40	52.70	50.80	62.52	63.74
FBR	Out	60.52	63.48	49.68	52.81	62.84	58.70
LBR	Out	77.52	76.40	68.68	72.69	70.53	72.73
CR	Out	75.50	71.35	58.10	55.22	64.84	64.98
SentiWordNet	Out	49.86	73.60	51.19	52.10	56.19	56.52
Opinion Lexicon	Out	54.18	76.97	68.90	72.14	70.90	69.00
Subj. Lexicon [15]	Out	-	72.10	54.30	58.10	59.10	62.47
TBR (10-fold)	In	83.29	74.16	**77.75**	**80.36**	79.39	**79.06**
FBR (10-fold)	In	60.23	61.80	76.03	75.15	63.42	66.68
LBR (10-fold)	In	78.10	**83.71**	71.71	79.36	71.32	75.92
CR (10-fold)	In	81.27	70.22	76.46	75.75	**80.92**	77.59
Ensemble	Out	**83.57**	80.90	69.11	69.68	**73.96**	**75.20**

compared to the CR trained on ED and 6.2% compared to the best lexicon. Moreover, it manages to compete even with various in-domain models, achieving only 3.86% lower accuracy than the best such model. OL is the only lexicon outperforming the MC, while SWN performs poorly. This is probably due to the informal nature of tweets, in which the most common words usually appear and a simplistic approach can achieve better results. Integrating POS tags to TBR provides a small boost, only when trained in a different domain. The LBR model is rather competitive with TBR and CR for in-domain tasks, whereas it confidently outperforms them for cross-domain problems. while the FBR model fails to perform equally well with the other models.

In order to explain our classifier's results, we move on analysing its componenents. HC and LC agree in more than half of the tweets on average (see Table 5, "agreement"), on which the average accuracy is high (81.81%). The higher the agreement level (%), the higher the accuracy that we achieve (correlation = 0.997); if such a claim holds universally, then one could argue about whether the ensemble model could be used effectively or not based on the agreement level. However, with only three datasets in use one cannot draw safe conclusions.

We used the "agreed" tweets to train a MNB classifier on TBR. On average, 55.15% of the originally considered as test data was used for training purposes of our ensemble classifier, whereas the rest 44.85% was used for testing. This means that our ensemble classifier learns a model on a training set in which 19.19% of the tweets are wrongly labeled. However, the same argument may apply on the ED set as well, since a tweet containing a happy emoticon may be ironic instead of positive. As expected, the accuracy is lower in the "disagreed" tweets (65.82%) due to the noisy training set.

Our results indicate that in the HCR set the MC achieves higher accuracy than all out-of-domain methods, probably due to its high imbalance. The HCR-

Table 5. Agreement level and results obtained on the test set by the ensemble classifier.

	STS-1	STS-2	HCR-dev	HCR-test	OMD	Average
Agreement(%)	70.03	66.29	52.92	51.00	55.98	55.15
Accuracy (agreed)	88.07	86.44	75.10	77.95	81.64	81.81
Accuracy (disagreed)	73.08	70.00	62.39	61.07	64.19	65.82

test set is also the only one in which an individual classifier outperforms our ensemble model. This dataset is the one with the lowest agreement level between HC and LC). Taking into account that this agreement level correlates with the accuracy on the agreed tweets, there might exist a threshold on the agreed documents, below which the classifier's algorithm should be switched to another model, instead of text-based; we leave this as an open problem for the future.

6 Conclusion

This paper proposed an ensemble algorithm to deal with the domain-dependence problem for the polarity classification task on Twitter. The basic idea is to automatically categorize some tweets of a given domain-specific test set and use them as a new train set. Our results show that combining algorithms trained on different features on a generic train set can prove beneficial, achieving high accuracy (81.81%) on the resulting train set. Using this new dataset for training, we achieve better results than all other out-of-domain approaches tested here, but also to compete against widely-applied in-domain learning methods.

Future work includes the incorporation of the "neutral" class in our problem as well as enhancing syntactic rules in our approach [11]. Finally, we plan to test our approach on different datasets for wider justification and test whether the number of documents for which our two combined classifiers agree on is in fact correlated with the accuracy that will be achieved on the whole test set.

Acknowledgments. This work is supported by the SocialSensor FP7 project, partially funded by the EC under contract number 287975.

References

1. Alina Andreevskaia and Sabine Bergler. When specialists and generalists work together: Overcoming domain dependence in sentiment tagging. In *ACL*, pages 290–298, 2008.
2. Luciano Barbosa and Junlan Feng. Robust sentiment detection on twitter from biased and noisy data. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 36–44. Association for Computational Linguistics, 2010.
3. Adam Bermingham and Alan F Smeaton. Classifying sentiment in microblogs: is brevity an advantage? In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 1833–1836. ACM, 2010.

4. Albert Bifet and Eibe Frank. Sentiment knowledge discovery in twitter streaming data. In *Discovery Science*, pages 1–15. Springer, 2010.
5. Johan Bollen, Huina Mao, and Alberto Pepe. Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena. In *ICWSM*, 2011.
6. Samuel Brody and Nicholas Diakopoulos. Cooooooooooooooollllllllllllll!!!!!!!!!!!!!!: using word lengthening to detect sentiment in microblogs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 562–570. Association for Computational Linguistics, 2011.
7. Andrea Esuli and Fabrizio Sebastiani. Sentiwordnet: A publicly available lexical resource for opinion mining. In *Proceedings of LREC*, volume 6, pages 417–422, 2006.
8. Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, pages 1–12, 2009.
9. Pollyanna Gonçalves, Matheus Araújo, Fabrício Benevenuto, and Meeyoung Cha. Comparing and combining sentiment analysis methods. In *Proceedings of the first ACM conference on Online social networks*, pages 27–38. ACM, 2013.
10. Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177. ACM, 2004.
11. Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. Target-dependent twitter sentiment classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 151–160. Association for Computational Linguistics, 2011.
12. Jonathon Read. Using emoticons to reduce dependency in machine learning techniques for sentiment classification. In *Proceedings of the ACL Student Research Workshop*, pages 43–48. Association for Computational Linguistics, 2005.
13. Hassan Saif, Yulan He, and Harith Alani. Semantic sentiment analysis of twitter. In *The Semantic Web–ISWC 2012*, pages 508–524. Springer, 2012.
14. Emmanouil Schinas, Symeon Papadopoulos, Sotiris Diplaris, Yiannis Kompatsiaris, Yosi Mass, Jonathan Herzig, and Lazaros Boudakidis. Eventsense: Capturing the pulse of large-scale events by mining social media streams. In *Proceedings of the 17th Panhellenic Conference on Informatics*, pages 17–24. ACM, 2013.
15. Michael Speriosu, Nikita Sudan, Sid Upadhyay, and Jason Baldridge. Twitter polarity classification with label propagation over lexical links and the follower graph. In *Proceedings of the First workshop on Unsupervised Learning in NLP*, pages 53–63. Association for Computational Linguistics, 2011.
16. Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 173–180. Association for Computational Linguistics, 2003.
17. Andranik Tumasjan, Timm Oliver Sprenger, Philipp G Sandner, and Isabell M Welpe. Predicting elections with twitter: What 140 characters reveal about political sentiment. *ICWSM*, 10:178–185, 2010.
18. Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*, pages 347–354. Association for Computational Linguistics, 2005.
19. Ley Zhang, Riddhiman Ghosh, Mohamed Dekhil, Meichun Hsu, and Bing Liu. Combining lexiconbased and learning-based methods for twitter sentiment analysis. *HP Laboratories, Technical Report HPL-2011*, 89, 2011.