Biocom_Usp: Tweet Sentiment Analysis with Adaptive Boosting Ensemble

Nádia F. F. Silva, Eduardo R. Hruschka
University of São Paulo, USP
São Carlos, SP, Brazil
nadia, erh@icmc.usp.br

Estevam Rafael Hruschka Jr.
Department of Computer Science
Federal University of Sao Carlos.
São Carlos, SP, Brazil
estevam@dc.ufscar.br

Abstract

We describe our approach for the *SemEval-2014 task 9: Sentiment Analysis in Twitter*. We make use of an ensemble learning method for sentiment classification of tweets that relies on varied features such as feature hashing, part-of-speech, and lexical features. Our system was evaluated in the Twitter message-level task.

1 Introduction

The sentiment analysis is a field of study that investigates feelings present in texts. This field of study has become important, especially due to the internet growth, the content generated by its users, and the emergence of the social networks. In the social networks such as Twitter people post their opinions in a colloquial and compact language, and it is becoming a large dataset, which can be used as a source of information for various automatic tools of sentiment inference. There is an enormous interest in sentiment analysis of Twitter messages, known as *tweets*, with applications in several segments, such as (i) directing *marketing* campaigns, extracting consumer reviews of services and products (Jansen et al., 2009); (ii) identifying manifestations of *bullying* (Xu et al., 2012); (iii) predicting to forecast box-office revenues for movies (Asur and Huberman, 2010); and (iv) predicting acceptance or rejection of presidential candidates (Diakopoulos and Shamma, 2010; O'Connor et al., 2010).

One of the problems encountered by researchers in tweet sentiment analysis is the scarcity of *public datasets*. Although Twitter sentiment datasets have already been created, they are either small — such as Obama-McCain Debate corpus (Shamma et al., 2009) and Health Care Reform corpus (Speriosu et al., 2011) or big and proprietary such as in (Lin and Kolcz, 2012). Others rely on noisy labels obtained from emoticons and hashtags (Go et al., 2009). The *SemEval-2014 task 9: Sentiment Analysis in Twitter* (Nakov et al., 2013) provides a public dataset to be used to compare the accuracy of different approaches.

In this paper, we propose to analyse tweet sentiment with the use of Adaptive Boosting (Freund and Schapire, 1997), making use of the well-known Multinomial Classifier. Boosting is an approach to machine learning that is based on the idea of creating a highly accurate prediction rule by combining many relatively weak and inaccurate rules. The AdaBoost algorithm (Freund and Schapire, 1997) was the first practical boosting algorithm, and remains one of the most widely used and studied, with applications in numerous fields. Therefore, it has potential to be very useful for tweet sentiment analysis, as we address in this paper.

2 Related Work

Classifier ensembles for tweet sentiment analysis have been underexplored in the literature — a few exceptions are (Lin and Kolcz, 2012; Clark and Wicentwoski, 2013; Rodriguez et al., 2013; Hassan et al., 2013).

Lin and Kolcz (2012) used logistic regression classifiers learned from hashed byte 4-grams as features – The feature extractor considers the tweet as a raw byte array. It moves a four-byte sliding window along the array,

Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pages 129–134,
Dublin, Ireland, August 23-24, 2014.

and hashes the contents of the bytes, the value of which was taken as the feature id. Here the 4-grams refers to four characters (and not to four words). They made no attempt to perform any linguistic processing, not even word tokenization. For each of the (proprietary) datasets, they experimented with ensembles of different sizes. The ensembles were formed by different models, obtained from different training sets, but with the same learning algorithm (logistic regression). Their results show that the ensembles lead to more accurate classifiers.

Rodrígues et al. (2013) and Clark et al. (2013) proposed the use of classifier ensembles at the expression-level, which is related to *Contextual Polarity Disambiguation*. In this perspective, the sentiment label (positive, negative, or neutral) is applied to a specific phrase or word within the tweet and does not necessarily match the sentiment of the entire tweet.

Finally, another type of ensemble framework has been recently proposed by Hassan et al. (2013), who deal with class imbalance, sparsity, and representational issues. The authors propose to enrich the corpus using multiple additional datasets related to the task of sentiment classification. Differently from previous works, the authors use a combination of unigrams and bigrams of simple words, part-of-speech, and semantic features.

None of the previous works used AdaBoost (Freund and Schapire, 1996). Also, lexicons and/or part-of-speech in combination with feature hashing, like in (Lin and Kolcz, 2012) have not been addressed in the literature.

3 AdaBoost Ensemble

Boosting is a relatively young, yet extremely powerful, machine learning technique. The main idea behind boosting algorithms is to combine multiple weak learners – classification algorithms that perform only slightly better than random guessing – into a powerful composite classifier. Our focus is on the well known AdaBoost algorithm (Freund and Schapire, 1997) based on Multinomial Naive Bayes as base classifiers (Figure 1).

AdaBoost and its variants have been applied to diverse domains with great success, owing to their solid theoretical foundation, accurate prediction, and great simplicity (Freund and Schapire, 1997). For example, Viola and Jones (2001) used AdaBoost to face detection, Hao and Luo (2006) dealt with image segmentation, recognition of handwritten digits, and outdoor scene classification problems. In (Bloehdorn and Hotho, 2004) text classification is explored.

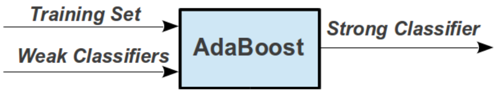

Figure 1: AdaBoost Approach

4 Feature Engineering

The most commonly used text representation method adopted in the literature is known as Bag of Words (BOW) technique, where a document is considered as a BOW, and is represented by a feature vector containing all the words appearing in the corpus. In spite of BOW being simple and very effective in text classification, a large amount of information from the original document is not considered, word order is ruptured, and syntactic structures are broken. Therefore, sophisticated feature extraction methods with a deeper understanding of the documents are required for sentiment classification tasks. Instead of using only BOW, alternative ways to represent text, including Part of Speech (PoS) based features, feature hashing, and lexicons have been addressed in the literature.

We implemented an ensemble of classifiers that receive as input data a combination of three features sets: i) *lexicon features* that captures the semantic aspect of a tweet; ii) *feature hashing* that captures the surface-form as abbreviations, slang terms from this type of social network, elongated words (for example, *loveeeee*), sentences with words without a space between them (for instance, *Ilovveapple!*), and so on; iii) and a *specific syntactic features* for tweets. Technical details of each feature set are provided in the sequel.

Lexicon Features

We use the sentimental lexicon provided by (Thelwall et al., 2010) and (Hu and Liu, 2004). The former is known as SentiStrength and provides: an emotion vocabulary, an emoticons list (with positive, negative, and neutral icons), a negation list, and a booster word list. We use the negative list in cases where the next term in a sentence is an opinion word (either positive or negative). In such cases we have polarity inversion. For example, in the sentence "The house is *not beautiful*", the negative word "*not*" invert the polarity of the opinion word *beautiful*. The booster word list is composed by adverbs that suggest more or less emphasis in the sentiment. For example, in the sentence "He was *incredibly rude*." the term "*incredibly*" is an adverb that lay emphasis on the opinion word "*rude*". Besides using SentiStrength, we use the lexicon approach proposed by (Hu and Liu, 2004). In their approach, a list of words and associations with positive and negative sentiments has been provided that are very useful for sentiment analysis.

These two lexicons were used to build the first feature set according to Table 1, where it is presented an example of tweet representation for the $tweet_1$: "The soccer team didn't play extremely bad last Wednesday." The word "bad" exists in the lexicon list of (Hu and Liu, 2004), and it is a negative word. The word "bad" also exists in the negation list provided by (Thelwall et al., 2010). The term "didn't" is a negative word according to SentiStrength (Thelwall et al., 2010) and there is a polarity inversion of the opinion words ahead. Finally, the term "extremely" belongs the booster word list and this word suggests more emphasis to the opinion words existing ahead.

	positive	negative	neutral	class
$tweet_1$	3	0	0	positive

Table 1: Representing Twitter messages with lexicons.

Feature hashing

Feature hashing has been introduced for text classification in (Shi et al., 2009), (Weinberger et al., 2009), (Forman and Kirshenbaum, 2008), (Langford et al., 2007), (Caragea et al., 2011). In the context of tweet classification, feature hashing offers an approach to reducing the number of features provided as input to a learning algorithm. The original high-dimensional space is "reduced" by *hashing* the features into a lower-dimensional space, i.e., mapping features to hash keys. Thus, multiple features can be mapped to the same hash key, thereby "aggregating" their counts.

We used the MurmurHash3 function (SMHasher, 2010), that is a non-cryptographic hash function suitable for general hash-based lookup tables. It has been used for many purposes, and a recent approach that has emerged is its use for feature hashing or hashing trick. Instead of building and storing an explicit traditional bag-of-words with n-grams, the feature hashing uses a hash function to reduce the dimensionality of the output space and the length of this space (features) is explicitly fixed in advance. For this paper, we used this code (in Python):

Code Listing 1: Murmurhash:

```
from sklearn.utils.murmurhash
import murmurhash3_bytes_u32

for w in "i loveee apple".split():
  print("{0} => {1}".format(
    w,murmurhash3_bytes_u32(w,0)%2**10))
```

The dimensionality is $2 * *10$, i.e 2^{10} features. In this code the output is a hash code for each word "w" in the phrase "i loveee apple", i.e. $i \ => 43$, $loveee \ => 381$ and $apple \ => 144$. Table 2 shows an example of feature hashing representation.

	1	2	3	4	···	1024	class
$tweet_1$	0	0	1	1	···	0	positive
$tweet_2$	0	1	0	3	···	0	negative
$tweet_3$	2	0	0	0	···	0	positive
⋮	⋮	⋮	⋮	⋮	...	⋮	⋮
$tweet_n$	0	0	2	1	···	0	neutral

Table 2: Representing Twitter messages with feature hashing.

Specific syntactic (PoS) features

We used the Part of Speech (PoS) tagged for tweets with the Twitter NLP tool (Gimpel et al., 2011). It encompasses 25 tags including Nominal, Nominal plus Verbal, Other open-class words like adjectives, adverbs and interjection, Twitter specific tags such as hashtags, mention, discourse marker, just to name a few. Table 3 shows an example of syntactic features representation.

	tag_1	tag_2	tag_3	tag_4	$\cdots$	tag_{25}	class
$tweet_1$	0	0	3	1	$\cdots$	0	positive
$tweet_2$	0	2	0	1	$\cdots$	0	negative
$tweet_3$	1	0	0	0	$\cdots$	0	positive
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ldots$	$\vdots$	$\vdots$
$tweet_n$	0	0	1	1	$\cdots$	0	neutral

Table 3: Representing Twitter messages with syntactic features.

A combination of lexicons, feature hashing, and part-of-speech is used to train the ensemble classifiers, thereby resulting in 1024 features from feature hashing, 3 features from lexicons, and 25 features from PoS.

5 Experimental Setup and Results

We conducted experiments by using the WEKA platform[1]. Table 4 shows the class distributions in training, development, and testing sets. Table 5 presents the results for positive and negative classes with the classifiers used in training set, and Table 6 shows the computed results by SemEval organizers in the test sets.

Training Set				
Set	Positive	Negative	Neutral	Total
Train	3,640 (37%)	1,458 (15%)	4,586 (48%)	9,684
Development Set				
Set	Positive	Negative	Neutral	Total
Dev	575 (35%)	340(20%)	739 (45%)	1,654
Testing Sets				
Set	Positive	Negative	Neutral	Total
LiveJournal	427 (37%)	304 (27%)	411 (36%)	1,142
SMS2013	492 (23%)	394(19%)	1,207 (58%)	2,093
Twitter2013	1,572 (41%)	601 (16%)	1,640 (43%)	3,813
Twitter2014	982 (53%)	202 (11%)	669 (36%)	1,853
Twitter2014Sar	33 (38%)	40 (47%)	13 (15%)	86

Table 4: Class distributions in the training set (Train), development set (Dev) and testing set (Test).

6 Concluding Remarks

From our results, we conclude that the use of AdaBoost provides good performance in the sentiment analysis (message-level subtask). In the cross-validation process, Multinomial Naive Bayes (MNB) has shown better results than Support Vector Machines (SVM) as a component for AdaBoost. However, we feel that further investigations are necessary before making strong claims about this result.

Set	Algorithm	F-Measure Positive	F-Measure Negative	Average
Train	MNB	63.40	49.40	56.40
Train	SVM	64.00	44.50	54.20
Train	AdaBoost w/ SVM	62.50	44.50	53.50
Train	**AdaBoost w/ MNB**	**65.10**	**49.60**	**57.35**

Table 5: Results from 10-fold cross validation in the training set with default parameters of Weka. MNB and SVM stand for Multinomial Naive Bayes and Support Vector Machine, respectively.

Scoring LiveJournal2014			
class	precision	recall	F-measure
positive	69.79	64.92	67.27
negative	76.64	61.64	68.33
neutral	51.82	69.84	59.50
overall score : 67.80			
Scoring SMS2013			
positive	61.99	46.78	53.32
negative	72.34	42.86	53.82
neutral	53.85	83.76	65.56
overall score : 53.57			
Scoring Twitter2013			
positive	68.07	66.13	67.08
negative	48.09	50.00	49.02
neutral	67.20	68.15	67.67
overall score : 58.05			
Scoring Twitter2014			
positive	65.17	70.48	67.72
negative	53.47	48.21	50.70
neutral	59.94	55.62	57.70
overall score : 59.21			
Scoring Twitter2014Sarcasm			
positive	63.64	44.68	52.50
negative	22.50	75.00,	34.62
neutral	76.92	37.04	50.00
overall score : 43.56			

Table 6: Results in the test sets — AdaBoost plus Multinomial Naive Bayes, which was the best algorithm in cross validation.

Overall, the SemEval Tasks have make evident the usual challenges when mining opinions from Social Media channels: noisy text, irregular grammar and orthography, highly specific lingo, and others. Moreover, temporal dependencies can affect the performance if the training and test data have been gathered at different.

Acknowledgements

The authors would like to thank the Research Agencies CAPES, FAPESP, and CNPq for their financial support.

References

Sitaram Asur and Bernardo A. Huberman. 2010. Predicting the future with social media. In *Proceedings of the 2010 International Conference on*

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

Web Intelligence and Intelligent Agent Technology - Volume 01, WI-IAT '10, pages 492–499, Washington, DC, USA. IEEE Computer Society.

Stephan Bloehdorn and Andreas Hotho. 2004. Text classification by boosting weak learners based on terms and concepts. In *Proceedings of the Fourth IEEE International Conference on Data Mining*, pages 331–334. IEEE Computer Society Press, November.

Cornelia Caragea, Adrian Silvescu, and Prasenjit Mitra. 2011. Protein sequence classification using feature hashing. In Fang-Xiang Wu, Mohammed Javeed Zaki, Shinichi Morishita, Yi Pan, Stephen Wong, Anastasia Christianson, and Xiaohua Hu, editors, *BIBM*, pages 538–543. IEEE.

Sam Clark and Rich Wicentwoski. 2013. Swatcs: Combining simple classifiers with estimated accuracy. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 425–429, Atlanta, Georgia, USA, June.

Nicholas A. Diakopoulos and David A. Shamma. 2010. Characterizing debate performance via aggregated twitter sentiment. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 1195–1198, New York, NY, USA. ACM.

George Forman and Evan Kirshenbaum. 2008. Extremely fast text feature extraction for classification and indexing. In *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management*, pages 1221–1230, New York, NY, USA. ACM.

Yoav Freund and Robert E. Schapire. 1996. Experiments with a new boosting algorithm. In *Thirteenth International Conference on Machine Learning*, pages 148–156, San Francisco. Morgan Kaufmann.

Yoav Freund and Robert E Schapire. 1997. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119 – 139.

Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A. Smith. 2011. Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics – Short Papers - Volume 2*, HLT '11, pages 42–47, Stroudsburg, PA, USA.

Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. *Processing*, pages 1–6.

Wei Hao and Jiebo Luo. 2006. Generalized Multiclass AdaBoost and Its Applications to Multimedia Classification. In *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW &#039;06. Conference on*, page 113, Washington, DC, USA, June. IEEE.

Ammar Hassan, Ahmed Abbasi, and Daniel Zeng. 2013. Twitter sentiment analysis: A bootstrap ensemble framework. In *SocialCom*, pages 357–364. IEEE.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '04, pages 168–177, New York, NY, USA. ACM.

Bernard J. Jansen, Mimi Zhang, Kate Sobel, and Abdur Chowdury. 2009. Twitter power: Tweets as electronic word of mouth. *J. Am. Soc. Inf. Sci. Technol.*, 60(11):2169–2188, nov.

John Langford, Alex Strehl, and Lihong Li. 2007. Vowpal wabbit online learning project. `http://mloss.org/software/view/53/`.

Jimmy Lin and Alek Kolcz. 2012. Large-scale machine learning at twitter. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, SIGMOD '12, pages 793–804, New York, NY, USA. ACM.

Preslav Nakov, Sara Rosenthal, Zornitsa Kozareva, Veselin Stoyanov, Alan Ritter, and Theresa Wilson. 2013. Semeval-2013 task 2: Sentiment analysis in twitter. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 312–320, Atlanta, Georgia, USA, June.

Brendan O'Connor, Ramnath Balasubramanyan, Bryan R. Routledge, and Noah A. Smith. 2010. From tweets to polls: Linking text sentiment to public opinion time series. In *ICWSM'10*, pages 1–1.

Penagos Carlos Rodriguez, Jordi Atserias, Joan Codina-Filba, David Garcıa-Narbona, Jens Grivolla, Patrik Lambert, and Roser Saurı. 2013. Fbm: Combining lexicon-based ml and heuristics for social media polarities. In *Proceedings of SemEval-2013 – International Workshop on Semantic Evaluation Co-located with *Sem and NAACL*, Atlanta, Georgia. Url date at 2013-10-10.

David A. Shamma, Lyndon Kennedy, and Elizabeth F. Churchill. 2009. Tweet the debates: Understanding community annotation of uncollected sources. In *In WSM ?09: Proceedings of the international workshop on Workshop on Social*.

Qinfeng Shi, James Petterson, Gideon Dror, John Langford, Alex Smola, and S.V.N. Vishwanathan. 2009. Hash kernels for structured data. *J. Mach. Learn. Res.*, 10:2615–2637.

SMHasher. 2010. The murmurhash family of hash functions.

Michael Speriosu, Nikita Sudan, Sid Upadhyay, and Jason Baldridge. 2011. 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, Stroudsburg, PA, USA.

Mike Thelwall, Kevan Buckley, Georgios Paltoglou, Di Cai, and Arvid Kappas. 2010. Sentiment in short strength detection informal text. *J. Am. Soc. Inf. Sci. Technol.*, 61(12):2544–2558, December.

Paul Viola and Michael Jones. 2001. Robust real-time object detection. In *International Journal of Computer Vision*.

Kilian Q. Weinberger, Anirban Dasgupta, John Langford, Alexander J. Smola, and Josh Attenberg. 2009. Feature hashing for large scale multitask learning. In Andrea Pohoreckyj Danyluk, L Bottou, and Michael L. Littman, editors, *ICML*, volume 382 of *ACM International Conference Proceeding Series*, page 140. ACM.

Jun-Ming Xu, Kwang-Sung Jun, Xiaojin Zhu, and Amy Bellmore. 2012. Learning from bullying traces in social media. In *HLT-NAACL*, pages 656–666.