Polarity Detection of Foursquare Tips

Felipe Moraes, Marisa Vasconcelos, Patrick Prado, Daniel Dalip,
Jussara Almeida, and Marcos Gonçalves

Universidade Federal de Minas Gerais, Brazil
{felipemoraes,marisav,patrickprado,hasan,jussara,mgolcalves}@dcc.ufmg.br

Abstract. In location-based social networks, such as Foursquare, users may post tips with their opinions about visited places. Tips may directly impact the behavior of future visitors, providing valuable feedback to business owners. Sentiment or polarity detection has attracted great attention due to its vast applicability in opinion summarization, ranking or recommendation. However, the automatic detection of polarity of tips faces challenges due to their short sizes and informal content. This paper presents an empirical study of supervised and unsupervised techniques to detect the polarity of Foursquare tips. We evaluate the effectiveness of four methods on two sets of tips, finding that a simpler lexicon-based approach, which does not require costly manual labeling, can be as effective as state-of-the-art supervised methods. We also find that a hybrid approach that combines all considered methods by means of stacking does not significantly outperform the best individual method.

Keywords: Web 2.0 applications, Sentiment Analysis, Micro-reviews

1 Introduction

The widespread use of *smartphones* with geolocation technologies like GPS (*Global Positioning System*) and the increasing interest in social networks led to the appearance of location-based social networks (LBSNs), such as Foursquare, as well as the use of geolocation services by other networks like Google Plus and Instagram.[1] On Foursquare, the currently most popular LBSN, users may share not only their locations, by checking in at venues, but also their opinions about those places, by writing micro-reviews or tips.

Tips are short and informal texts containing opinions about any aspect related to the target venue. For instance, a tip left at a restaurant may contain a recommendation or a complaint about a specific dish or service offered. Moreover, users may *like* a tip in sign of agreement with or interest in its content. Thus, tips and likes foster interactions among users, and provide valuable feedback to business owners to improve the quality of their products and services.

This paper analyzes methods for polarity or sentiment detection of Foursquare tips. In general terms, polarity detection aims at determining the attitude of a

[1] http://foursquare.com/, http://plus.google.com/ and https://instagram.com/

speaker (or writer) towards some topic, classifying it as, for example, positive or negative. Automatic polarity detection has several applications, including opinion summarization in online reviews [1] and real time monitoring of people's opinions [2]. The polarity detection of Foursquare tips could be used to summarize the sentiment of users towards a specific place (venue), providing quick feedback to venue owners from their potential customers, and assisting other users when choosing places to visit. However, tips have inherent characteristics that bring challenges to polarity detection: they are typically very short (limited to 200 characters), contain very informal content and often slangs and expressions (e.g.,"cooool!!"), which are hard to analyze and make their polarity unclear.

Existing polarity detection techniques are grouped into *supervised* and *unsupervised* methods. In the former, automatic classifiers are learned from previously labeled examples [1],[3–7], whereas the latter often relies on lists of positive and negative words (lexicons), using the polarity of each term to classify a text [3],[4],[6],[7]. Although supervised methods have been effectively used for polarity detection in "traditional" environments (e.g., long texts) [8], their efficacy strongly depends on the availability of reliable labeled examples for training. As these examples are often labeled by people, they are subject to errors caused by misinterpretation of the text. The number of training examples also affects classification accuracy, as ideally they should cover as many scenarios as possible [9], which implies high costs for building training sets. In contrast, lexicon-based methods do not require any training and can be applied in various contexts and applications. However, there are not many lexicons suitable to all application domains, as the sentiment of some terms may depend on the topic domain [9]. Moreover, there is no consensus as to which of the two approaches behaves best in short texts like tips [4],[7],[6].

This paper analyzes alternative techniques for automatic polarity detection of Foursquare tips, namely three supervised classifiers - Naïve Bayes, Maximum Entropy and Support Vector Machine [10], and one unsupervised method based on the SentiWordNet lexicon [11]. Our study is based on two sets of tips: one was manually labeled (positive or negative) by volunteers, while the polarity of tips in the other was inferred from emoticons. Our experimental results show that the unsupervised approach produces average Macro-F1 results that are statistically tied to those of the best supervised method - Naïve Bayes - in both datasets, without the cost of labeling. We also find that the unsupervised method is at least as good as, if not better than, Naïve Bayes to detect positive tips, particularly in terms of F1 and recall, whereas the latter produces slightly better results for negative tips. Finally, we also evaluated a hybrid classifier that combines all techniques using stacking, finding that it leads to no further gains over the best individual method, possibly due to the large agreement among the techniques.

2 Related Work

Existing polarity detection solutions can be grouped into supervised machine learning based methods and unsupervised lexicon based methods. Regardless

of the technique employed, most previous studies target long texts, often long reviews. For example, Pang *et al.* [1] evaluated three supervised classification algorithms - Naïve Bayes, Maximum Entropy and SVM - in detecting the polarity of movie reviews, representing each review as a bag-of-words based on unigrams and bigrams. Ohana *et al.* [3], instead, proposed two approaches to use the SentiWordNet lexicon for the same task: (1) using the sum of scores of positive and negative words in the text and taking the highest score as the polarity of the text, and (2) using the scores as features to train an SVM classifier. We here consider an unsupervised method similar to the first approach, representing each tip as a bag-of-words with TF-IDF (product of the term frequency by the inverse document frequency) weights. We also evaluate a hybrid approach that combines supervised and unsupervised methods, similarly to the aforementioned second approach [3], but that goes beyond by combining the predictions of the supervised methods along with the scores of the lexicon for the polarity detection.

Recently, the polarity detection of short texts has attracted attention, with focus mainly on Twitter [5],[4],[6]. In [5], the authors used features extracted from the textual content and related to the social context of the tweet's author (e.g., polarity of the followers' messages) to detect its polarity. They also exploited the presence of positive or negative emoticons to determine the polarity of a tweet, as we do here to build one of our datasets of tips.

We are aware of only two previous studies of polarity of Foursquare tips [12],[13]. In [12], the authors proposed a lexicon-based method that relies on SentiWordNet to build classifiers, and evaluated them using a dataset collected from both Yelp and Foursquare. Although the datasets are different - in particular Yelp reviews tend to be much longer than Foursquare tips - the results reported in [12] are worse than those obtained with our unsupervised method, perhaps due to the way they estimated polarity scores. The other previous effort also used SentiWordNet to detect the polarity of Foursquare tips, with the goal of building a location recommendation method [13]. Thus, they did not evaluate the effectiveness of the detection methods considered, which is our goal.

A few studies compared supervised and unsupervised approaches to classify the polarity of short texts. In [4], the authors showed that the supervised methods analyzed in [1] may be applied to short tweets, comparing their accuracy with that of an unsupervised lexicon-based method.[2] Bermingham *et al.* [7] concluded that the supervised SVM and multinomial Naïve Bayes classifiers outperform an unsupervised method based on SentiWordNet for tweets and micro-reviews in Blippr. In contrast, Paltoglou *et al.* [6] proposed an unsupervised method based on the LIWC lexicon [14] to detect the polarity of *tweets* and user comments, showing that their method outperforms the Naïve Bayes, Maximum Entropy and SVM classifiers. In sum, there is no consensus as to the best approach for short texts, particularly for Foursquare tips, as previous efforts on that context focused only on lexicon based approaches. This work aims at contributing to this discussion, focusing on detecting the polarity of tips, a noisy environment,

[2] `http://twitrratr.com/`

but rich in information. To our knowledge, this is the first work that provides a deeper investigation of alternative polarity detection methods in this context.

3 Overview of Datasets

We here use two datasets of tips, which are random samples of a larger dataset containing around 10 million tips posted by 13 million users, collected between August and October 2011. To build our datasets, we considered only tips posted in venues with English as the official language,[3] since the tools used by the unsupervised method to determine polarity are constrained to the English language.

To build our first dataset, referred to as **manually labeled dataset**, we randomly selected 1,250 tips to be labeled by 15 volunteers. Each tip was analyzed by 3 volunteers. Each volunteer received 250 tips along with information about the venue (i.e., name, category) where each tip was posted, and was asked to label the content of each tip as positive, negative or neutral. There was agreement among at least two of the volunteers in 94% of the tips. The remaining 6% of tips were discarded due to lack of any agreement. The result of this manual classification was: 57.78% of tips were classified as positive, 15.64% as negative, and 26.58% as neutral. As in [5],[1], we focus on classifying tips as either positive or negative, and thus disregard neutral tips.

The second dataset, referred to as **emoticon based dataset**, was built from a sample of 3,512 tips in English containing at least one emoticon. Emoticons may serve as noisy labels [4], as some texts are not easily classified, such as those that express sarcasm. As in [5],[4], we assume that positive emoticons, notably ':)', '(:', ':-)', '(-:', ':)', ':D', '=D', indicate positive tips, whereas ':(', '):', ':-(', ')-:', ': (', ') :' indicate negative tips.

For both datasets, we considered only tips with at least one word in SentiWordNet to be able to apply our unsupervised method, which caused the removal of only a small fraction of tips (at most 1.6%) from both datasets. In sum, our manually labeled dataset consists of 669 positive and 182 negative tips, and our emoticons based dataset contains 3014 positive and 440 negative tips. Note that both datasets are very unbalanced, reflecting a general trend towards users writing positive tips more often (at least among those written in English).

4 Polarity Detection

This section presents the techniques used to automatically detect the polarity (positive or negative) of Foursquare tips, starting with the supervised methods (Section 4.1), and then introducing our unsupervised approach (Section 4.2).

4.1 Supervised Methods

Given a training set with instances (tips) represented by various features and previously labeled in interest classes (polarities), a supervised method "learns"

[3] `http://en.wikipedia.org/wiki/List_of_official_languages`

a model, which can then be applied to classify unlabeled data (test set) into the given classes. We analyze three state-of-the-art text classifiers - Naïve Bayes (NB), Maximum Entropy (ME) and Support Vector Machine (SVM) [10].

Naïve Bayes is a probabilistic classifier that makes use of Bayes' theorem to infer the probability that a new document (or tip) belongs to each defined class (polarity). It has been applied to various applications such as spam filtering, disease diagnosis, and classification of text polarity [1],[7]. We used the multinomial version of this classifier, which is more adequate for text classification [15], where the probability of a class is parameterized by a multinomial distribution.

The main disadvantage of Naïve Bayes is the assumption of independence between the features exploited by the classifier. Maximum Entropy does not make such assumption. Instead, to estimate the probability distribution, it assumes that, without external knowledge, the distribution should be as uniform as possible, and thus have maximal entropy. The training data is then used to derive a set of constraints that represent the class-specific expectations for the distribution. An improved iterative scaling algorithm is then used to find the maximum entropy distribution without violating the given constraints.

Finally, Support Vector Machines try to find the best hyperplane, defined in the feature space, that separates with maximum distance (margin) the training instances of the two classes (positive and negative). We use a linear kernel, since the number of instances is smaller than the number of features, which is common for textual classification and usually produces a linearly separable problem. We here used the implementations of Naïve Bayes and Maximum Entropy provided in *scikit-learn*[4], and the SVM implementation available in the LIBSVM package.[5]

To apply these algorithms, we modeled each tip as a bag-of-words, removing first the stopwords, as in [1]. However, instead of using the presence/absence of unigrams in the tip as features, we used the representation proposed in [5]: each tip t is modeled as a vector $p_1, ...p_n$, where p_i is the frequency of a term i in tip t and normalized by the frequency of term i in the training set (i.e., TF-IDF). Preliminary experiments showed that this representation gives better results than those obtained using unigrams or bigrams. The values of p_i were used as features exploited by the classification algorithms.

4.2 Unsupervised Method

Supervised methods require previously labeled instances (training) for the development of the classifiers. Unsupervised techniques minimize this need by directly exploiting the contents of the text, often relying on lexicons.

Opinion lexicons are sets of words that express some form of positive or negative feeling (e.g., "amazing" or "bad"). These lists are largely used by polarity detection methods [3], which typically "count" the number of positive or negative words found in a piece of text. Unlike the training sets exploited by supervised techniques, which are typically application-specific, lexicons may be

[4] `http://scikit-learn.org/`
[5] `http://www.csie.ntu.edu.tw/~cjlin/LIBSVM/`

generic enough to be useful for multiple applications in different contexts. Thus, the cost of building the lexicon may be amortized over a larger number of applications and investigations. One aspect to consider when choosing a lexicon is the number of terms included or its *coverage*, which may impact the effectiveness of methods that use the lexicon [3]. Many previously proposed unsupervised methods [3],[16],[7] make use of two specific lexicons: SentiWordNet [11] and LIWC [14]. We here chose SentiWordNet due to its larger coverage.

SentiWordNet is a lexical resource for opinion mining derived from WordNet[17]. WordNet is an English lexicon which groups nouns, verbs, adjectives and adverbs into synonym sets, or *synsets*, each expressing a distinct concept.[6] The SentiWordNet combines three scores - positive, negative and neutral - for every WordNet *synset*, representing the positive, negative or neutral feeling or sentiment associated with that word. These scores are normalized between 0 and 1, so that they add up to 1. Our unsupervised approach to classify a given tip as positive or negative consists of the following steps:

1. Part-Of-Speech and Stemming[7]: Each word in the tip is first associated with a single grammatical category, such as an adjective, adverb, verb and pronoun, using a part-of-speech parser, and then converted to its canonical form (e.g., verbs are kept in their infinitive form).

2. Treatment of negative terms:the polarity of a word may be influenced if a negative term (e.g., *not*) precedes it. To handle this scenario, we build a dependency tree that models the grammatical relations of each word or phrase of the tip, and used it to identify the words that are influenced by a negative term. These words have their positive and negative SentiWordNet scores exchanged.

3. Sense of a word: a word in SentiWordNet may have multiple *synsets* associated with the same part-of-speech. Thus, we considered the mean of the scores of all the *synsets* associated with the part of speech of the word.

4. Tip Polarity: we assign positive, negative and neutral scores to the tip, each one computed as the average of the corresponding scores of the *synsets* of all words of the tip that were found in SentiWordNet. A tip is considered as having a positive polarity if its final positive score is higher than the negative one, and vice-versa. In case of tie, the polarity of the tip is considered as undefined, and the tip is discarded from our evaluation.

5 Experimental Results

We evaluated the four polarity detection methods on our two datasets (Section 3) using a 5-fold cross-validation: each dataset was divided into 5 folds, 4 folds were used as training set, and 1 fold as test set. The training set was used only by the supervised methods to "learn" the classification models. In particular, best parameter values were determined by performing cross-validation within the training sets. All methods were evaluated only on the test sets. In order to compensate for the large class imbalance, we applied a commonly used technique

[6] Every possible meaning of the same word corresponds to a different *synset*

[7] We use a tool from `http://www-nlp.stanford.edu/software/corenlp.shtml`.

of undersampling, in which the smallest class determines the number of instances of each class used for training. Thus, for each round of the cross-validation we produced 5 random samples for each of the training classes, and repeated the process 5 times. The results discussed here are averages of 25 runs, along with corresponding 95% confidence intervals.

Table 1: Results of Polarity Detection Methods

Metric	Method	Manually labeled dataset			Emoticon based dataset		
		Positive class	Negative class	Average	Positive class	Negative class	Average
Precision	NB	**0.9173±0.0067**	0.4333±0.0180	**0.6753±0.0103**	**0.9393±0.0033**	**0.2457±0.0110**	**0.5925±0.0056**
	ME	0.9017±0.0092	0.3950±0.0206	0.6484±0.0120	0.9270±0.0040	0.2193±0.0098	0.5731±0.0050
	SVM	**0.9097±0.0096**	0.4169±0.0221	0.6633±0.0127	**0.9399±0.0040**	**0.2394±0.0133**	**0.5896±0.0065**
	Lexicon	0.8861±0.0121	**0.4846±0.0390**	**0.6853±0.0189**	0.9139±0.0054	**0.2416±0.0127**	0.5778±0.0070
	Hybrid	**0.9179±0.0082**	0.4381±0.0231	**0.6780±0.0119**	**0.9365±0.0037**	**0.2437±0.0134**	**0.5901±0.0066**
Recall	NB	0.7311±0.0126	**0.7547±0.0241**	**0.7429±0.0119**	0.6888±0.0078	**0.6936±0.0181**	**0.6912±0.0087**
	ME	0.7015±0.0201	0.7124±0.0364	0.7070±0.0146	0.6670±0.0078	0.6400±0.0156	0.6535±0.0085
	SVM	0.7176±0.0270	**0.7302±0.0387**	0.7239±0.0157	0.6698±0.0227	**0.7028±0.0266**	**0.6863±0.0096**
	Lexicon	**0.8183±0.0180**	0.6159±0.0276	0.7171±0.0154	**0.7651±0.0055**	0.5101±0.0247	0.6376±0.0122
	Hybrid	0.7335±0.0227	**0.7521±0.0338**	**0.7428±0.0132**	0.6900±0.0206	**0.6768±0.0248**	**0.6834±0.0900**
F1	NB	0.8133±0.0083	**0.5496±0.0190**	**0.6814±0.0116**	0.7946±0.0051	**0.3623±0.0133**	**0.5785±0.0078**
	ME	0.7879±0,0116	0.5058±0.0222	0.6469±0.0135	0.7756±0.0055	0.3261±0.0118	0,5508±0.0075
	SVM	0.8003±0.0166	**0.5278±0.0226**	0.6640±0.0157	0.7807±0.0153	**0.3554±0.0150**	**0.5681±0.0128**
	Lexicon	**0.8502±0.0118**	**0.5369±0.0313**	**0.6935±0.0187**	**0.8328±0.0040**	0.3271±0.0156	**0.5800±0.0086**
	Hybrid	**0.8141±0.0125**	**0.5504±0.0213**	**0.6823±0.0140**	0.7934±0.0135	**0.3569±0.0152**	**0.5752±0.0123**

The effectiveness of each method was evaluated in terms of precision, recall and F1. The precision p of a class c is the number of tips correctly classified in class c by the number of tips predicted as c. The recall r of a class c is the number of tips correctly classified in class c by the number of tips in c. The F1 measure is the harmonic mean, $2pr/(p+r)$ between precision p and recall r. We computed precision, recall and F1 for each class (polarity) separately, as well as average values for the two classes.

Table 1 shows the results of each supervised approach - Naïve Bayes (NB), Maximum Entropy (ME), SVM - and the unsupervised method (Lexicon) for both datasets. Best results, including statistical ties, are shown in bold. The significance of these values was tested using a paired t-test considering a confidence interval of 95%.

Regarding precision, the best results for the positive class are produced by the supervised NB and SVM methods, which are statistically tied in both datasets. NB, in particular, produces gains of up to 3.6% and 2.8% over the other methods, on average, in the manually labeled and emoticon based datasets, respectively. For the negative class, in turn, the unsupervised lexicon based method is the best performer, with gains of up to 22.7% over the others in the manually labeled dataset, although it is statistically tied with both SVM and NB in the emoticon based dataset. Note that, in general, precision results are smaller for the negative class, due to the class imbalance which leads to a dominance of the largest (positive) class on the classification results. The differences are larger in the

emoticon-based dataset where the imbalance is more severe. Finally, considering average precision, both NB and the lexicon based method are statistically tied as the best solutions for the manually labeled dataset, whereas, in the emoticon based dataset, the supervised SVM and NB produce better results, with statistically significant but small gains of up to 2.5% over the unsupervised method.

In terms of recall of the positive class, the unsupervised approach outperforms all supervised methods, in both datasets, with average gains of up to 11.9% over the best supervised method. However, for the negative class, the best recall is produced by the NB and SVM classifiers, which are statistically tied in both datasets, with gains over the unsupervised method of 22.5% (manually labeled dataset) and 36% (emoticon based dataset). The larger gains in the negative class lead to a superiority of the supervised methods in terms of average recall in both classes: the best performer in both datasets is NB, although SVM appears tied with it as best solution in the emoticon based dataset.

Considering the F1 metric, which combines precision and recall, the lexicon based method produces the best results for positive tips in both datasets. For negative tips, NB and SVM are tied with the lexicon based method as best performers in the manually labeled dataset, whereas in the emoticon based dataset, the two supervised classifiers stand out as the best methods. Overall, NB and the lexicon based methods appear tied as the best methods in both datasets, whereas, in the emoticon based dataset, this tie also includes SVM.

Finally, we also tested a hybrid approach that combines, by means of a stacking method [18], the results of the supervised and the unsupervised methods. In this technique, the predictions of the three supervised methods as well as the scores produced by the unsupervised method are used as input to another supervised classifier (an SVM with a linear kernel, in the present case), which learns how to combine the outputs of all methods (e.g., by assigning proper weights for each single prediction). The results of this hybrid approach are also shown in Table 1. Note that, in both datasets, the results of the hybrid method are, at best, statistically tied with the best performer, although in some cases (e.g., recall of positive class), the results are clearly worse. The lack of improvements from the hybrid method is possibly due to the large agreement among the methods (e.g., the NB and the lexicon based methods produce the same results for 70% of the tips in the manually labeled dataset), which leaves little room for improvement from the stacking approach. Further investigations with other classifiers and alternative strategies to combine multiple methods are required and are left for future work.

Our results can be summarized as follows:

- The unsupervised lexicon based method produces better or statistically tied results, in terms of average F1, when compared to the best supervised methods (NB and SVM) in both datasets. The hybrid method, in turn, does not lead to further improvements.

- The unsupervised method improves the F1 of the positive class in up to 4.8% over the best supervised method. The gains in recall, which is particularly

important if one is interested in retrieving most tips of that class, reach 11.9%. Thus, this method should be used when the focus is on *positive tips*.

- If the focus is on the *negative tips*, the best methods are the supervised ones, particularly NB and SVM. In the emoticon based dataset, these supervised methods produce gains in F1 of up to 10.8% over the unsupervised solution. In terms of recall, the gains are even more impressive, reaching 36% in that dataset.

- All methods perform better, in all metrics, in the manually labeled dataset (differences of up to 19.5%), possibly due to a higher level of noise (e.g., sarcasm) and uncertainty in the automatic labeling through emoticons.

These results indicate that, in the specific context of Foursquare tips, the overall effectiveness of the unsupervised lexicon based method is comparable to the best supervised methods (NB and SVM), without the labeling costs associated with the latter. However, the choice of the best method should take the costs and limitations of each approach into account. A supervised method generally requires a costly manual labeling effort. Automatic labeling, for example by exploring emoticons, may be employed. However its effectiveness may be limited by the coverage of emoticons in the tips[8] as well as by higher levels of noise and uncertainty. Unsupervised methods, on the other hand, require the availability of a lexicon for the target language or domain. Moreover, with this type of method, there are constraints in the lexicon coverage of the target domain, which may cause some tips not to be classified. In particular, the method based on the SentiWordNet could not be applied in 1.4% and 1.6% of the tips originally present in the manually labeled and emoticon based datasets, respectively, since none of the words in these tips could be found in the lexicon.[9]

6 Conclusions and Future Work

We have analyzed the effectiveness of three state-of-the-art supervised classifiers - Naïve Bayes, SVM and Maximum Entropy, an unsupervised method that uses the SentiWordNet lexicon as data source, as well as a hybrid approach that combines all four methods by means of stacking, for polarity detection of Foursquare tips. We evaluated all methods in two sets of tips: one manually labeled by volunteers and the other automatically labeled by exploring the presence of emoticons. Our results indicate that, in terms of average F1, the unsupervised method produces better or statistically tied results when compared to the best supervised methods, which are Naïve Bayes and SVM, in both sets. Considering each class separately, we find that although the lexicon based method is better to retrieve positive tips, the Naïve Bayes and SVM classifiers produce great gains, in terms of both F1 and recall, if the focus is on retrieving the negative tips. However, the choice of the best solution should also take the availability of resources and costs associated with each method (training set, lexicon) into account. Finally, we also found that the hybrid approach does not produce significant improvements over

[8] Only 3.39% of English tips in our dataset of 10 million tips contain emoticons.
[9] These tips were filtered from both datasets (see Section 3).

the best individual technique, possibly due to the large agreement among the methods.

Future work includes extending our evaluation to other datasets, investigating other strategies to combine multiple methods, and using our methods to build opinion summarization and venue recommendation methods.

7 Acknowledgments

This research is partially funded by the Brazilian National Institute of Science and Technology for the Web (MCT/CNPq/INCT grant number 573871/2008-6), CNPq, CAPES and FAPEMIG.

References

1. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? Sentiment Classification using Machine Learning Techniques. In: Proc. EMNLP. (2002)
2. Guerra, P., Veloso, A., Meira, W., Almeida, V.: From Bias to Opinion: a Transfer-Learning Approach to Real-Time Sentiment Analysis. In: Proc. SIGKDD. (2011)
3. Ohana, B., Tierney, B.: Sentiment classification of reviews using SentiWordNet. In: Proc. of 9th IT & T. (2009)
4. Go, A., Bhayani, R., Huang, L.: Twitter Sentiment Classification using Distant Supervision. Technical report, Stanford University (2009)
5. Aisopos, F., Papadakis, G., Tserpes, K., Varvarigou, T.: Content vs. Context for Sentiment Analysis: a Comparative Analysis over Microblogs. In: Proc. HT. (2012)
6. Paltoglou, G., Thelwall, M.: Twitter, MySpace, Digg: Unsupervised Sentiment Analysis in Social Media. ACM TIST **3**(4) (2012)
7. Bermingham, A., Smeaton, A.: Classifying Sentiment in Microblogs: is Brevity an Advantage? In: Proc. CIKM. (2010)
8. Pang, B., Lee, L.: Opinion Mining and Sentiment Analysis. Foundations and Trends in Information Retrieval **2**(1-2) (2008)
9. Lu, Y., Castellanos, M., Dayal, U., Zhai, C.: Automatic Construction of a Context-Aware Sentiment Lexicon: an Optimization Approach. In: Proc. WWW. (2011)
10. Pustejovsky, J., Stubbs, A.: Natural Language Annotation for Machine Learning. O'Reilly Media (2012)
11. Esuli, A., Sebastiani, F.: Sentiwordnet: A Publicly Available Lexical Resource for Opinion Mining. In: Proc. LREC. (2006)
12. Carlone, D., Ortiz-Arroyo, D.: Semantically Oriented Sentiment Mining in Location-Based Social Network Spaces. In: Proc. FQAS. (2011)
13. Yang, D., Zhang, D., Yu, Z., Wang, Z.: A sentiment-enhanced personalized location recommendation system. In: Proc. ACM HT. (2013)
14. Tausczik, Y., Pennebaker, J.: The Psychological Meaning of Words: LIWC and Computerized Text Analysis Methods. JLS **29**(1) (2010)
15. McCallum, A., Nigam, K.: A Comparison of Event Models for Naive Bayes Text Classification. In: Proc. AAAI Workshop Learning for Text Categorization. (1998)
16. Hamouda, A., Rohaim, M.: Reviews Classification Using SentiWordNet Lexicon. OJCSIT **2**(4) (2011)
17. Miller, G.: WordNet: a Lexical Database for English. Comm. of ACM **38**(11) (1995)
18. Dzeroski, S., Zenko, B.: Is Combining Classifiers with Stacking Better than Selecting the Best One? JMLR **54(3)** (2004)