Random Walk Weighting over SentiWordNet for Sentiment Polarity Detection on Twitter

A. Montejo-Ráez, E. Martínez-Cámara, M. T. Martín-Valdivia, L. A. Ureña-López
University of Jaén
E-23071, Jaén (Spain)
{amontejo, emcamara, maite, laurena}@ujaen.es

Abstract

This paper presents a novel approach in Sentiment Polarity Detection on Twitter posts, by extracting a vector of weighted nodes from the graph of WordNet. These weights are used on SentiWordNet to compute a final estimation of the polarity. Therefore, the method proposes a non-supervised solution that is domain-independent. The evaluation over a generated corpus of tweets shows that this technique is promising.

1 Introduction

The birth of Web 2.0 supposed a breaking down of the barrier between the consumers and producers of information, i.e. the Web has changed from a static container of information into a live environment in which any user, in a very simple manner, can publish any type of information. This simplified means of publication has led to the rise of several different websites specialized in the publication of users opinions. Some of the most well-known sites include Epinions[1], RottenTomatoes[2] and Muchocine[3], where users express their opinions or criticisms on a wide range of topics. Opinions published on the Internet are not limited to certain sites, but rather can be found in a blog, forum, commercial website or any other site allowing posts from visitors.

On of the most representative tools of the Web 2.0 are social networks, which allow millions of users to publish any information in a simple way and to share it with their network of contacts or "friends". These social networks have also evolved and become a continuous flow of information. A clear example is the microblogging platform Twitter[4]. Twitter publishes all kinds of information, disseminating views on many different topics: politics, business, economics and so on. Twitter users regularly publish their comments on a particular news item, a recently purchased product or service, and ultimately on everything that happens around them. This has aroused the interest of the Natural Language Processing (NLP) community, which has begun to study the texts posted on Twitter, and more specifically related to Sentiment Analysis (SA) challenges.

In this manuscript we present a new approach to resolve the scoring of posts according to the expressed positive or negative degree in the text. This polarity detection problem is resolved by combining SentiWordNet scores with a random walk analysis of the concepts found in the text over the WordNet graph. In order to validate our non-supervised approach, several experiments have been performed to analyze major issues in our method and to compare it with other approaches like plain SentiWordNet scoring or machine learning solutions such as Support Vector Machines in a supervised approach. The paper is structured as follows: first, an introduction to the polarity detection problem is provided, followed by the description of our approach. Then, the experimental setup is given with a description of the generated corpus and the results obtained. Finally, conclusions and further work are discussed.

[1]http://epinions.com
[2]http://rottentomatoes.com
[3]http://muchocine.net
[4]http://twitter.com

Proceedings of the 3rd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis, pages 3–10,
Jeju, Republic of Korea, 12 July 2012.

2 The polarity detection problem

In the literature related to the SA in long text a distinction is made between studies of texts where we assume that the text is a opinion and therefore solely need to calculate its polarity, and those in which before measuring polarity it is necessary to determine whether the text is subjective or objective. A wide study on SA can be found in (Pang and Lee, 2008), (Liu, 2010) and (Tsytsarau and Palpanas, 2011). Concerning the study of the polarity in Twitter, most experiments assume that tweets[5] are subjective. One of the first studies on the classification of the polarity in tweets was published in 2009 by (Go et al., 2009), in which the authors conducted a supervised classification study of tweets in English.

Zhang et al. (Zhang et al., 2011) proposed a hybrid method for the classification of the polarity in Twitter, and they demonstrated the validity of their method over an English corpus on Twitter. The classification is divided into two phases. The first one consists on applying a lexicon-based method. In the second one the authors used the SVM algorithm to determine the polarity. For the machine learning phase, it is needed a labelled corpus, so the purpose of the lexicon-method is to tag the corpus. Thus, the authors selected a set of subjective words from all those available in English and added hash-tags with a subjective meaning. After labelling the corpus, it is used SVM for classifying new tweets.

In (Agarwal et al., 2011) a study was conducted on a reduced corpus of tweets labelled manually. The experiment tests different methods of polarity classification and starts with a base case consisting on the simple use of unigrams. Then a tree-based model is generated. In a third step, several linguistic features are extracted and finally a final model learned as combination of the different models proposed is computed. A common feature used both in the tree-based model and in the feature-based one is the polarity of the words appearing in each tweet. In order to calculate this polarity the authors used DAL dictionary (Whissell, 1989).

Most of the proposed systems for polarity detection compute a value of negativeness or positiveness. Some of them even produce a neutrality value. We will consider the following measurement of polarity (which is very common, indeed): a real value in the interval [-1, 1] would be sufficient. Values over zero would reflect a positive emotion expressed in the tweet, while values below zero would rather correspond to negative opinions. The closer to the zero value a post is, the more its neutrality would be. Therefore, a polarity detection system could be represented as a function p on a text t such as:

$$p : \mathbb{R}^N \to \mathbb{R}$$

so that $p(t) \in [-1, 1]$. We will define how to compute this function, but before an explanation of the techniques implied in such a computation is provided.

3 The approach: Random Walk and SentiWordNet

3.1 The Random Walk algorithm

Personalized Page Rank vectors (PPVs) consists on a ranked sequence of WordNet (Fellbaum, 1998) synsets weighted according to a random walk algorithm. Taking the graph of WordNet, where nodes are synsets and axes are the different semantic relations among them, and the terms contained in a tweet, we can select those synsets that correspond to the closest sense for each term and. Then, it starts an iterative process so more nodes are selected if they are not far from these "seeds". After a number of iterations or a convergence of the weights, a final list of valued nodes can be retrieved. A similar approach has been used recently by (Ramage et al., 2009) to compute text semantic similarity in recognizing textual entailment, and also as a solution for word sense disambiguation (Agirre and Soroa, 2009). We have used the UKB software from this last citation to generate the PPVs used in our system. Random walk algorithms are inspired originally by the Google PageRank algorithm (Page et al., 1999). The idea behind it is to represent each tweet as a vector weighted synsets that are semantically close to the terms included in the post. In some way, we are *expanding* these sort texts by a set of disambiguated concepts related to the terms included in the text.

As an example of a PPV,the text *"Overall, we're still having a hard time with it, mainly because we're not finding it in an early phase.*" becomes the vector of weighted synsets:

[5] The name of posts in Twitter.

```
[02190088-a:0.0016, 12613907-n:0.0004,
01680996-a:0.0002, 00745831-a:0.0002, ...]
```

Here, the synset `02190088-a` has a weight of 0.0016, for example.

3.2 SentiWordNet

SentiWordNet (Baccianella et al., 2008) is a lexical resource based on the well know WordNet (Fellbaum, 1998). It provides additional information on synsets related to sentiment orientation. A synset is the basic item of information in WordNet and it represents a "concept" that is unambiguous. Most of the relations over the lexical graph use synsets as nodes (hyperonymy, synonymy, homonymy and more). SentiWordNet returns from every synset a set of three scores representing the notions of "positivity", "negativity" and "neutrality". Therefore, every concept in the graph is weighting according to its subjectivity and polarity. The last version of SentiWordNet (3.0) has been constructed starting from manual annotations of previous versions, populating the whole graph by applying a random walk algorithm. This resource has been used by the opinion mining community, as it provides a domain-independent resource to get certain information about the degree of emotional charge of its concepts (Denecke, 2008; Ogawa et al., 2011).

3.3 Computing the final estimation

As a combination of SentiWordNet scores with random walk weights is wanted, it is important that the final equation leads to comparable values. To this end, the weights associated to synsets after the random walk process are L_1 normalized so vectors of "concepts" sum up the unit as maximum value. The final polarity score is obtained by the product of this vector with associated SentiWordNet vector of scores, as expressed in equation 1.

$$p = \frac{\mathbf{r} \cdot \mathbf{s}}{|\mathbf{t}|} \tag{1}$$

where p is the final score, $\mathbf{r}$ is the vector of weighted synsets computed by the random walk algorithm of the tweet text over WordNet, $\mathbf{s}$ is the vector of polarity scores from SentiWordNet, t is the set of concepts derived from the tweet. The idea behind it is to "expand" the set of concepts with additional ones that are close in the WordNet graph, corresponding to those synset nodes which have been activated during the random walk process. Therefore, terms like *dog* and *bite* (both mainly neutral in SentiWordNet) appearing in the same tweet could eventually be expanded with a more emotional term like *hurt*, which holds, in SentiWordNet, a negative score of 0.75.

4 Experiments and results

Our experiments are focused in testing the validity of applying this unsupervised approach compared to a classical supervised one based on Support Vector Machines (Joachims, 1998). To this end, the corpus has been processed obtaining lemmas, as this is the preferred input for the UKB software. The algorithm takes the whole WordNet graph and performs a disambiguation process of the terms as a natural consequence of applying random walk over the graph. In this way, the synsets that are associated to these terms are all of them initialized. Then, the iterative process of the algorithm (similar to Page Rank but optimized according to an stochastic solution) will change these initial values and propagate weights to closer synsets. An interesting effect of this process is that we can actually obtain more concepts that those contained in the tweet, as all the related ones will also finalize with a certain value due to the propagation of weights across the graph. We believe that our approach benefits from this effect, as texts in tweets use to suffer from a very sort length, allowing us to expand short posts.

Another concern is, therefore, the final size of the PPV vector. If too many concepts are taken into account we may introduce noise in the understanding of the latent semantic of the text. In order to study this fact, different sizes of the vector have been explored and evaluated.

4.1 Our Twitter corpus

The analysis of the polarity on microblogging is a very recent task, so there are few free resources (Saša et al., 2010). Thus, we have collected our own English corpus in order to accomplish the experiments. The work of downloading tweets is not nearly difficult due to the fact that Twitter offers two kinds of API to those purposes. We have used the

Search API of Twitter[6] for automatically accessing tweets through a query. For a supervised polarity study and to evaluate our approach, we need to generate a labelled corpus. We have built a corpus of tweets written in English following the procedure described in (Read, 2005) and (Go et al., 2009).

According to (Read, 2005), when authors of an electronic communication use an emotion, they are effectively marking up their own text with an emotional state. The main feature of Twitter is that the length of the messages must be 140 characters, so the users have to express their opinions, thoughts, and emotional states with few words. Therefore, frequently users write "smileys" in their tweets. Thus, we have used positive emoticons to label positive tweets and negative emoticons to tag negative tweets. The full list of emoticons that we have considered to label the retrieved tweets can be found in Table 1. So, following (Go et al., 2009), the presumption in the construction of the corpus is that the query ":)" returns tweets with positive smileys, and the query ":(" retrieves negative emotions. We have collected a set of 376,296 tweets (181,492 labelled as positive tweets and 194,804 labelled as negative tweets), which were published on Twitter's public message board from September 14^{th} 2010 to March 19^{th} 2011. Table 2 lists other characteristics of the corpus.

On the other hand, the language used in Twitter has some unique attributes, which have been removed because they do not provide relevant information for the polarity detection process. These specific features are:

1. **Retweets**: A retweet is the way to repeat a message that users consider interesting. Retweets can be done through the web interface using the Retweet option, or as the old way writing RT, the user name and the post to retweet. The first way is not a problem because is the same tweet, so the API only return it once, but old way retweets are different tweets but with the same content, so we removed them to avoid pitting extra weight on any particular tweet.

2. **Mentions**: Other feature of Twitter is the so called Mentions. When a user wants to refer to another one, he or she introduces a Mention. A Mention is easily recognizable because all of them start with the symbol "@" followed by the user name. We consider that this feature does not provide any relevance information, so we have removed the mentions in all the tweets.

3. **Links**: It is very common that tweets include web directions. In our approach we do not analyze the documents that links those urls, so we have eliminated them from all tweets.

4. **Hash-tags**: A hash-tag is the name of a topic in Twitter. Anybody can begin a new topic by typing the name of the topic preceded by the symbol "#". For this work we do not classify topics so we have neglected all the hash-tags.

[6] https://dev.twitter.com/docs/api/1/get/search

Emoticons mapped to :) (positive tweets)	:)	:)	:-)
	;)	;-)	=)
	^_^	:-D	:D
	:d	=D	C:
	Xd	XD	xD
	Xd	(x	(=
	^^	^o^	'u'
	n_n	*-*	*O*
	o	*_*	
Emoticons mapped to :((negative tweets)	:-(	:(	:((
	: (	D:	Dx
	'n'	:\	/:
	):-/	:'	='[
	:_(	/T_T	TOT
	;_;		

Table 1: Emoticons considered as positives and negatives

Due to the fact that users usually write tweets with a very casual language, it is necessary to preprocess the raw tweets before feeding the sentiment analyzer. For that purpose we have applied the following filters:

1. **Remove new lines**: Some users write tweets in two or three different lines, so all newlines symbols were removed.

2. **Opposite emoticons**: Twitter sometimes considers positive or negative a tweet with smileys

			Total
Positive tweets	181,492		
Negative tweets	194,804		376,296
Unique users in positive tweets	157,579		
Unique users in negative tweets	167,479		325,058
Words in positive tweets	418,234		
Words in negative tweets	334,687		752,921
Average number of words per positive tweet	9		
Average number of words per negative tweet	10		

Table 2: Statistical description of the corpus.

that have opposite senses. For example:

```
@Harry_Styles I have all day to try
get a tweet off you :) when are
you coming back to dublin i missed
you last time,I was in spain :(
```

The tweet has two parts one positive and the other one negative, so the post cannot be considered as positive, but the search API returns as a positive tweet because it has the positive smiley ":)". We have removed this kind of tweets in order to avoid ambiguity.

3. **Emoticons with no clear sentiment**: The Twitter Search API considers some emoticons like ":P" or ":PP" as negative. However, some users do not type them to express a negative sentiment. Thus, we have got rid of all tweets with this kind of smileys (see Table 3).

Fuzzy emoticons	:-P	:P	:PP	\(

Table 3: Emoticons considered as fuzzy sentiments

4. **Repeated letters**: Users frequently repeat several times letters of some words to emphasize their messages. For example:

```
Blood drive todayyyy!!!!! :)
Everyone donateeeee!!
```

This can be a problem for the classification process, because the same word with different repetitions of the same letter would be considered as a different word. Thus, we have normalized all the repeated letters, and any letter occurring more than two times in a word is replaced with two occurrences. The example above would be converted into:

```
blood drive todayy :) everyone
donatee!!
```

5. **Laugh**: There is not a unique manner to express laugh. Therefore, we have normalized the way to write laugh. Table 4 lists the conversions.

Laugh	Conversion
hahahaha...	haha
hehehehe...	hehe
hihihihi...	hihi
hohohoho...	hoho
huhuhuhu...	huhu
Lol	haha
Huashuashuas	huas
muahahaha	Buaha
buahahaha	Buaha

Table 4: Normalization for expressions considered as "Laugh"

Finally, although the emoticons have been used to tag the positive and negative samples, the final corpora does not include these emoticons. In addition, all the punctuation characters have been neglected in order to reduce the noise in the data. Figure 1 shows the process to generate our Twitter corpus.

4.2 Results obtained

Our first experiment consisted on evaluating a supervised approach, like Support Vector Machines, using the well know vector space model to build the vector of features. Each feature corresponds to the TF.IDF weight of a lemma. Stop words have not been removed and the minimal document frequency required was two, that is, if the lemma is not present in two o more tweets, then it is discarded as a dimension in the vectors. The SVM-Light[7] software was used to compute support vectors and to evaluate them using a random leave-one-out strategy. From

[7] http://svmlight.joachims.org/

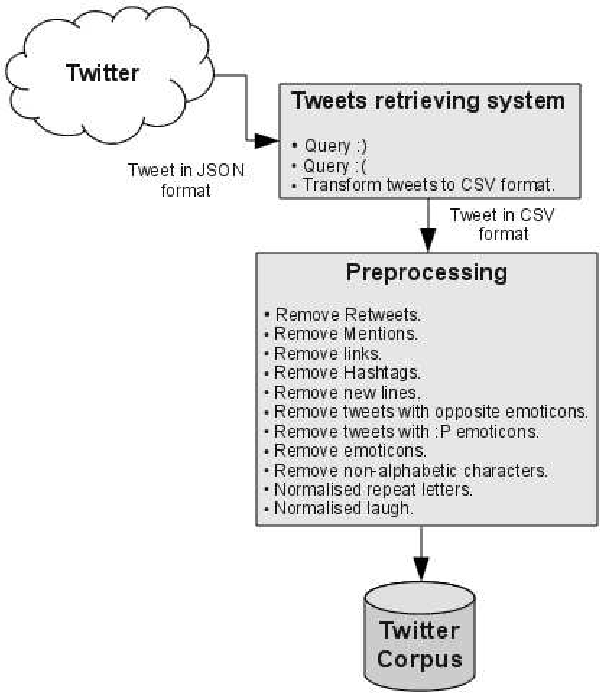

Figure 1: Corpus generation work-flow

a total of 376,284 valid samples 85,423 leave-one-out evaluations were computed. This reported the following measurements:

Precision	**Recall**	**F1**
0.6429	0.6147	0.6285

In our first implementation of our method, the final polarity score is computed as described in equation 1. More precisely, it is the average of the product between the difference of positive and negative SentiWordNet scores, and the weight obtained with the random walk algorithm, as unveiled in equation 2.

$$p = \frac{\sum_{\forall s \in t} rw_s \cdot (swn_s^+ - swn_s^-)}{|t|} \tag{2}$$

Where s is a synset in the tweet t, rw_s is the weight of the synset s after the random walk process over WordNet, swn_s^+ and $swn_s^-)$ are positive and negative scores for the synset s retrieved from SentiWordNet.

The results obtained are graphically shown in figures 2, 3 and 4 for precision, recall and F1 values respectively. As can be noticed from the shapes of the graphs, the size of the PPV vectors affects the performance. Sizes above 10 presents an stable behavior, that is, considering a large number of synsets does not improves the performance of the system, but it gets worse neither. The WordNet graph considered for the random walk algorithm includes antonyms relations, so we wanted to check whether discarding these connections would affect the system. From these graphs we can extract the conclusion that antonyms relations are worth keeping.

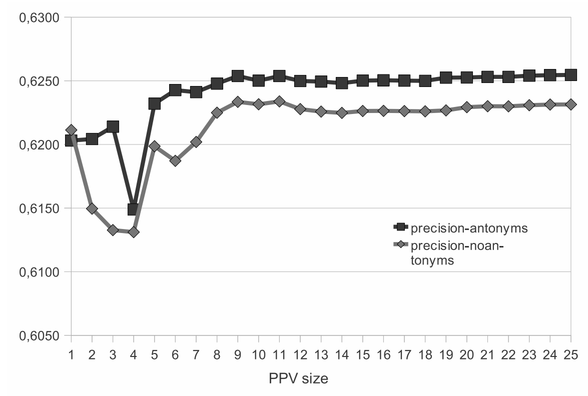

Figure 2: Precision values against PPV sizes

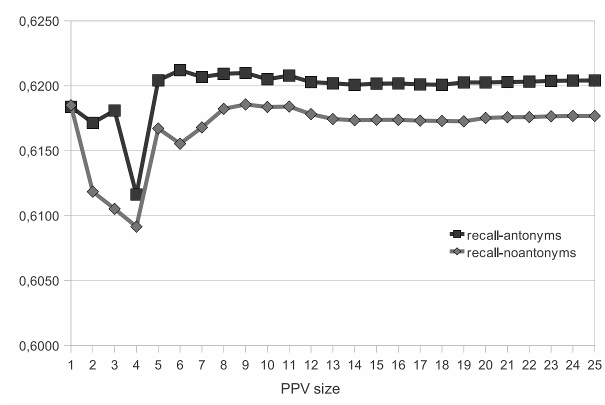

Figure 3: Recall values against PPV sizes

Comparing our best configuration to the SVM approach, the results are not better, but quite close (table 5). Therefore, this unsupervised solution is an interesting alternative to the supervised one.

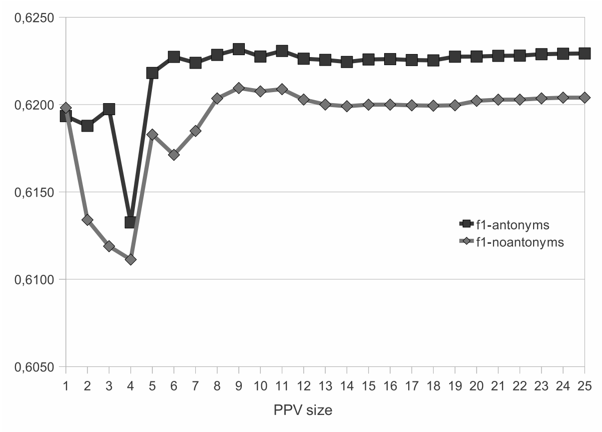

Figure 4: F1 values against PPV sizes

	Precision	**Recall**	**F1**
SVM	0.6429	0.6147	0.6285
RW·SWN	0.6259	0.6207	0.6233

Table 5: Approaches comparative table

5 Conclusions and further work

A new unsupervised approach to the polarity detection problem in Twitter posts has been proposed. By combining a random walk algorithm that weights synsets from the text with polarity scores provided by SentiWordNet, it is possible to build a system comparable to a SVM based supervised approach in terms of performance. Our solution is a general approach that do not suffer from the disadvantages associated to supervised ones: need of a training corpus and dependence on the domain where the model was obtained.

Many issues remain open and they will drive our future work. How to deal with negation is a major concern, as the score from SentiWordNet should be considered in a different way in the final computation if the original term comes from a negated phrase. Our "golden rules" must be taken carefully, because emoticons are a rough way to classify the polarity of tweets. Actually, we are working in the generation of a new corpus in the politics domain that is now under a manual labeling process. Another step is to face certain flaws in the computation of the final score. In this sense, we plan to study the context of a tweet among the time line of tweets from that user to identify publisher's mood and adjust final scores. As an additional task, the processing of original texts is important. The numerous grammatical and spelling errors found in this fast way of publication demand for a better sanitization of the incoming data. An automatic spell checker is under development.

As final conclusion, we believe that this first attempt is very promising and that it has arose many relevant questions on the subject of sentiment analysis. More extensive research and experimentation is being undertaken from the starting point introduced in this paper.

Acknowledgments

This work has been partially supported by a grant from the Fondo Europeo de Desarrollo Regional (FEDER), TEXT-COOL 2.0 project (TIN2009-13391-C04-02) from the Spanish Government. This paper is partially funded by the European Commission under the Seventh (FP7 - 2007-2013) Framework Programme for Research and Technological Development through the FIRST project (FP7-287607). This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

References

Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, and Rebecca Passonneau. 2011. Sentiment analysis of twitter data. In *Proceedings of the Workshop on Language in Social Media (LSM 2011)*, pages 30–38, Portland, Oregon, jun. Association for Computational Linguistics.

Eneko Agirre and Aitor Soroa. 2009. Personalizing pagerank for word sense disambiguation. In *EACL '09: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 33–41, Morristown, NJ, USA. Association for Computational Linguistics.

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2008. Sentiwordnet 3.0 : An enhanced lexical resource for sentiment analysis and opinion mining. *Proceedings of the Seventh conference on International Language Resources and Evaluation LREC10*, 0:2200–2204.

K. Denecke. 2008. Using sentiwordnet for multilingual sentiment analysis. In *Data Engineering Workshop,* *2008. ICDEW 2008. IEEE 24th International Conference on*, pages 507 –512, april.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA.

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

T. Joachims. 1998. Text categorization with support vector machines: learning with many relevant features. In *European Conference on Machine Learning (ECML)*.

Bing Liu. 2010. Sentiment analysis and subjectivity. *Handbook of Natural Language Processing*, 2nd ed.

Tatsuya Ogawa, Qiang Ma, and Masatoshi Yoshikawa. 2011. News Bias Analysis Based on Stakeholder Mining. *IEICE TRANSACTIONS ON INFORMATION AND SYSTEMS*, E94D(3):578–586, MAR.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford University.

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, 2(1-2):1–135.

Daniel Ramage, Anna N. Rafferty, and Christopher D. Manning. 2009. Random walks for text semantic similarity. In *TextGraphs-4: Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing*, pages 23–31, Morristown, NJ, USA. Association for Computational Linguistics.

Jonathon Read. 2005. Using emoticons to reduce dependency in machine learning techniques for sentiment classification. In *Proceedings of the ACL Student Research Workshop*, ACLstudent '05, pages 43–48, Stroudsburg, PA, USA. Association for Computational Linguistics.

Petrović Saša, Miles Osborne, and Victor Lavrenko. 2010. The edinburgh twitter corpus. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Linguistics in a World of Social Media*, WSA '10, pages 25–26, Stroudsburg, PA, USA. Association for Computational Linguistics.

Mikalai Tsytsarau and Themis Palpanas. 2011. Survey on mining subjective data on the web. *Data Mining and Knowledge Discovery*, pages 1–37, October.

C M Whissell, 1989. *The dictionary of affect in language*, volume 4, pages 113–131. Academic Press.

Ley Zhang, Riddhiman Ghosh, Mohamed Dekhil, Meichun Hsu, and Bing Liu. 2011. Combining lexicon-based and learning-based methods for twitter sentiment analysis. Technical Report HPL-2011-89, HP, 21/06/2011.