GPLSI: Supervised Sentiment Analysis in Twitter using Skipgrams

Javi Fernández, Yoan Gutiérrez, José M. Gómez, Patricio Martínez-Barco
Department of Software and Computing Systems
University of Alicante
{javifm,ygutierrez,jmgomez,patricio}@dlsi.ua.es

Abstract

In this paper we describe the system submitted for the *SemEval 2014 Task 9 (Sentiment Analysis in Twitter) Subtask B*. Our contribution consists of a supervised approach using machine learning techniques, which uses the terms in the dataset as features. In this work we do not employ any external knowledge and resources. The novelty of our approach lies in the use of words, ngrams and *skipgrams* (not-adjacent ngrams) as features, and how they are weighted.

1 Introduction

The Web 2.0 has become one of the most important sources of data to extract useful and heterogeneous knowledge from. Texts can provide *factual information*, such as descriptions and lists of features, and *opinion-based information*, which would include reviews, emotions, or feelings. This subjective information can be expressed through different textual genres, such as blogs, forums, social networks and microblogs.

An example of microblogging social network is Twitter[1], which has gained much popularity in the last years. This website enables its users to send and read text-based messages of up to 140 characters, known as *tweets*. This site can be a vast source of subjective information in real time; millions of users share opinions on different aspects of their everyday life. Extracting this subjective information has a great value for both general and expert users. However, it is difficult to exploit it accordingly, mainly because of the short length of the tweets, the informality, and the lack of context. *Sentiment Analysis* (SA) systems must be adapted to face the challenges of this new textual genre.

International competitions related to the assessment of SA systems in Twitter have taken place. Some of them include the *TASS* workshop in the *SEPLN* conference (Villena-Román et al., 2013), the *RepLab* workshop in the *CLEF* conference (Amigó et al., 2012), and the *Sentiment Analysis in Twitter* task (Task 2) in the last *SemEval* workshop (Nakov et al., 2013).

In this paper we describe the system submitted for the *SemEval 2014 Sentiment Analysis in Twitter* task (Task 9 Subtask B)[2] (Rosenthal et al., 2014). This task consists of performing an automatic sentiment analysis to determine whether a message expresses a *positive*, *negative*, or *neutral* sentiment. The organisers of this task provide three datasets: *training*, *development training*, and *development test*. The participants can use the *training* and *development training* datasets to train and validate their models, but the *development test* dataset can only be used for validation. The size and distribution of polarities of these datasets is shown in Table 1. Once their systems are ready, the participants must classify each text in the *official test* corpus and send the results to the organisers, who will perform the official evaluation.

Polarity	Train	Dev Train	Dev Test
Positive	2,148	362	1,572
Neutral	2,915	448	1,640
Negative	836	187	601
Total	5,899	997	3,813

Table 1: Dataset distribution in number of tweets.

The goal of the present work is to create a reliable polarity classifier, built only from a training set without any external knowledge and resources.

This work is licensed under a Creative Commons Attribution 4.0 International Licence. Page numbers and proceedings footer are added by the organisers. Licence details: creativecommons.org/licenses/by/4.0/

[1] http://twitter.com

[2] http://alt.qcri.org/semeval2014/

Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pages 294–299,
Dublin, Ireland, August 23-24, 2014.

Our contribution consists of a supervised approach using machine learning techniques, which uses the terms in the dataset as features. The novelty of our approach lies in the feature generation and weighting, using not only *single words* and *ngrams* as features but also *skipgrams*. This approach is described in detail in Section 3. Subsequently, in Section 4 we show the assessment of our model in the competition. Finally, the conclusions and future work are presented in Section 5. The following Section 2 shows some relevant background related to this work.

2 Related Work

The goal of *Sentiment analysis* (SA) is to identify the opinions expressed in text and classify texts accordingly (Dadvar et al., 2011). Two main approaches can be followed (Annett and Kondrak, 2008; Liu, 2010; Taboada et al., 2011): *lexical* approaches (unsupervised SA) and *machine learning* approaches (supervised SA). Lexical approaches focus on building dictionaries and lexicons of labelled words. This labeling gives a score for each word, that indicates how strong is the relation between that word and each polarity. The most common way to classify a text using these scores is by adding the positive values and subtracting the negative values of the terms in that text. If the total score is positive, that text is classified as positive, otherwise it is classified as negative. These dictionaries can be created manually (Stone et al., 1966) or automatically (Turney, 2002). Examples of lexicons are *WordNet Affect* (Strapparava and Valitutti, 2004), *SentiWordNet* (Esuli and Sebastiani, 2006), *MicroWNOP* (Cerini et al., 2007) or *JRC Tonality* (Balahur et al., 2009). However, it is very difficult to collect and maintain a universal sentiment lexicon because different words may be used in different domains (Qiu et al., 2009) and some words are domain dependent (Turney, 2002).

The second approach uses machine learning techniques. These techniques require the previous creation of a corpus containing a set of classified texts to train a classifier, which can then be applied to classify a set of unclassified texts. The majority of the researches employ *Support Vector Machines* (Mullen and Collier, 2004; Prabowo et al., 2009; Wilson et al., 2005) or *Naïve Bayes* (Pang and Lee, 2004; Wiebe et al., 2005; Tan et al., 2009) classifiers because they usually obtain the best results. In this approach, texts are represented as vectors of features, and depending on the features used the system can reach better results (bag-of-words and lexeme-based features are the more commonly used (Pang and Lee, 2008)). These classifiers perform very well in the domain that they are trained on, but their performance drops when the same classifier is used in a different domain (Pang and Lee, 2008; Tan et al., 2009).

The problem of the *domain dependence* is common to both approaches. When the lexicons and classifiers generated are used in a domain different from the one they were built for, they use to perform worse (Turney, 2002; Pang and Lee, 2008; Qiu et al., 2009; Tan et al., 2009). Creating a domain-specific lexicon or classifier means making a manual effort. Although some studies try to overcome this problem by generating the lexicons *automatically* (Turney, 2002), learning from *unannotated* texts (Wiebe et al., 2005) or using *hybrid* approaches (Andreevskaia and Bergler, 2008; Bollen et al., 2011; Zhang and Ye, 2008), a minimal intervention from experts in the domain is needed. In this study we use the machine learning approach due to the promising results obtained in previous works (Boldrini et al., 2009; Fernández et al., 2011; Fernández et al., 2013).

3 Methodology

Our contribution consists of a supervised approach using machine learning techniques, which uses the terms in the dataset as features. In summary, our approach starts making a basic normalisation of each tweet in the dataset (see Section 3.1). Next, these texts are tokenised to extract their terms, and these terms are combined to create *skipgrams* (see Section 3.2). Finally, these skipgrams are employed as features for a supervised machine learning algorithm (see Section 3.3).

3.1 Basic normalisation

We perform a very basic normalisation, as we do not want to lose the potential subjective information given by the not normalised original text. Each tweet in the dataset is normalised following these steps:

1. **Lower case conversion**. All the characters in the tweet text are converted to lower case.

2. **Character repetition removal**. If the same character is repeated more than 3 times, the rest of repetitions are removed, so we can

still recognize if a word had repeated characters. For example, the words *gooood* and *gooooood* would be normalised to *goood*, but the word *good* would remain the same. We assume the ambiguity of some words like the one in the example, which can refer to the words *good* and *god*.

3. **Usernames and hashtags substitution**. We do not consider usernames and hashtags as they are not usually the words that represent a subjective sentence, they use to be the *topic* of the tweet. They are not removed completely but they are replaced by the strings `USERNAME` and `HASHTAG`.

So excited to go to #Alicante tomorrow with the best friend everrrrr @John!!!!
↓
so excited to go to #alicante tomorrow with the best friend everrrrr @john!!!!
↓
so excited to go to #alicante tomorrow with the best friend everrr @john!!!
↓
so excited to go to HASHTAG tomorrow with the best friend everrr USERNAME!!!

Figure 1: Example of normalisation process.

3.2 Tokenisation

Once we have normalised the texts, we extract all their terms. In this work, we consider a term as a group of adjacent characters of the same type (letters, numbers or punctuation symbols). For example, the text *want2go!!* would be tokenised to the terms *want*, *2*, *go*, and *!!*. Note that we employ all the terms extracted, not filtering any of them.

Finally, we obtain the *skipgrams* of the terms in the text. Skipgrams are a technique largely used in the field of speech processing, whereby n-grams are formed (bigrams, trigrams, etc.) but in addition to allowing adjacent sequences of words, it also allows tokens to be *skipped* (Guthrie et al., 2006). More specifically, in a *k-skip-n-gram*, *n* determines the maximum number of terms, and *k* the maximum number of skips allowed. In this way skipgrams are new terms that retain part of the sequentiality of the terms, but in a more flexible way than ngrams. Note that a ngram can be described as a skipgram where $k = 0$. An example is shown in Figure 2.

Normalised tweet
so excited to go to HASHTAG tomorrow with the best friend everrr USERNAME!!!
↓
Single terms
(so) (excited) (to) (go) (to) (HASHTAG) (with) (the) (best) (friend) (everrr) (USERNAME) (!!!)
↓
Skipgrams ($n = 2, k = 1$)
(so) (so excited) (so to) (excited) (excited to) (excited go) (to) (to go) (to to) (go) (go to) (go HASHTAG) (to) (to HASHTAG) (to with) (HASHTAG) (HASHTAG with) (HASHTAG the) (with) (with the) (with best) (the) (the best) (the friend) (best) (best friend) (best everrr) (friend) (friend everrr) (friend USERNAME) (everrr) (everrr USERNAME) (everrr !!!) (USERNAME) (USERNAME !!!)

Figure 2: Example of tokenisation process.

3.3 Supervised Learning

To build our model we employed *Support Vector Machines* (SVM) as the supervised machine learning algorithm, as it has been proved to be effective on text categorisation tasks and robust on large feature spaces (Sebastiani, 2002; Mohammad et al., 2013). More specifically, we used the *Weka*[3] (Hall et al., 2009) *LibSVM* (Chang and Lin, 2011) implementation with the default parameters (*linear kernel*, $C = 1$, $\epsilon = 0.1$).

The skipgrams extracted in the previous step are employed as features for the SVM. The weight of each feature in each text will be calculated depending on the skipgram it represents, using the formula in Equation 1.

$$w(s,t) = \frac{terms(s)}{terms(s) + skips(s,t)} \quad (1)$$

Where $w(s,t)$ represents the weight of the skipgram s in the text t, $terms$ is a function that returns the number of terms in skipgram s, and $skips$ is a function that returns the number of skips of the skipgram s in the text t. This formula gives more importance to the skipgrams with a lower number of skips. In the example of the Figure 2, the skipgram *best friend* would have a weight of $2/(2+0) = 1$, while skipgram *best everrr* would have a weight of $2/(2+1) = 0.66$.

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

	Parameters	P	R	F1	Score
Baseline		0.630	0.604	0.580	0.447
Words	$n = 1$	0.611	0.612	0.604	0.530
Ngrams	$n = 2$	0.617	0.620	0.618	0.557
	$n = 3$	0.620	0.621	0.621	0.564
	$n = 4$	0.620	0.621	0.620	0.565
	$n = max$	0.621	0.622	0.621	0.566
Skipgrams	$n = 2, k = 1$	0.623	**0.625**	0.624	0.571
	$n = 2, k = 2$	0.626	0.624	**0.626**	0.572
	$n = 2, k = max$	0.627	0.624	0.625	**0.575**
	$n = 3, k = 1$	0.620	0.616	0.617	0.566
	$n = 3, k = 2$	0.625	0.614	0.618	0.564
	$n = 3, k = max$	**0.636**	0.588	0.599	0.544

Table 2: Experiments performed and scores obtained.

4 Evaluation

We performed a series of experiments employing both the *training* corpus and the *development training* corpus to create our model, and the *development test* corpus to validate it. We used as baseline the system presented to the workshop *TASS 2012* (Fernández et al., 2013), which also uses skipgrams and scores them depending on their density but, instead of using the skipgrams as features of a machine learning model, the polarity of each text is determined by a combination of the scores of its skipgrams.

The results of our experiments are shown in Table 2. In this table we show the *weighted precision* (**P**), the *weighted recall* (**R**), the *weighted F-score* (**F1**) and the *score* obtained using the scorer tool provided by the workshop organisers (**Score**). The notation $n = max$ indicates there was no limit with the number of terms, and $k = max$ indicates there was no restriction with the number of skips. As we can see, the presented approach outperforms the baseline proposed and the best results are obtained using skipgrams, specifically when $n = 2$ and $k = max$. These are the parameters of the system submitted to the competition.

Our main observation is that incrementing the number of terms increases the precision of the system. A possible explanation for this might be that ngrams/skipgrams with a greater number of words are more specific and representative of a given polarity. In addition, using skipgrams insted of ngrams also improves the precision. However, no significant differences were found between experiments with a different number of skips.

In Table 3 we can see the official results obtained in the SemEval 2014 competition. The best rank was obtained in the experiments with the *Twitter 2014 Sarcasm* dataset.

Dataset	Rank	Score
Live Journal	34	0.573
SMS 2013	35	0.466
Twitter 2013	28	0.575
Twitter 2014	30	0.561
Twitter 2014 Sarcasm	8	0.539

Table 3: Official SemEval 2014 Subtask B results.

5 Conclusions

In this paper we described the system submitted for the *SemEval 2014 Task 9 (Sentiment Analysis in Twitter)*. It consists of a supervised approach using machine learning techniques, without employing any external knowledge and resources. The novelty of our approach lies in the feature generation and weighting, using not only single words and *ngrams* as features but also *skipgrams*. In the experiments performed we showed that employing skipgrams instead of single words or ngrams improves the results for these datasets. This fact suggests that our approach is promising and encourages us to continue with our research.

As future work, we plan to find new methods to combine the weights of the skipgrams, evaluate our approaches on different corpora and different domains (in order to check their robustness), and start adding external knowledge and resources.

Acknowledgements

This research work has been partially funded by the *University of Alicante*, *Generalitat Valenciana*, *Spanish Government* and the *European Commission* through the projects, "Tratamiento inteligente de la información para la ayuda a la toma de decisiones" (GRE12-44), *ATTOS* (TIN2012-38536-C03-03), *LEGOLANG* (TIN2012-31224), *SAM* (FP7-611312), *FIRST* (FP7-287607) and *ACOMP*/2013/067.

References

Enrique Amigó, Adolfo Corujo, Julio Gonzalo, Edgar Meij, and Maarten de Rijke. 2012. Overview of RepLab 2012: Evaluating Online Reputation Management Systems. In *Conference and Labs of the Evaluation Forum (CLEF 2012)*.

Alina Andreevskaia and Sabine Bergler. 2008. When Specialists and Generalists Work Together: Overcoming Domain Dependence in Sentiment Tagging. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies (ACL HLT 2008)*, pages 290–298.

Michelle Annett and Grzegorz Kondrak. 2008. A Comparison of Sentiment Analysis Techniques: Polarizing Movie Blogs. In *Proceedings of the 21st Canadian Conference on Artificial Intelligence (CCAI 2008)*, pages 25–35.

Alexandra Balahur, Ralf Steinberger, Erik Van Der Goot, Bruno Pouliquen, and Mijail Kabadjov. 2009. Opinion Mining on Newspaper Quotations. In *2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, pages 523–526.

Ester Boldrini, Javier Fernández Martínez, José Manuel Gómez Soriano, Patricio Martínez Barco, et al. 2009. Machine learning techniques for automatic opinion detection in non-traditional textual genres.

Johan Bollen, Alberto Pepe, and Huina Mao. 2011. Modeling Public Mood and Emotion: Twitter Sentiment and Socio-Economic Phenomena. In *Fifth International AAAI Conference on Weblogs and Social Media (ICWSM 2011)*.

Sabrina Cerini, Valentina Compagnoni, Alice Demontis, Maicol Formentelli, and G Gandini. 2007. Micro-WNOp: A Gold Standard for the Evaluation of Automatically Compiled Lexical Resources for Opinion Mining. *Language resources and linguistic theory: Typology, second language acquisition, English linguistics*, pages 200–210.

Chih-chung Chang and Chih-jen Lin. 2011. LIBSVM : A Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2:1–39.

Maral Dadvar, Claudia Hauff, and FMG De Jong. 2011. Scope of Negation Detection in Sentiment Analysis. In *Proceedings of the Dutch-Belgian Information Retrieval Workshop (DIR 2011)*, pages 16–20.

Andrea Esuli and Fabrizio Sebastiani. 2006. Sentiwordnet: A Publicly Available Lexical Resource for Opinion Mining. In *Proceedings of LREC*, volume 6, pages 417–422.

Javi Fernández, Ester Boldrini, José M. Gómez, and Patricio Martínez-Barco. 2011. Evaluating EmotiBlog Robustness for Sentiment Analysis Tasks. In *Natural Language Processing and Information Systems*, pages 290–294.

Javi Fernández, Yoan Gutiérrez, José M. Gómez, Patricio Martínez-Barco, Andrés Montoyo, and Rafael Muñoz. 2013. Sentiment Analysis of Spanish Tweets Using a Ranking Algorithm and Skipgrams. In *XXIX Congreso de la Sociedad Española de Procesamiento de Lenguaje Natural (SEPLN 2013)*, pages 133–142.

David Guthrie, Ben Allison, Wei Liu, Louise Guthrie, and Yorick Wilks. 2006. A Closer Look at Skip-gram Modelling. In *5th international Conference on Language Resources and Evaluation (LREC 2006)*, pages 1–4.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. 2009. The WEKA Data Mining Software: an Update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18.

Bing Liu. 2010. Sentiment Analysis and Subjectivity. In *Handbook of Natural Language Processing*, pages 1–38.

Saif M. Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. 2013. NRC-Canada: Building the State-of-the-Art in Sentiment Analysis of Tweets. In *Proceedings of the International Workshop on Semantic Evaluation (SemEval-2013)*.

Tony Mullen and Nigel Collier. 2004. Sentiment Analysis using Support Vector Machines with Diverse Information Sources. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)*, pages 412–418.

Preslav Nakov, Sara Rosenthal, Alan Ritter, and Theresa Wilson. 2013. SemEval-2013 Task 2: Sentiment Analysis in Twitter. In *Proceedings of the International Workshop on Semantic Evaluation (SemEval-2013)*, volume 2, pages 312–320.

Bo Pang and Lillian Lee. 2004. A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. In *Proceedings of the 42nd annual meeting on Association for Computational Linguistics (ACL 2004)*, page 271.

Bo Pang and Lillian Lee. 2008. Opinion Mining and Sentiment Analysis. *Foundations and Trends in Information Retrieval*, 2(1–2):1–135.

Rudy Prabowo, Mike Thelwall, and Wulfruna Street. 2009. Sentiment Analysis: A Combined Approach. *Journal of Informetrics*, 3:143–157.

Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. 2009. Expanding Domain Sentiment Lexicon through Double Propagation. In *Proceedings of the 21st international Joint Conference on Artificial Intelligence (IJCAI 2009)*, pages 1199–1204.

Sara Rosenthal, Preslav Nakov, Alan Ritter, and Veselin Stoyanov. 2014. SemEval-2014 Task 9: Sentiment Analysis in Twitter. In *Proceedings of the International Workshop on Semantic Evaluation (SemEval-2014)*.

Fabrizio Sebastiani. 2002. Machine Learning in Automated Text Categorization. *ACM Computing Surveys (CSUR)*, 34(1):1–47, March.

Philip J. Stone, Dexter C. Dunphy, and Marshall S. Smith. 1966. The General Inquirer: A Computer Approach to Content Analysis.

Carlo Strapparava and Alessandro Valitutti. 2004. WordNet Affect: an Affective Extension of WordNet. In *LREC*, volume 4, pages 1083–1086.

Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. 2011. Lexicon-based methods for sentiment analysis. *Computational Linguistics*, 37(2):267–307.

Songbo Tan, Xueqi Cheng, Yuefen Wang, and Hongbo Xu. 2009. Adapting Naive Bayes to Domain Adaptation for Sentiment Analysis. *Advances in Information Retrieval*, pages 337–349.

Peter D. Turney. 2002. Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL 2002)*, pages 417–424.

Julio Villena-Román, Eugenio Martínez-Cámara, Sara Lana-Serrano, and José Carlos González-Cristóbal. 2013. TASS - Workshop on Sentiment Analysis at SEPLN. *Procesamiento del Lenguaje Natural*, 50:37–44.

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating Expressions of Opinions and Emotions in Language. *Language resources and evaluation*, 39(2-3):165–210.

Theresa Wilson, Paul Hoffmann, Swapna Somasundaran, Jason Kessler, Janyce Wiebe, Yejin Choi, Claire Cardie, Ellen Riloff, and Siddharth Patwardhan. 2005. OpinionFinder: A System for Subjectivity Analysis. In *Proceedings of HLT/EMNLP on Interactive Demonstrations*, pages 34–35.

Min Zhang and Xingyao Ye. 2008. A Generation Model to Unify Topic Relevance and Lexicon-based Sentiment for Opinion Retrieval. In *Proceedings of the 31st annual international ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2008)*, pages 411–418, New York, New York, USA. ACM Press.