Comparing SVMs and TSVMs for Sentiment Analysis on Organized Crime

Guillermo Cabrera
Department of Computer Science
University of Texas at Austin
1 University Station B5100
Austin, TX 78712-0198 USA
gcabrera@cs.utexas.edu

Abstract

This paper explores sentiment analysis on blogs relating to the organized crime domain. We focus on this domain given that blog posts exhibit a great number of named entities and make use of a new lexicon. A semi-supervised learning approach will be taken in order to correctly classify polarity in blog posts; we use accuracy as the metric to evaluate the performance of our model and compare to other supervised learning approaches (SVM). We achieve a 78.8% accuracy when using a TSVM which is not breakthrough, but we have identified the source of some problems that will likely increase the accuracy percentage.

1 Introduction and motivation

Blogs have become an integral part of the Internet user experience; users will normally visit and engage by expressing their opinion towards topics (blog posts). Many businesses, politicians and other entities have recognized the importance of gathering this kind of information; which is why sentiment analysis has gained popularity in the past years.

Sentiment analysis has been applied to various domains; however, we have not come across any work on organized crime. We believe this domain has unique properties that sets it apart from other domains and makes it an interesting problem to investigate. As a sample scenario, let us consider the current situation in Mexico, where a dramatic increase in violence due to organized crime activity has led to self-censorship. As a result, people are now turning to blogs as their source of information and sites such as blogdelnarco.com have attracted as many as three million hits per week. Individuals expressing their opinions on blog posts include: Civilians, police and criminals. All these entities interact in the blog by providing very strong opinions and many do so by making heavy use of profanity, colloquial language and other Internet jargon.

In this work, we study the sentiment associated with entities such as: Army, police or other governmental institutions within the organized crime domain. As mentioned above, we claim that blog activity is significantly different than other domains. We have used a semi-supervised approach by means of a Transductive SVM (TSVM) [1] model to serve in sentiment classification. A TSVM can be thought of as simply a SVM with a transductive learning procedure, all this means is that a TSVM initially trains a SVM with labeled data. Then, it uses unlabeled data to retrain the original model, and does this for some steps until the unlabeled data has had an influence on the model. As a result of our study, we report on the accuracy of our model and compare that with existing supervised methods (SVMs).

In the past, SVM's have been demonstrated to outperform Naive Bayes or maximum entropy classification in terms of relative performance Pang et al. (2002). However, in our problem we have mostly noisy unlabeled data that we would like to use to train our TSVM, as well as exploit co-occurring patterns in text. Therefore, a supervised approach is not an affordable option for our problem. Furthermore, given the heavy use of colloquial language, made up

[1] Also known as a Semi Supervised Support Vector Machine (S3VM)

verbs and nouns (ex. *troka* as a way to say truck, instead of the Spanish word *camioneta*), translating into English as is done in Denecke (2008) would not give us good results. Consider the case of the following two sample comments extracted from our dataset:

(1) a la bio a la bao a la bim bom ba, los federales, ra ra ra / hooray for the federales

(2) la p****e pfp solo viene a robar, matar y extorsionar, fuera / the f**** pfp only comes to steal, kill and extort, out

An individual with knowledge of Spanish language and culture in Mexico might find it trivial to assign a positive label to (1) and a negative to (2), and know that both refer to the same entity (federal police). However, a machine might find it difficult given the heavy use of sarcasm, humor, intense use of profanity, multiple aliases for named entities and the inclusion of a new lexicon. From a government perspective, the importance of having a list of entities with their associated sentiment would be important as it would give an insight into the feeling towards police or army extortion. One could also imagine that sentiment could be tracked over time or even space Bautin et al. (2010) for criminal groups or their members, allowing intelligence agencies to analyze possible expansion of certain organized crime groups.

We have collected over 300 MB in comments from 455 blog posts in blogdelnarco.com, the most popular blog amongst the other four most active blogs. Collecting the data posed a difficult task as the DISQUS[2] commenting system used by the blog does not allow extraction of comments, unless you are the blog owner. To overcome this problem we have used the Blogger API in combination with the iMacro plug-in for the Firefox web browser to scrape all of the comments. These posts range from mid October back to May 2010 and are now publicly available[3].

2 Related Work

A majority of the efforts in sentiment analysis in blogs takes mainly one of two forms, the first one is a knowledge based approach, where a dictionary determines the polarity for words and is then matched to the dataset. The other approach makes use of machine learning techniques where a classifier is used and is fed labeled instances as training data.

The work in Godbole et al. (2007) makes use of a knowledge based approach; lexicons are automatically generated for blogs in various sub domains including crime, then, this marked dictionary is correlated with the corpus in an effort to apply polarity labels to words. On the other hand, Melville et al. (2009) takes a supervised learning approach and constructs a generative model from a polarity annotated lexicon and then builds a model trained on labeled documents.

A similar approach makes use of semi supervised learning. The task of blog classification in Ikeda et al. (2009) tries to identify blogger's gender and age from a limited set of labeled data and a great number of unlabeled instances. It makes use of sub classifiers and trains them for different domains. The output of the classifiers give a weight which can be compared with the output of other classifiers to establish similarity between blogs in order to apply the correct label.

Furthermore, it is worth mentioning some of the problems inherent with sentiment analysis and blogs. Chen and Lin (2010) argues the importance of the class imbalance problem by stating that in blogs there are far fewer instances that will be negative. If there are a greater number of positive instances, then the classifier is highly likely to produce positive labels for the unlabeled instances. Also, when dealing with sentiment analysis in another language, there has been work where machine translation is used to convert text into English, such as the work from Bautin et al. (2008) where they explore the analysis of international news for blogs in nine languages. They mention that although the translation process has some negative effect, this was not a significant issue in their experiments.

3 Approach

Our problem involves determining polarity for entities, yet, we have mentioned that our data set is considerably noisy. As a first step we have cleaned the text by removing diacritical marks and case fold-

[2] http://www.disqus.com

[3] http://www.cs.utexas.edu/ gcabrera/data.zip

ing tokens. We then removed stop words (total of 249) and split the comments that had paragraphs into separate comments. There were cases were a user would write four or five paragraphs as a comment and other instances where a comment would simply be a couple of words. The idea behind breaking long comments by paragraph was done on the assumption that it would help create more cohesive comments; in following sections we see how our assumption was not entirely valid.

The above steps were performed in an effort to reduce the vocabulary for our feature vectors and remove non-contributing words in determining polarity. Thereafter, we focused on filtering out any comments that did not contain entities we were tracking, for our case we mean those not referring to "Felipe Calderon" (current President of Mexico). To achieve this, we initially considered employing a named entity recognition (NER) tool such as the Illinois Named Entity Tagger (Ratinov and Roth, 2009), which compared to others makes use of a user- supplied gazetteer to increase matches. However, it is normally the case that NER systems rely on part of speech tagging or proper casing to correctly extract entities. But as we described earlier for this particular domain users normally fail at making correct use of upper case letters for names. Furthermore, grammar and orthography make it hard to assign part of speech tags to our Spanish corpus. We decided to use a simple dictionary lookup of entities from a built ontology of names, synonyms and nicknames given to the President. We are aware that such approach would give us a low number of matches, for instance (Ratinov and Roth, 2009) mentions how this approach gave a 71.91 F score for a particular test set, a score well below the 90.8 F score that the Illinois NER system claims to get.

Despite the potential low number of entity matches, we are confident that by having an extensive ontology we have compensated and obtained a decent amount of comments referring to the President. With this set of messages we then used other human annotators to label a subset of these messages. For labeling purposes, human annotators were presented with the title of the blog post, a one paragraph summary of the news item and comments containing a reference to the President and published under that blog post. This helped annotators with the context in which a comment was posted.

Finally, as for feature selection, we followed the approach from Gamon (2004) and Pang et al. (2002) where we capture the absence or presence of features and do not consider the TF nor TF*IFD. Both report better results when using this approach on unigram features as opposed to bigram features. Thereafter, SVM light [4], an existing implementation of SVM libraries that include support for TSVMs will be used to train a model and classify instances.

4 Implementation

Our overall goal is to show a comparison between an SVM and TSVM approach when dealing with mostly unlabeled data in a rarely seen domain. In addition we wish to show how a TSVM implementation is a better option in terms of the ratio of needed labeled instances to accuracy. For this project we employed several tools for different tasks, as an example, regular expressions were heavily used to parse and extract text from comments. Java was used to clean the text as well as create a vocabulary and feature vectors needed for the SVM Light Java library. Although minimal use, we did use the linguistics Hadoop cluster to run basic word count for all of our comments.

4.1 Gathered Data

We first started by grouping our 455 files (one per blog post) by months, since one of our goals is to plot sentiment on entities over time. Then, we started cleaning our data set, leaving only the text value for comments. The scope of our study does not involve usernames or other metadata associated with comments. For this reason we only use the text value. The text retrieval task was assumed to be trivial, yet, it posed some issues during parsing. The HTML was not a well formed XML document; this made us heavily rely on the use of regular expressions.

Through regular expressions we were able to leave single comments per lines, we decided not to further divide these comments by sentences based on periods as this could introduce further complications. As an example consider the case where users include URLS in their messages, "http://www"

[4] http://www.cs.cornell.edu/People/tj

could potentially be a sentence, which makes no sense. Similarly, we were able to detect users writing acronyms with periods, so "PFP" would be written as "P.F.P". The cleaning process gave us a total of 130,091 comments, the elimination of stop words cut by half the number of tokens in our data set. We refer to this set of comments as the "Clean Comments", further breakdown of the set is provided in the table below.

Category	Quantity
Words in vocabulary after Normalizing	164,615
Comments	130, 091
Tokens	4,521,182
Tokens after dropping stop words	2,212,212
Words in vocabulary after removal of low freq.	30,146

Figure 1: Data set summary

From Table 1 we see that we reduced the number of features by a factor of five with the discharge of tokens with low frequency in our vocabulary. We followed the approach in Annett and Kondrak (2008) and dropped any token with 4 or less repetitions. Regardless of this action, our vocabulary still includes many tokens; after manual inspection we see a great amount of numeric tokens (ex. phone numbers, dates) and many tokens sharing the same stem. Also, as a comparison it is worth nothing that the work in Pang et al. (2002) ends up with 16,000 dimensions where we currently stand at over 30,000, amount we would like to decrease in the future.

4.2 Entity Recognition

For this project, we are currently working with a single entity (President of Mexico). We extracted common names that individuals use to refer to the President and used them as a reference when extracting comments where the President is mentioned. We collected 15 terms that could be used as aliases. It is worth mentioning that we included some derogatory aliases which already embed a negative sentiment associated with the entity. We will later discuss how this was a bad idea to consider and talk about the negative effects in accuracy of our TSVM model.

In the end we had 6,163 comments mentioning the President which we refer to as "Entity Comments". These comments were separated by files, so that each file contained only the comments posted for a particular month. This information was then used to create a chart as seen in Figure 1 with the level of activity in terms of comments towards the president (positive, negative or neutral). It is interesting to note that the rise in August could be related to two major news happening at that time: A famous drug lord 'La Barbie" was captured, and at this time there was also the incident involving 72 dead South American immigrants assassinated. We noticed that there were some problems with this initial collection. For instance some of the comments did contain the token "presidente", yet, it preceded the token "municipal". Thus, people were referring to a "presidente municipal" (city mayor) and not the President of Mexico. Similarly, we removed two other aliases ("chaparro" and "enano") that were commonly used to refer to other entities.

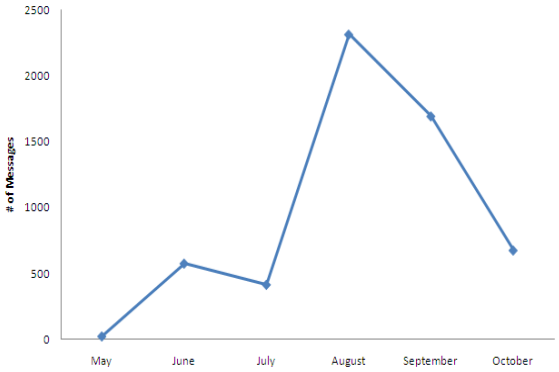

Figure 2: Message Distribution on President

Furthermore, out of the "Entity Comments" we labeled a total of 420, as positive or negative and discarded any comments that seemed neutral. The constitution of these comments was 102 as positive and 318 as negative. It becomes difficult to label X number of instances, as it turns out that even if the President is mentioned, sentiment might not be present and be neutral. For instance, we came across comments that list various unrelated keywords or others that do mention the President but remain neutral in their comments and provide a very generic opinion.

4.3 Data Observations

Stemming was not applied on our data given that its use has been a matter of debate when applied to the Spanish language (Figuerola et al., 2002). However, inspection of our vocabulary has shown cases where stemming could greatly reduce our vocabulary size. As an example consider the verb "agarrar" (to grab/to catch), it appears as 38 different forms (including those with orthographic errors) which could be summarized by a single token.

4.4 SVM light

We chose to use the SVM light package for Java; we first developed classes that would read our vocabulary files and create an in-memory representation as a hash-table. This table was then used to create our feature vectors. We start by reading a comment from the "Entity Comments" file and then looking up all the tokens in the comment against the table. We add a 1 for a feature value indicating the feature is present in the comment and ignore any absent features. As for labels, SVM light requires us to assign a "-1" to any instance which has been labeled negative and "+1" if we consider that instance to be part of the other class. A "0" is added to those instances for which we do not know the class to which they belong and also indicates that the instances is to be used by the TSVM model for retraining.

After reading all comments from a file and creating a feature vector out of these comments we then train and classify instances based on a 10-fold cross validation technique that will compare accuracy of an SVM and TSVM model. As in other literature, we define accuracy to be:

$$accuracy = \frac{(tp + tn)}{\text{total num of tested instances}}$$

Where tp is the number true positives and tn is the number of true negatives.

5 Experiments

We use the extracted data mentioned above as our test corpus. Our main hypothesis to test was: The use of a TSVM for sentiment classification will produce higher accuracy when compared with a SVM. In order to test this hypothesis we designed the two following experiments:

1. SVM Classification: To develop a baseline to test against we start by taking four different files as input to train and classify. From the 420 labeled comments we first begin by shuffling the order of the comments, as we don't want our model to train on comments specific to a certain month. Then we created the four input files, each one with 105, 210, 315 and 420 comments respectively. Then, a 10-fold cross validation approach was taken; during this validation procedure it is important to note that our SVM was trained on all of the labeled instances for each of the 9 partitions of each fold. For instance in the case of the file with the 420 labeled instances, 378 (9 parts of 42 instances) instances were used for training and the remaining 42 were used for validation (testing whether the predictions of the trained SVM were correct).

2. TSVM Classification: This experiment was slightly different because of the nature of the transductive process (retraining from unlabeled instances). Since our goal is to compare against the results obtained by the SVM classification, we initially proceed in the same way by also using four files with 105, 210, 315 and 420 labeled instances. For each of the files we decide to control the number of labeled instances that the TSVM will initially train with. So, we train using 20, 40, 60 and 80 percent of the labeled instances. Let us consider an example, again consider the file with 420 labeled instances. Out of the 378 labeled instances we will now use only 20 of them for initial training. Thus, 76 instances are used to train an initial model and the remaining 302 are used in the retraining process. We will then perform the same process but this time with 40% of those 378 labeled instances and so on. In the end this will result in 16 different training and classification runs for which we also use 10-fold cross validation.

For each of these experiments we take the average of the accuracy reported for each of the folds during validation and then pick the highest accuracy value for each one of the input files. This value is then compared for both approaches (SVM and TSVM). The graph produced will reflect the proportion of

accuracy with respect to the number of labeled instances. Thus we can see which model gives us better performance, where performance in this context is defined to be the lowest number of labeled instances for the highest accuracy possible. In other words, we are looking for the greatest difference between accuracy values for each of the four cases (various numbers of labeled instances for training).

Finally, as part of our last output we take the best performing model and use it to classify the remaining 5000+ unlabeled instances that make a reference to the President. Once these are classified we plot them in a graph so that we can easily observe the ratio of negative and positive comments.

6 Results

After running all the needed experiments we were made aware of the difference in time for both experiments 1 and 2. An SVM experiment would take less than one minute (including 10-fold cross validation) while a TSVM would take a little over two minutes. In Table 1 we can see the results from experiment 1, we can see how accuracy gets better as we include more labeled instances to train our model. Thus for an SVM to obtain 77.8% accuracy one needs to train with 378 instances for this specific corpus and domain. It is also worth nothing that the biggest jump in accuracy happens from 105 to 210 labeled instances (recall that even though we say 105 and 210, we really train on 95 and 189 instances, since we do 10-fold cross validation).

Table 1: SVM Accuracy

Labeled Instances	100 % for Training
105	67.7
210	74.2
315	76.8
420	77.8

For the TSVM, we noticed a significant improvement in accuracy even though we never used 100% of all training instances to train the initial TSVM model (at which point the TSVM would have been a simply SVM, since there would not be any unlabeled instances to take into account while training). In table 2 we have marked with bold the highest accuracies per number of labeled instances. So, for instance for the case where we use 105 instances for training and validation, 73.3% was the accuracy using only 60% as the initial number of labeled instances to train the TSVM model. The remaining 40% were unlabeled instances that were used in retraining.

Table 2: TSVM Accuracy

Labeled Instances	20%	40%	60%	80%
105	70.7	70.7	73.3	72.0
210	69.5	74.3	75.5	76.3
315	67.5	71.3	73.9	77.2
420	73.4	71.3	76.6	78.8

What do these results tell us? To begin with, one can definitely take advantage of unlabeled data. Furthermore, in our particular example we have plenty of unlabeled data, and we have seen how labeling this type of data becomes a tedious process. Also, we can use a far less number of labeled instances (compare the 73.3% at 60% for TSVM to the 67.7% at 100% for SVM) and still get better results using a TSVM.

We can see a better comparison of the best of both approaches in figure 3. It is a lot more clear how, the biggest gain happens with the 105 labeled instances case. For the other cases, the gain from a TSVM is minimal. This brings up the question of what will happen if we use even less labeled instances, what if we use only 50. Another factor to take into consideration when looking at this graph is the highest accuracy for both cases is between 78 and 79 percent. These are not breakthrough results but there are very promising. In Pang et al. (2002), the highest accuracy reported using presence of unigram features was 82.9% (when using a SVM). In our case, we have identified several errors that could be fixed and we expect these changes to boost the accuracy in a significant way. It was also reassuring to see that a TSVM performed better than an SVM as initially conjectured.

As a final output, we wanted to make it evident what type of sentiment was given to the chosen entity over time. In the figure 4, we plot the number of negative and positive comments for each of the months. There are several issues with this graph: First idea that comes to mind is the class imbalance

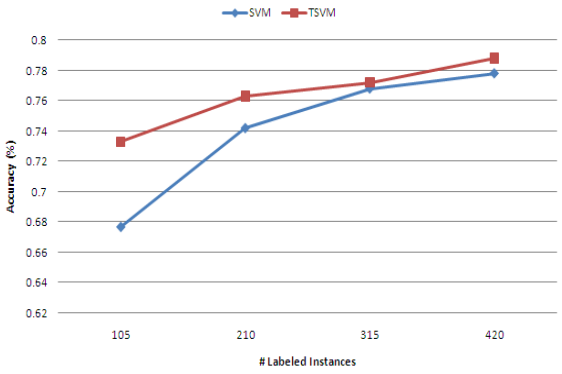

Figure 3: Accuracy comparison: SVM & TSVM

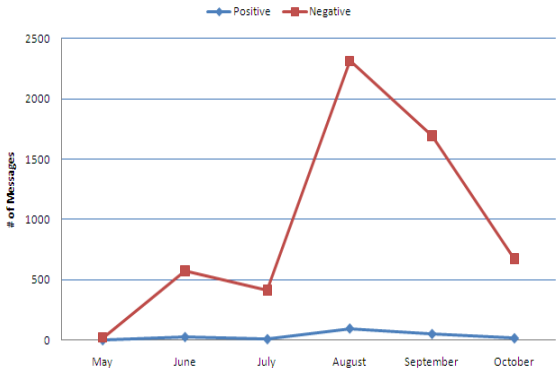

Figure 4: Sentiment Over Time for President

problem mentioned in Chen and Lin (2010). From previous sections we can recall that the constitution of comments in any of the cases (105, 210, etc.) is 75% negatively labeled messages and the rest are positive. This means that our model will be biased to label an unlabeled instance as negative with a higher probability. The second problem with this depiction of sentiment is the initial ontology we used to select messages that referenced the President, out of the 15 terms we selected; at least 7 of them were derogative aliases. This means that from this step we are already defining what the constitution of negative messages will be. Alternatively we could reduce the number of aliases to simply neutral terms, at the cost of less number of messages retrieved. Third and last, we were not very surprised that the majority of comments would be negative since users who visit these blogs tend to be civilians who are directly affected by all this violence and tend to be very angry towards their situation and the government.

7 Discussion

Our graph showing the summary of sentiment has yet another problem. In our approach or implementation we never considered neutral comments. The only time these were considered was during the validation phase for the experiments. To build the set of 420 labeled instances many other neutral comments had to be discarded. On the other hand while running the TSVM model to get predictions for the 5000+ comments we were forcing a comment to be either positive or negative (a bad practice, as there are always comments that are neutral, that if labeled as any of the other classes will start deteriorating the model as these are used for retraining).

One trivial solution without changing the approach would require a confidence value to be taken into account. This way, if such value is not higher than a user defined threshold we can discard that message. Therefore remaining only with labeled instances for which me know to be positive or negative. By not removing neutral messages we affected the accuracy for our model, and also the incorrect labeling for instances. Let us consider three examples and how they got labeled:

(3) BIEN VENDIDO EL CALDERON! bien vendido el EJERCITO MEXICANO! / Calderon corrupt! The Mexican army, corrupt!

(4) PD; Calderon (F***) C**** a toda tu p**** p*** p**** a***** y p***** larva m*** !! / PD; Calderon...

(5) FELIPE CALDERON HINOJOSA, ATENTAMENTE LE HAGO LLEGAR ESTE GRITO CON GRADO DE ATENCION. TOME EN CUENTA QUE NUESTRO MEXICO SE DESTROZA LENTAMENTE, QUE NECESITA APOYO. / Felipe Calderon Hinojosa, I cordially would like to exress my cry as a sign of attention. Take into consideration our Mexico which is slowly being destroyed and needs support.

Example (3) was misclassified and labeled as pos-

itive, at a first glance even I labeled as positive as there is a slight play on words that make it seem as the user wrote “Bienvenido” (welcomed), yet in this case the user has actually expressed a negative view on the President and the army by stating that both are very corrupt. I can understand that “bien” (good/well) is a word denoting positive sentiment, yet in this particular case, it is being used as a way to denote the degree of another event and not as an adjective on the President.

As for (4) which got correctly classified as negative, one can easily see that a great number of the terms in this comment are profanity words and helped to determine that the instance belonged in this class. Finally example (5) lets us see the problem with neutral comments. (5) was classified as positive, but if we read it carefully, the comment never projects a sentiment towards the President. The user in this case is just expressing his concern about the country. It is difficult to decide whether comments like these should be taken into consideration or if they should be discarded as mentioned earlier.

Apart from problems mentioned so far, we believe there are other minor adjustments that can be taken to improve our model and its accuracy. In our initial approach we split large comments into separate comments into separate ones (if they used paragraphs). We were afraid to take this approach further and consider a finer granularity (ex. sentences) because of reason previously mentioned. However, in an effort to improve the succinctness of comments we plan to make use of openNLP’s Sentence Detector [5] to accomplish this task.

Furthermore, we consider stemming would be beneficial given the multiple cases where terms in our vocabulary could with multiple forms can be simplified into a single feature, tools such as snowball or possible the Spanish version of Wordned (A. et al.) could help in this regard. Finally, to continue with the effort of dimensionality reduction, we plant to implement the compression idea in Pandey and Iyer (2009). The basic idea consists on creating the minimal shadow for a word that has consecutive repeated characters. Thus, a word like “ssslooooooww” would be represented as simply “slow”. We saw how this would be extremely important as we discovered 114 cases of users expressing laughter in their comments. As an example some users would use “jajaja” others would use “jajajaja” (difference of 2 more letters).

[5] http://sourceforge.net/apps/mediawiki/opennlp/

8 Conclusion

We introduced the problem with blogs in the organized crime domain, we also proposed that a TSVM would be the appropriate tool to do sentiment analysis on the comments to blog posts. As a form of evaluation we compared our TSVM with a SVM in order to see the gain from a TSVM when using a minimal number of labeled instances for training. The results in our work clearly show the advantage of following a semi-supervised learning approach by means of a TSVM. Yet, one may argue that training TSVMs is time consuming, in order to classify the 5000+ instances a dual core machine toke approximately three hours whereas a SVM took less than one minute to classify the same amount of instances.

Nevertheless, despite the time tradeoff, accuracy seems promising, especially now that we have identified the sources for potential improvements in our model. Our current results are not breakthrough, yet, it is reassuring to see that a TSVM indeed performs better than a SVM for text classification as suggested by the literature. At this point our work has proven to work and we are able to plot sentiment over time for any entity we may want to analyze. All that is needed is ontology for the specified entity and we can easily produce a sentiment graph. In coming weeks we plan to incorporate the suggestions given in this paper and report on whether a gain or loss was achieved by their implementation.

References

Fernandez-Montraveta A., Vazquez G., and Fellbaum C. The spanish version of wordnet 3.0. In *Text Resources and Lexical Knowledge*.

Michelle Annett and Grzeorz Kondrak. A comparison of sentiment analysis techniques: Polarizing movie blogs. In *Lecture Notes in Artificial Intelligence*, Berlin Heidelberg, 2008.

Mikhail Bautin, Lohit Vijayarenu, and Steven Skiena. International sentiment analysis for news and blogs. In *International AAAI Conference on Weblogs and Social Media*, 2008.

Mikhail Bautin, Charles B. Ward, Akshay Patil, and Steven S. Skiena. Access: News and blog analysis for the social sciences. In *Proceedings of the International World Wide Web Conference*, Raleigh, North Carolina, USA, 2010.

Long-Sheng Chen and Li-Wei Lin. Two methods for classifying bloggers's sentiment. In *Proceedings of the International Multi Conference of Engineers and Computer Science*, Hong Kong, 2010.

Kerstin Denecke. Using sentiwordnet for multilingual sentiment analysis. In *IEEE 24th International Conference on Data Engineering Workshop*, 2008.

Carlos Figuerola, Raquel Gomez, Angel Zazo Rodriguez, and Jose Luis Alonso Berrocal. Spanish monolingual track: The impact of stemming on retrieval. In *Evaluation of cross-language information retrieval systemsl*. Springer-Verlag, 2002.

Michael Gamon. Sentiment classification on customer feedback data: Noisy data, large feature vectors, and the role of linguistic analysis. In *Proceedings of the International Conference on Computational Linguistics*, 2004.

Namrata Godbole, Manjunath Srinivasaiah, and Steven Skiena. Largescale sentiment analysis for news and blogs. In *ICWSM*, 2007.

Daisuke Ikeda, Hiroya Takamura, and Mnabu Okumura. Semi-supervised learning for blog classification. In *Proceedings of Twenty Third AAAI Conference on Artificial Intelligence*, Hong Kong, 2009.

Prem Melville, Wojciech Gryc, and Richard D. Lawrence. Sentiment analysis of blogs by combining knowledge with text classification. In *ACM Knowledge Discovery and Data Mining*, 2009.

Vipul Pandey and C.V. Krishnakumar Iyer. Sentiment analysis of microblogs. In *Stanford Machine Learning Course CS229*, 2009.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up? sentiment classification using machine learning. In *Proceedings of EMNLP*, 2002.

Lev Ratinov and Dan Roth. Design challenges and misconceptions in named entity recognition. In *CoNLL*, 2009.