SAIL: A hybrid approach to sentiment analysis

Nikolaos Malandrakis[1], **Abe Kazemzadeh**[2], **Alexandros Potamianos**[3], **Shrikanth Narayanan**[1]
[1] Signal Analysis and Interpretation Laboratory (SAIL), USC, Los Angeles, CA 90089, USA
[2] Annenberg Innovation Laboratory (AIL), USC, Los Angeles, CA 90089, USA
[3]Department of ECE, Technical University of Crete, 73100 Chania, Greece
malandra@usc.edu, kazemzad@usc.edu, potam@telecom.tuc.gr, shri@sipi.usc.edu

Abstract

This paper describes our submission for SemEval2013 Task 2: Sentiment Analysis in Twitter. For the limited data condition we use a lexicon-based model. The model uses an affective lexicon automatically generated from a very large corpus of raw web data. Statistics are calculated over the word and bigram affective ratings and used as features of a Naive Bayes tree model. For the unconstrained data scenario we combine the lexicon-based model with a classifier built on maximum entropy language models and trained on a large external dataset. The two models are fused at the posterior level to produce a final output. The approach proved successful, reaching rankings of 9th and 4th in the twitter sentiment analysis constrained and unconstrained scenario respectively, despite using only lexical features.

1 Introduction

The analysis of the emotional content of text, is relevant to numerous natural language processing (NLP), web and multi-modal dialogue applications. To that end there has been a significant scientific effort towards tasks like product review analysis (Wiebe and Mihalcea, 2006; Hu and Liu, 2004), speech emotion extraction (Lee and Narayanan, 2005; Lee et al., 2002; Ang et al., 2002) and pure text word (Esuli and Sebastiani, 2006; Strapparava and Valitutti, 2004) and sentence (Turney and Littman, 2002; Turney and Littman, 2003) level emotion extraction.

The rise of social media in recent years has seen a shift in research focus towards them, particularly twitter. The large volume of text data available is particularly useful, since it allows the use of complex machine learning methods. Also important is the interest on the part of companies that are actively looking for ways to mine social media for opinions and attitudes towards them and their products. Similarly, in journalism there is interest in sentiment analysis for a way to process and report on the public opinion about current events (Petulla, 2013).

Analyzing emotion expressed in twitter borrows from other tasks related to affective analysis, but also presents unique challenges. One common issue is the breadth of content available in twitter: a more limited domain would make the task easier, however there are no such bounds. There is also a significant difference in the form of language used in tweets. The tone is informal and typographical and grammatical errors are very common, making even simple tasks, like Part-of-Speech tagging much harder. Features like hashtags and emoticons can also be helpful (Davidov et al., 2010).

This paper describes our submissions for SemEval 2013 task 2, subtask B, which deals primarily with sentiment analysis in twitter. For the constrained condition (using only the organizer-provided twitter sentences) we implemented a system based on the use of an affective lexicon and part-of-speech tag information, which has been shown relevant to the task (Pak and Paroubek, 2010). For the unconstrained condition (including external sources of twitter sentences) we combine the constrained model with a maximum entropy language

model trained on external data.

2 Experimental procedure

We use two separate models, one for the constrained condition and a combination for the unconstrained condition. Following are short descriptions.

2.1 Lexicon-based model

The method used for the constrained condition is based on an affective lexicon containing out-of-context affective ratings for all terms contained in each sentence. We use an automated algorithm of affective lexicon expansion based on the one presented in (Malandrakis et al., 2011), which in turn is an expansion of (Turney and Littman, 2002).

We assume that the continuous (in $[-1, 1]$) valence and arousal ratings of any term can be represented as a linear combination of its semantic similarities to a set of seed words and the affective ratings of these words, as follows:

$$\hat{v}(w_j) = a_0 + \sum_{i=1}^{N} a_i \; v(w_i) \; d_{ij}, \qquad (1)$$

where w_j is the term we mean to characterize, $w_1 ... w_N$ are the seed words, $v(w_i)$ is the valence rating for seed word w_i, a_i is the weight corresponding to seed word w_i (that is estimated as described next), d_{ij} is a measure of semantic similarity between w_i and w_j. For the purposes of this work, the semantic similarity metric is the cosine similarity between context vectors computed over a corpus of 116 million web snippets collected by posing one query for every word in the Aspell spellchecker's vocabulary to the Yahoo! search engine and collecting up to 500 of the top results.

Given a starting, manually annotated, lexicon we can select part of it to serve as seed words and then use 1 to create a system of linear equations where the only unknowns are the weights a_i. The system is solved using Least Squares Estimation. That provides us with an equation that can generate affective ratings for every term (not limited to words), as long as we can estimate the semantic similarity between it and the seed words.

Seed word selection is performed by a simple heuristic (though validated through experiments): we want seed words to have extreme affective ratings (maximum absolute value) and we want the set to be as closed to balanced as possible (sum of seed ratings equal to zero).

Given these term ratings, the next step is combining them through statistics. To do that we use simple statistics (mean, min, max) and group by part of speech tags. The results are statistics like "maximum valence among adjectives", "mean arousal among proper nouns" and "number of verbs and nouns". The dimensions used are: valence, absolute valence and arousal. The grouping factors are the 39 Penn treebank pos tags plus higher order tags (adjectives, verbs, nouns, adverbs and combinations of 2,3 and 4 of them). The statistics extracted are: mean, min, max, most extreme, sum, number, percentage of sentence coverage. In the case of bigram terms no part-of-speech filtering/grouping is applied. These statistics form the feature vectors.

Finally we perform feature selection on the massive set of candidates and use them to train a model. The model selected is a Naive Bayes tree, a tree with Naive Bayes classifiers on each leaf. The motivation comes by considering this a two stage problem: subjectivity detection and polarity classification, making a hierarchical model a natural choice. NB trees proved superior to other types of trees during our testing, presumably due to the smoothing of observation distributions.

2.2 N-gram language model

The method used for the unconstrained condition is based on a combination of the automatically expanded affective lexicon described in the previous section together with a bigram language model based on the work of (Wang et al., 2012), which uses a large set of twitter data from the U.S. 2012 Presidential election. As a part of the unconstrained system, we were able to leverage external annotated data apart from those provided by the SEMEVAL 2013 sentiment task dataset. Of the 315 million tweets we collected about the election, we annotated a subset of 40 thousand tweets using Amazon Mechanical Turk. The annotation labels that we used were "positive", "negative", "neutral", and "unsure", and additionally raters could mark tweets for sarcasm and humor. We excluded tweets marked as "unsure" as well as tweets that had disagree-

ment in labels if they were annotated by more than one annotator. To extract the bigram features, we used a twitter-specific tokenizer (Potts, 2011), which marked uniform resource locators (URLs), emoticons, and repeated characters, and which lowercased words that began with capital letters followed by lowercase letters (but left words in all capitals). The bigram features were computed as presence or absense in the tweet rather than counts due to the small number of words in tweets. The machine learning model used to classify the tweets was the Megam maximum entropy classifier (Daumé III, 2004) in the Natural Language Toolkit (NLTK) (Bird et al., 2009).

2.3 Fusion

The submitted system for the unconstrained condition leverages both the lexicon-based and bigram language models. Due to the very different nature of the models we opt to not fuse them at the feature level, using a late fusion scheme instead. Both partial models are probabilistic, therefore we can use their per-class posterior probabilities as features of a fusion model. The fusion model is a linear kernel SVM using six features, the three posteriors from each partial model, and trained on held out data.

3 Results

Following are results from our method, evaluated on the testing sets (of sms and twitter posts) of SemEval2013 task 2. We evaluate in terms of 3-class classification, polarity classification (positive vs. negative) and subjectivity detection (neutral vs. other). Results shown in terms of per category f-measure.

3.1 Constrained

The preprocessing required for the lexicon-based model is just part-of-speech tagging using Treetagger (Schmid, 1994). The lexicon expansion method is used to generate valence and arousal ratings for all words and ngrams in all datasets and the part of speech tags are used as grouping criteria to generate statistics. Finally, feature selection is performed using a correlation criterion (Hall, 1999) and the resulting feature set is used to train a Naive Bayes tree model. The feature selection and model training/classification was conducted using Weka (Witten and Frank, 2000).

Table 1: F-measure results for the lexicon-based model, using different machine learning methods, evaluated on the 3-class twitter testing data.

model	per-class F-measure		
	neg	neu	pos
Nbayes	0.494	0.652	0.614
SVM	0.369	0.677	0.583
CART	0.430	0.676	0.593
NBTree	0.561	0.662	0.643

Table 2: F-measure results for the constrained condition, evaluated on the testing data.

set	classes	per-class F-measure		
		neg	neu	pos/other
twitter	3-class	0.561	0.662	0.643
	pos vs neg	0.679		0.858
	neu vs other		0.685	0.699
sms	3-class	0.506	0.709	0.531
	pos vs neg	0.688		0.755
	neu vs other		0.730	0.628

The final model uses a total of 72 features, which can not be listed here due to space constraints. The vast majority of these features are necessary to detect the neutral category: positive-negative separation can be achieved with under 30 features.

One aspect of the model we felt worth investigating, was the type of model to be used. Using a multi-stage model, performing subjectivity detection before positive-negative classification, has been shown to provide an improvement, however single models have also been used extensively. We compared some popular models: Naive Bayes, linear kernel SVM, CART-trained tree and Naive Bayes tree, all using the same features, on the twitter part of the SemEval testing data. The results are shown in Table 1. The two Naive Bayes-based models proved significantly better, with NBTree being clearly the best model for these features.

Results from the submitted constrained model are shown in Table 2. Looking at the twitter data results and comparing the positive-negative vs the

3-class results, it appears the main weakness of this model is subjectivity detection, mostly on the neutral-negative side. It is not entirely clear to us whether that is an artifact of the model (the negative class has the lowest prior probability, thus may suffer compared to neutral) or of the more complex forms of negativity (sarcasm, irony) which we do not directly address. There is a definite drop in performance when using the same twitter-trained model on sms data, which we would not expect, given that the features used are not twitter-specific. We believe this gap is caused by lower part-of-speech tagger performance: visual inspection reveals the output on twitter data is fairly bad.

Overall this model ranked 9th out of 35 in the twitter set and 11th out of 28 in the sms set, among all constrained submissions.

3.2 Unconstrained

Table 3: F-measure results for the maximum entropy model with bigram features, evaluated on the testing data.

set	classes	per-class F-measure		
		neg	neu	pos/other
twitter	3-class	0.403	0.661	0.623
	pos vs neg	0.586		0.804
	neu vs other		0.661	0.704
sms	3-class	0.390	0.587	0.542
	pos vs neg	0.710		0.648
	neu vs other		0.587	0.641

Table 4: F-measure results for the unconstrained condition, evaluated on the testing data.

set	classes	per-class F-measure		
		neg	neu	pos/other
twitter	3-class	0.565	0.679	0.655
	pos vs neg	0.672		0.881
	neu vs other		0.667	0.732
sms	3-class	0.502	0.723	0.538
	pos vs neg	0.625		0.772
	neu vs other		0.710	0.637

In order to create the submitted unconstrained model we train an SVM model using the lexicon-based and bigram language model posterior probabilities as features. This fusion model is trained on held-out data (the development set of the SemEval data). The results of classification using the bigram language model alone are shown in Table 3 and the results from the final fused model are shown in Table 4. Looking at relative per-class performance, the results follow a form most similar to the constrained model, though there are gains in all cases. These gains are less significant when evaluated on the sms data, resulting in a fair drop in ranks: the bigram language model (expectedly) suffers more when moving to a different domain, since it uses words as features rather than the more abstract affective ratings used by the lexicon-based model. Also, because the external data used to train the bigram language model was from discussions of politics on Twitter, the subject matter also varied in terms of prior sentiment distribution in that the negative class was predominant in politics, which resulted in high recall but low precision for the negative class.

This model ranked 4th out of 16 in the twitter set and 7th out of 17 in the sms set, among all unconstrained submissions.

4 Conclusions

We presented a system of twitter sentiment analysis combining two approaches: a hierarchical model based on an affective lexicon and a language modeling approach, fused at the posterior level. The hierarchical lexicon-based model proved very successful despite using only n-gram affective ratings and part-of-speech information. The language model was not as good individually, but provided a noticeable improvement to the lexicon-based model. Overall the models achieved good performance, ranking 9th of 35 and 4th of 16 in the constrained and unconstrained twitter experiments respectively, despite using only lexical information.

Future work will focus on incorporating improved tokenization (including part-of-speech tagging), making better use of twitter-specific features like emoticons and hashtags, and performing affective lexicon generation on twitter data.

References

J. Ang, R. Dhillon, A. Krupski, E. Shriberg, and A. Stolcke. 2002. Prosody-based automatic detection of annoyance and frustration in human-computer dialog. In *Proc. ICSLP*, pages 2037–2040.

S. Bird, E. Klein, and E. Loper. 2009. *Natural Language Processing with Python*. O'Reilly Media.

H. Daumé III. 2004. Notes on cg and lm-bfgs optimization of logistic regression. *Paper available at http://pub. hal3. name# daume04cg-bfgs, implementation available at http://hal3. name/megam*.

D. Davidov, O. Tsur, and A. Rappoport. 2010. Enhanced sentiment learning using twitter hashtags and smileys. In *Proc. COLING*, pages 241–249.

A. Esuli and F. Sebastiani. 2006. Sentiwordnet: A publicly available lexical resource for opinion mining. In *Proc. LREC*, pages 417–422.

M. A. Hall. 1999. *Correlation-based feature selection for machine learning*. Ph.D. thesis, The University of Waikato.

M. Hu and B. Liu. 2004. Mining and summarizing customer reviews. In *Proc. SIGKDD*, KDD '04, pages 168–177. ACM.

C. M. Lee and S. Narayanan. 2005. Toward detecting emotions in spoken dialogs. *IEEE Transactions on Speech and Audio Processing*, 13(2):293–303.

C. M. Lee, S. Narayanan, and R. Pieraccini. 2002. Combining acoustic and language information for emotion recognition. In *Proc. ICSLP*, pages 873–876.

N. Malandrakis, A. Potamianos, E. Iosif, and S. Narayanan. 2011. Kernel models for affective lexicon creation. In *Proc. Interspeech*, pages 2977–2980.

A. Pak and P. Paroubek. 2010. Twitter as a corpus for sentiment analysis and opinion mining. In *Proc. LREC*, pages 1320–1326.

S. Petulla. 2013. Feelings, nothing more than feelings: The measured rise of sentiment analysis in journalism. *Neiman Journalism Lab*, January.

C. Potts. 2011. Sentiment symposium tutorial: Tokenizing. Technical report, Stanford Linguistics.

H. Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *Proc. International Conference on New Methods in Language Processing*, volume 12, pages 44–49.

C. Strapparava and A. Valitutti. 2004. WordNet-Affect: an affective extension of WordNet. In *Proc. LREC*, volume 4, pages 1083–1086.

P. Turney and M. L. Littman. 2002. Unsupervised Learning of Semantic Orientation from a Hundred-Billion-Word Corpus. Technical report ERC-1094 (NRC 44929). National Research Council of Canada.

P. Turney and M. L. Littman. 2003. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems*, 21:315–346.

H. Wang, D. Can, A. Kazemzadeh, F. Bar, and S. Narayanan. 2012. A system for real-time twitter sentiment analysis of 2012 u.s. presidential election cycle. In *Proc. ACL*, pages 115–120.

J. Wiebe and R. Mihalcea. 2006. Word sense and subjectivity. In *Proc. COLING/ACL*, pages 1065–1072.

Ian H. Witten and Eibe Frank. 2000. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann.