Improving Sentiment Analysis in an Online Cancer Survivor Community Using Dynamic Sentiment Lexicon

Nir Ofek[1,3], Cornelia Caragea[2], Lior Rokach[1,3], Prakhar Biyani[3], Prasenjit Mitra[3], John Yen[3],
Kenneth Portier[4], Greta Greer[4]
[1]Information System Engineering, Ben-Gurion University of the Negev, Be'er sheve, Israel
[2]Computer Science and Engineering, University of North Texas, Denton, TX, USA
[3]College of Information Sciences and Technology, Pennsylvania State University, University Park, PA, USA
[4]American Cancer Society, Inc., Atlanta, GA, USA
{nirofek, liorrk}@bgu.ac.il, ccaragea@unt.edu, pxb5080@psu.edu, {jyen,pmitra}@ist.psu.edu,
{greta.greer,kenneth.portier}@cancer.org

***Abstract*—Online Health Communities is a major source for patients and their family members in the process of gathering information and seeking social support. The American Cancer Society Cancer Survivors Network has many users and presents a large number of users' interactions with regards to coping with cancer. Sentiment analysis is an important step in understanding participants' needs and concerns and the impact of users' responses on other members. We present an automated approach for sentiment analysis in an online cancer survivor community and compare it with a previous sentiment analysis approach. Both approaches are machine learning based and are tested on the same dataset. However, this work uses features derived from a dynamic sentiment lexicon, whereas the previous work uses a general sentiment lexicon to extract features. Tested on several classifiers, with only six features (versus thirteen), our results show 2.3% improvement on average, in terms of accuracy, and greater improvement in F-measure and AUC. An additional experiment was conducted that showed a positive impact of dimensionality reduction by extracting abstract features, instead of using term frequency (TF) vector space as attribute values.**

Keywords—sentiment analysis; dynamic sentiment lexicon; abstract features.

I. INTRODUCTION

Many Internet users utilize Online Health Communities in order to obtain health-related information as well as to get social support through online social interactions [1]. People diagnosed with cancer as well as cancer caregivers, join the Cancer Survivors Network[1] of the American Cancer Society to seek social support and cancer-related information from others who have experienced a particular situation first hand and emotional support. Many take advantage of the anonymity offered by the online community. Initiated in June 2000, the Cancer Survivors Network currently has more than 164,000 member participants and offers a way to share people's experiences about cancer and cancer treatments and to support one another. Understanding emotional impacts of online participation on survivors and their informal caregivers can help provide useful insight into the design of new features or enhancement of the existing ones in improve the facilitation of emotional support to the network members.

Sentiment analysis aims to determine the members' subjective attitude and reflect their emotions. Analyzing the sentiment of posts in Online Health Communities is important, since it enables investigation of factors that affect the sentiment change and discovery of sentiment change patterns [2]. However, the increasing amount of online information makes a manual analysis infeasible.

In a substantial number of works [3, 4, 5], prior knowledge, i.e., a general sentiment lexicon, plays a central role. In such a lexicon, a prior sentiment score is paired with each term. The terms and scores are used to predict the overall sentiment. However, we hypothesize that there is not a general sentiment lexicon that works well in any context, since it is well known that sentiments of terms are sensitive to the domain [6, 7].

In this paper, we aim to mitigate the drawbacks in the abovementioned approaches and, hence, to improve sentiment predictive performance for textual posts in an online cancer survivor community. To achieve this, we propose an approach to constructing a dynamic sentiment lexicon, which is adapted to the cancer survivor community domain and assumes no prior knowledge about the domain. For this purpose, the text is represented as a bag-of-words with term-frequency (TF) as the attribute values. The learning algorithm then tries to identify the most informative terms and use them for classifying new texts. In addition to being domain-specific, the terms are not limited by any constraint, such as specific part-of-speech (POS) tags, or occurrence in a pre-defined repository, which contains a limited, pre-defined number of terms. Hence, our lexicon can contain the term *kinda*, whereas this term would not be included in lexicons that do not contain slang.

Since our approach represents a text instance (or a post) as TF attribute values of all unique terms in the corpus, the learning algorithm can potentially use a relatively large number of attributes, and thus, the classification performance might be affected by the curse-of-dimensionality. This problem is considerably aggravated when the training dataset is relatively small as in our case, and can lead to overfitting of the learned models. Therefore, to avoid overfitting, our features are generalized to a higher level of abstraction. Feature abstraction methods have been shown to effectively reduce the number of parameters of standard model without sacrificing classification accuracy [8, 9].

[1] http://csn.cancer.org

Specifically, in our approach, referred as "Dynamic senti lexicon", we first generate a dynamic sentiment lexicon with terms that are observed in the training set. Their sentiment scores are represented by odds for the negative and positive classes. The odds are computed by dividing the frequency of a term in the instances from one class by the frequency of the term in instances from the other class. In the second phase, we extract and compute *abstract features* based on the dynamic sentiment lexicon. The abstract features represent an aggregation of the high sentiment score features.

The rest of the paper is organized as follows: In Section II, we present background and related work. The construction of the dynamic lexicon and the computation of abstract features are elaborated in Section III. In Section IV, we present our experiments and results, and conclude the paper in Section V.

II. BACKGROUND

A substantial number of works construct a lexicon in order to estimate a polarity of words [4, 10, 11]. This value could be "negative," "positive," "neutral," [12] or any numeric scale. Wilson et al. (2005) train a classifier to address the question of how useful prior polarity alone is for identifying contextual polarity. They recommend a two-phase classification process. A first classifier identifies neutral and non-neutral phrases, and a second one applies a label with a positive or negative result, for the non-neutral. Turney [4] defines a phrase as two consecutive words—one is an adjective or adverb and uses a search engine to count co-occurrences of unambiguously positive (e.g., "excellent") and negative (e.g., "poor") terms with ambiguous terms. He uses the statistics to calculate each term's polarity score. Finally, he determines the prediction of the orientation of a review based on the average semantic orientation of the phrases in the review. However, his lexicon uses prior knowledge, by using the number of the retrieved results of the search queries.

Other research uses WordNet [3,13] in various ways. However, most of this research is limited by terms that are present in the lexical graph, and hence, missing some sentiment laden expressions. Kamps et al. [14] explore the WordNet graph and use the absolute distance of adjectives from "good" and "bad" to determine their orientation. This method assumes that all the words have a fixed sentiment score for each domain. Hu and Liu [3] use the WordNet graph by exploring mainly antonyms and synonyms of a seed set of adjectives, whose polarity is known. Reference [15] constructs the lexicon by using a seed set of 360 positive, negative, and neutral terms and an expansion mechanism using WordNet. Yet, the two last works entail a manual annotation of the seed sentiment lexicon, and are limited to some extent to words that are presented in WordNet.

A. The General Sentiment Lexicon Approach

Sentiment lexicon holds a score for each of its terms *t* representing the degree of sentiment of *t*. This score can be obtained by a variety of methods. In a substantial number of sentiment classification works, these scores are used to predict the overall sentiment of document by extracting related features [2, 3, 15].

A *general sentiment lexicon* is a static dictionary of terms and scores that can be used for any sentiment analysis domain. As such, it encodes prior knowledge that can be used for any domain. However, there is no general sentiment lexicon that is optimal in any domain, since, as was mentioned before, sentiments of some terms are sensitive to the context or domain. A good example is the term *unpredictable*, which has a negative polarity in an automotive or a health domain, but could have a positive orientation in the domain of a video-games: *What makes FF7 a masterpiece is its immersive, complex, and unpredictable plot*. In the sentiment lexicon SentiWordNet [16], *unpredictable* has a negative orientation regardless of the domain in which it appears. Another example is the sentence: *The device was small and handy*. The word *small* appears in a positive context, whereas in the sentence, *The waiter brought the food on time, but the portion was very small*, the term *small* conveys a negative sentiment.

Conventionally, sentiment scores range from -1 to +1, however, many lexicons hold binary scores: positive (+1) and negative (-1), such as [3]. This poses an additional drawback since terms' polarity may convey different sentiment strength, even being both from the same class. For example, 'like' and 'love' are both considered as positive, yet 'love' is considered to convey a stronger positive sentiment.

In contrast to the general sentiment lexicon and previous works described above, in our work we wish not only to create these term sets but also to assign each term with a ratio score that represents how likely it is that the term has the designated positive or negative sentiment with respect to the opposite sentiment class. Since these ratio scores are computed only based on the observed dataset, they are representative for the cancer survivor community domain.

III. METHODOLOGY

Our documents are sparse word sequences, which at the same time constitute opinions. We anticipate that the opinions are predictive, in terms of positive or negative sentiment.

A. Data

The posts that were used in this work are taken from [2]. More than 468,000 forum posts from 48,779 threads were downloaded from the Cancer Survivor Network, posted from July 2000 to October 2010 by 27,173 participants. Since labeling manually all the posts with a sentiment class is not feasible, a subset was sampled.

TABLE I. EXAMPLES OF POSTS AND THEIR SENTIMENT LABELS

Label	Post
Positive	Make me go to itunes to see if i can find it ... . lol starts out sad - reads to me that it ends on a somewhat positive vein.
Negative	I'm afraid there is an environmental problem you should see about

In total, 293 posts were selected randomly from the breast cancer forum of the Cancer Survivor Network and each post was manually classified as being of positive or negative sentiment, with the result that 201 of them were labeled as

positive and 92 were labeled as negative. Table I shows examples of a negative and of a positive post.

B. Constructing a Dynamic Sentiment Lexicon

We extracted from each post all of its uni-grams and bi-grams in order to also incorporate expressions into the analysis. All of these words (uni-grams) and expressions (bi-grams), namely terms, which are observed in at least 3 posts from the same class, are stored along with their normalized frequencies, according to each class they represent. For example, the term *to hear* appears in 16 positive posts, out of 180, and therefore its positive frequency is 0.089, and observed in 2 posts, out of 82 training negative posts, and therefore its negative frequency is 0.024. Since longer words' sequences (n-grams, for n>2) are not frequent, they are not likely to add any useful information to our analysis; thus, we only chose to store sequences of words of size 1 or 2.

For every entry, we calculated the positive to negative likelihood ratio and then the negative to positive likelihood ratio as follows: The ratio score was calculated by dividing the frequency of the term in each class by its frequency in all other classes. For the term *to hear* the score is 0.089/0.024=3.7 for the positive class and 0.024/0.089=0.27 for the negative. In case that a term was observed only in a single class examples, then it is considered as observed once in the second class, for calculating the ratio. Examples of positive and negative terms-ratio pairs are given in Table II.

After constructing the lexicon, the next step is to perform abstract feature generation.

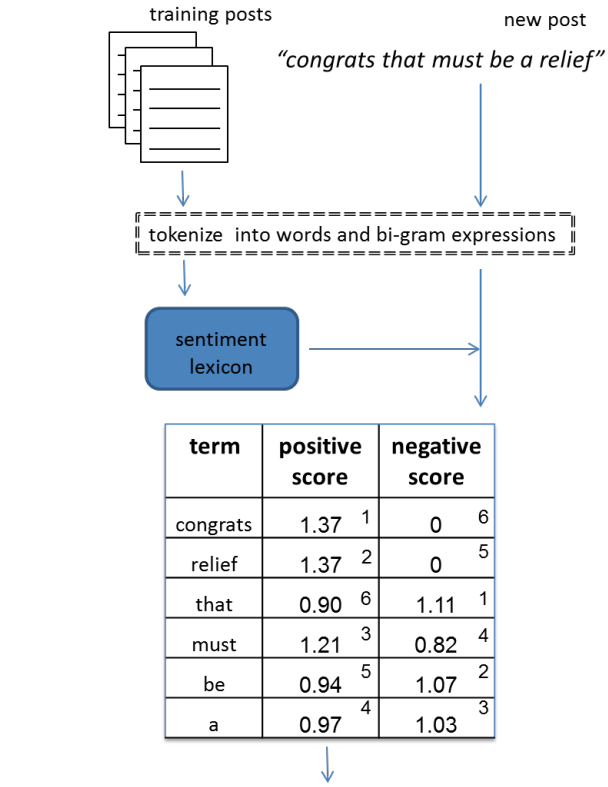
C. Abstract Feature Generation

We wish to extract predictive features based on the sentiment lexicon terms, but not use the actual terms as features because a relatively large number of dimensions in conjunction with a small number of training instances might result in a poor predictive model as a result of *overfitting*. Having said that, we compute three types of features that are an abstraction of the lexicon terms. These features are computed twice, once for each class value. The 'top 1' feature is the highest sentiment score among all the terms of the post instance. The 'top 3' feature is an accumulation of the top three score terms. Similarly, the third feature is 'top 6', which accumulates the scores of six highest sentiment score terms.

The calculation of the features is as follows. First, we tokenize each post to all of its terms; then assign each term with the two sentiment scores from the lexicon, one is the positive score and the second is the negative. Note that at this stage the sentiment scores in the lexicon represent the frequency ratio for each class with respect to the second class. The terms are sorted (in descending order) into two lists, according to each sentiment score in each class (see Fig. 1, and the table labeled by **term**; note that the numbers close to the scores represent the word ranks in the sorted lists). Next, we iterate the negative score list and the positive score list separately, and calculate the three features. The 'top 1' feature is the score of the top term, the 'top 3' feature is an accumulation of the top three scores, and the 'top 6' accumulated the top six scores. This is repeated for the positive and negative classes. The total number of features is six (the top 1, 3, 6 scores multiplied by each of the two class types) as shown in Fig. 1. In the figure, we only show the calculation of uni-gram scores since the bi-gram terms did not exceed the predefined threshold of 3 occurrences in a single class instances. The bi-gram scores can be calculated in a similar fashion. This method can be used for multi-class classification tasks, where the terms' scores will be computed according to their frequency ratio in each class, with respect to all other classes.

TABLE II. EXAMPLE OF PROMINENT POSITIVE AND NEGATIVE ENTRIES IN THE SENTIMENT LEXICON

Term	Positive frequency ratio	Term	Negative frequency ratio
hugs	10.02	seem	13.17
glad	9.11	stage	5.49
happy	9.11	diagnosed	3.66
love	7.52	radiation	3.29
news	6.38	asked	3.29
thanks for	5.47	doctor	2.74



term	positive score	negative score
congrats	1.37 1	0 6
relief	1.37 2	0 5
that	0.90 6	1.11 1
must	1.21 3	0.82 4
be	0.94 5	1.07 2
a	0.97 4	1.03 3

	positive			negative		
feature name	top_1	top_3	top_6	top_1	top_3	top_6
feature value	**1.37**	**3.95**	**6.76**	**1.11**	**3.21**	**4.03**

Fig 1. An example for calculating the six features for a post instance. The small numbers in the terms list represent their sorted (descending) order by score, per class

TABLE III. COMARISON RESULTS ON A VARIOUS OF CLASSIFIERS

Classifier	Method	% Classification Accuracy	F - measure	ROC Area
Logistic Regression	Prior-knowledge senti	78.43	0.774	0.819
	Dynamic senti lexicon	78.13	0.778	0.846
	Dynamic senti lexicon + Prior- knowledge senti	**79.45**	**0.793**	**0.860**
Random Forest	Prior- knowledge senti	75.41	0.753	0.794
	Dynamic senti lexicon	79.51	0.790	0.820
	Dynamic senti lexicon + Prior- knowledge senti	**81.48**	**0.811**	**0.852**
Rotation Forest	Prior- knowledge senti	75.05	0.735	0.791
	Dynamic senti lexicon	77.11	0.765	0.803
	Dynamic senti lexicon + Prior- knowledge senti	**79.78**	**0.793**	**0.823**
Adaboost (LMT)	Prior- knowledge senti	76.11	0.760	0.780
	Dynamic senti lexicon	**79.52**	**0.793**	**0.849**
	Dynamic senti lexicon + Prior- knowledge senti	79.10	0.785	0.845

TABLE IV. COMPARING TF VECTOR SPACE WITH OUR METHOD ON SVM CLASSIFIER

Method	% Classification Accuracy	F - measure	ROC Area
TF vector space (uni grams + bi grams)	72.72	0.717	0.702
Dynamic senti lexicon	**75.78**	**0.753**	**0.746**

IV. EXPERIMENTS AND RESULTS

We designed a binary classification task, where each post is an instance, and we only use its textual content. The target class values are positive and negative. The goal is to train a model that can be used to predict the class of any unlabeled instance.

A. Experimental Design

The first type of experiments are designed to compare our 'Dynamic senti lexicon' approach, that construct a dynamic sentiment lexicon based on which features are extracted, with the method presented in [2], referred as 'Prior-knowledge senti'. In this second approach, a prior-knowledge is used by extracting features based of a general sentiment lexicon. In additional to the general lexicon based features, more features were extracted by the 'Prior-knowledge senti' approach, such as the number of names mentioned and the number of words in a post. The full list of features, thirteen in total is detailed in [2]. In addition to a comparison between the two approaches, we tried a third method, where our dynamic lexicon based feature set is augmented by the 'Prior-knowledge senti' feature set.

In each experiment, we split the dataset into ten sets. 90% of the examples were used for training and 10% for testing, and a 10-folds cross validation evaluation was performed, each using a corresponding sentiment lexicon (that was constructed based on the 90% training instances), therefore 10 lexicons were constructed in total. We learned a classification model based on the training set to predict the actual class of the test instances. Several classifiers that we found adequate for that task were used in our evaluation, specifically Logistic Regression, Random-forest [17], Rotation-Forest [18] and Adaboost [19] with J48 [20] as base classifier, and a Logistic Regression at each leaf, (LMT) [21].

The second type of experiment aims to evaluate the effect of feature space reduction by abstraction. We compared our method with bag-of-words TF as the attribute values, where the dimensionality of the dataset is determined by the number of unique terms. Here terms are defined by uni-grams and bi-grams, denoted by 'TF vector space' approach. Again 10 folds cross validation experiments were conducted; the folds and lexicon that were used are the same as in the first experiment type. After evaluating several classifiers, we show the results based on the SVM algorithm [22], that yields the best results for the 'TF vector space' approach.

B. Results

Table III presents the classification results of the first experiments type. The percentage of accuracy, F-measure and area under ROC curve are given on the four classifiers used. Our approach outperforms the 'Prior-knowledge senti' approach by all measurements in all of the four classifiers, except for the Logistic Regression's accuracy, where the difference is minor (0.3%). The difference for the two approaches, taken as an average on all the classifiers, is from 2.3% up to 3.3% for all of the three measurements: accuracy, F-measure and ROC area. Adaboost, with the LMT base classifier, achieved the best results for our 'Dynamic senti lexicon' method, in terms of classification accuracy and F-measure.

These results are aligned with our assumption, that classifying sentiment of posts in an online cancer survivor

community yields better results when using a dynamic sentiment lexicon. This becomes even clearer considering the fact that the 'Prior-knowledge senti' approach uses thirteen features, some of which are not related to general lexicon, while our approach uses only six features, all of which are derived from the dynamic lexicon.

We experimented with an augmented feature approach as well, the 'Dynamic senti lexicon + Prior-knowledge senti', where the feature space composed of the features of the two approaches. This approach outperforms our approach by all measurements, and algorithms, except for the Adaboost. The best performance in each measurement is given by the augmented feature approach. This implies that the information of the two first approaches is not redundant, and can be used to improve classification results.

In the second type of experiments, our approach outperforms the 'TF vector space' approach by all measurements, as Table IV demonstrates. The difference is more than 3% for all: the accuracy, F-measure and ROC area. This finding shows that by extracting abstracted features, and reducing the feature space, we might avoid overfitting the training set.

V. Conclusions and Future Work

In this work, we examined the impact of using a dynamic sentiment lexicon and the generation of abstract features to avoid the curse-of-dimensionality, in improving sentiment predictive performance for textual posts in an online cancer survivor community.

The results of our experiments show that classifiers trained using abstract features extracted from a dynamic sentiment lexicon outperform those trained using features extracted from a general sentiment lexicon. In future, it would be interesting to extend the approach proposed here to semi-supervised settings that can exploit large amounts of unlabeled data together with limited amounts of labeled data in training classifiers.

Acknowledgements

We would like to thank the American Cancer Society for providing us the Cancer Survivor Network data that was used in this research. This work was done while Nir Ofek and Lior Rokach were visiting The Pennsylvania State University.

References

[1] K. Zickuhr, “Generations 2010,” Pew Internet, Tech. Rep., 2010

[2] B. Qiu, K. Zhao, P. Mitra, D. Wu, C. Caragea, J. Yen, G. E. Greer, K. Portier. (2011) Get Online Support, Feel Better--Sentiment Analysis and Dynamics in an Online Cancer Survivor Community. In: SocialCom 2011.

[3] H. Minqing and L. Bing. Mining and Summarizing Customer Reviews. SIGKDD 2004, pages 168-177, 2004

[4] P. D. Turney. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In Procs. of ACL’02 , pages 417–424. 2002

[5] T. Wilson, J. Wiebe, and P. Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In Proceedings of the Human Langugae Technology Conference and Conference on EMNLP, pages 347–354. ACL, 2005.

[6] P. D. Turney and M. L. Littman. Measuring praise and criticism: Inference of semantic orientation from association. ACM Trans. Inf. Syst., 21(4):315-346, 2003

[7] J. Blitzer, M. Dredze, and F. Pereira. 2007. Biographies, Bollywood, Boom-boxes, and Blenders: Domain Adaptation for Sentiment Classification. ACL 2007

[8] L. D. Baker, and A. K. McCallum. 1998. Distributional clustering of words for text classification. In ACM SIGIR , 96–103. ACM Press.

[9] C. Caragea, A. Silvescu, S. Kataria, D. Caragea, and P. Mitra. "Classifying Scientific Publications Using Abstract Features." In: SARA, Parador de Cardona, Spain, 2011

[10] T. Mullen and N. Collier, “Sentiment analysis using support vector machines with diverse information sources,” in Proceedings of the Conference on EMNLP, pp. 412–418, July 2004. (Poster paper).

[11] K. Yang, N. Yu, and H. Zhang. WIDIT in TREC 2007. Blog Track: Combining Lexicon-Based Methods to Detect Opinionated Blogs. In Proceedings of TREC 2007

[12] S. R. Das and M. Y. Chen, “Yahoo! for Amazon: Sentiment extraction from small talk on the Web,” Management Science, vol. 53, pp. 1375–1388, 2007

[13] G. A. Miller. Wordnet: A lexical database for English. Communications of the ACM, (11):39–41, 1995. http://wordnet.princeton.edu/

[14] J. Kamps, M. Marx, R. J. Mokken, and M. De Rijke. 2004. Using WordNet to measure semantic orientation of adjectives. In: LREC, 2004, Lisbon, PT

[15] S. Blair-Goldensohn, K. Hannan, R. McDonald, T. Neylon, G. A. Reis, and J. Reynar. Building a sentiment summarizer for local service reviews. In WWW Workshop on NLP in the Information, 2008.

[16] A. Esuli and F. Sebastiani. 2006. Sentiwordnet: A publicly available lexical resource for opinion mining. In Proc. of LREC 2006.

[17] L. Breiman (2001). Random Forests. Machine Learning. 45(1):5-32.

[18] J. J. Rodriguez, L. I. Kuncheva, C. J. Alonso (2006). Rotation Forest: A new classifier ensemble method. IEEE Transactions on Pattern Analysis and Machine Intelligence. 28(10):1619-1630

[19] Y. Freund, R. E. Schapire: Experiments with a new boosting algorithm. In: Thirteenth International Conference on Machine Learning, San Francisco, 1996.

[20] H. Witten and E. Frank.: Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann, 1999.

[21] N. Landwehr, M. Hall, E. Frank (2005). Logistic Model Trees. Machine Learning. 95(1-2):161-205.

[22] C. Cortes and V. Vapnik, “Support-vector network,” Machine Learning, vol. 20, pp. 273–297, 1995