Sparse unsupervised feature learning for sentiment classification of short documents

Simone Albertini **Ignazio Gallo** **Alessandro Zamberletti**

University of Insubria
Varese, Italy
simone.albertini@uninsubria.it

Unpublished Manuscript

Abstract

The rapid growth of Web information led to an increasing amount of user-generated content, such as customer reviews of products, forum posts and blogs. In this paper we face the task of assigning a sentiment polarity to user-generated short documents to determine whether each of them communicates a positive or negative judgment about a subject. The method we propose exploits a Growing Hierarchical Self-Organizing Map to obtain a sparse encoding of user-generated content. The encoded documents are subsequently given as input to a Support Vector Machine classifier that assigns them a polarity label. Unlike other works on opinion mining, our model does not use a priori hypotheses involving special words, phrases or language constructs typical of certain domains. Using a dataset composed by customer reviews of products, the experimental results we obtain are close to those achieved by other recent works.

1 Introduction

E-commerce has grown significantly over the past decade. As such, there has been a proliferation of reviews written by customers for different products. Those reviews are of great value for the businesses as they convey a lot of information both about sellers and products; the most important information that may be inferred from these reviews is the overall satisfaction of customers.

With *sentiment analysis* or *opinion mining* we refer to the task of assigning a sentiment polarity to text documents to determine whether the reviewer expressed a positive, neutral or negative judgment about a subject (Bo and Pang, 2008). *Sentiment analysis* is a difficult task, therefore several issues arise when trying to solve it. For example, in some works the problem of unbalanced information sources is dealt with (Li et al., 2011; Li et al., 2012). Other approaches manage to build a lexicon of opinion-bearing words or phrases to expose syntactic dependencies (Kanayama and Nasukawa, 2006; Wen and Wu, 2011; Ku et al., 2011). Different natural language processing techniques are adopted to support the building of dictionaries and lexicons to identify opinion-bearing words such as the polarity of specific part-of-speech influenced by the context (Hatzivassiloglou and McKeown, 1997; Turney and Littman, 2002; Nakagawa et al., 2010). In has been proved that machine learning models can be successfully exploited to face the problem of *sentiment analysis* (Pang and Vaithyanathan, 2002; Wilson et al., 2004). Recent works use unsupervised (Maas et al., 2011; Turney and Littman, 2002) or semi-supervised (Socher et al., 2011) learning algorithms to generate a proper vector-space representation of the documents.

The extraction of opinions expressed by customers about specific features is an interesting and useful task that has been successfully applied to several different sources of information, such as movies (Zhuang et al., 2006) or product reviews (Hu and Liu, 2004; Popescu and Etzioni, 2005; Ding et al, 2008). Such approaches usually lack of generality as they require prior information strictly related to the specific topic or domain.

In this paper, we face the problem of classifying short documents associated to product reviews in order to assign them a positive or negative polarity. We explore the possibility to solve such task without using any prior information such as assumptions on the language, linguistic patterns or idioms; moreover, no opinion-bearing words dictionaries are employed. In our model, we adopt several well-known techniques to encode text documents. The encoded documents are clustered in an

unsupervised manner using a Growing Hierarchical Self-Organizing Map (GHSOM) (Rauber et al., 2002) to obtain a new sparse encoding that is provided as input to a Support Vector Machine (SVM) classifier (Cortes and Vapnik, 1995) that assigns them the correct polarity labels. The experimental results we present prove that the proposed method can overcome the baseline results obtained using bag-of-words encodings without employing any sparse features learning; furthermore, we show that our domain-independent approach is able to obtain results comparable with those achieved by approaches that exploit prior information defined for the specific domain.

2 Related Works

Several works in literature face the *sentiment analysis* task using machine learning algorithms. In the following paragraphs we introduce some the models that we consider strictly related to the method presented in our paper.

Pang and Vaithyanathan (2002) adopt corpus based methods using machine learning techniques rather than relying on prior intuitions; their main goal is to identify opinion-bearing words. The documents are encoded using a standard bag-of-words framework and the sentiment classification task is treated as a binary topic-based categorization task. They prove that: (i) the SVM classification algorithm outperforms the others, (ii) good results can be achieved using unigrams as features with presence/absence binary values rather than the frequency, unlike what usually happens in topic-based categorization.

Maas et al. (2011) propose an unsupervised probabilistic model based on the Latent Dirichlet Allocation (Blei et al., 2003) to generate a vector representation of the documents. A supervised classifier is employed to cause semantically similar words to have similar representation in the same vector space. They argue that incorporating sentiment information in Vector Space Model approaches lead to good overall results.

Socher et al. (2011) employ a semi-supervised recursive auto-encoder to obtain a new vector representation of the documents. Such representation is used during the classification task. Note that this approach does not employ any language specific sentiment lexicon or bag-of-words representations.

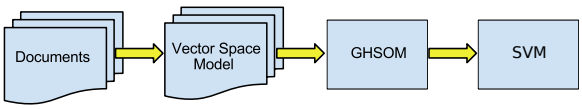

Figure 1: An overview of the proposed model. From left to right: the short documents are represented in a VSM, they are given as input to a GHSOM, the output of the GHSOM is exploited by an SVM classifier.

3 Proposed Model

A detailed description of the proposed method is given in the following paragraphs. It consists in a set of phases in which each step addresses a specific task. The whole solution involves a supervised training procedure that exploits: (i) an unsupervised neural network for feature learning and (ii) a supervised classifier for document classification.

In Figure 1 we present an overview of the proposed method, it can be observed that the set of raw documents received as input by our model are represented in a Vector Space Model (VSM). The weight assigned to each term of the dictionary is computed using a weighting function w. In details, given a set of documents D, a dictionary of terms T is extracted. The weighting function $w_T : D \rightarrow X,\ X \subset [0,1]^{|T|}$ produces a vector representation $\vec{x} \in X$ of the document d in the space defined by the terms in the dictionary T. In Section 3.1 we discuss in details all the weighting functions tested in our experiments.

The vector space representation of the input documents is given as training data to a GHSOM that learns a new representation for the input data as described in details in Section 3.2. The GHSOM generates a set of maps that hierarchically represent the distribution of the training data. Note that, after the initial training phase, the topology of each map is fixed. At the end of the training phase, we assign a progressive numerical identifier to each k leaf units in the maps generated by the trained GHSOM and we define the learned k-dimensional feature space as F. Each training pattern $\vec{x} \in X$ of the GHSOM is mapped into a sparse feature vector by a function $feat : X \rightarrow F,\ F \subset [0,1]^k$. For each feature vector $\vec{f} \in F$ the following holds:

$$\vec{f}(i) = \begin{cases} 1 & \text{if } \vec{x} \text{ activates } u_i \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Algorithm 1 Overview of the Proposed Model.

Training

1. Build the dictionary of terms T from the set of all documents D.
2. Map all the training documents $d \in D$ in the VSM representation $w_T(d) = \vec{x},\ \vec{x} \in X$ using the dictionary T.
3. Train a GHSOM with the patterns in X. Once the training phase ends, the number of maps generated by the GHSOM is k.
4. Each pattern $\vec{x} \in X$ is mapped in the k-dimensional feature space F using the function $feat(\vec{x}) = \vec{f}$. Let Y be the set of all feature patterns computed in this way.
5. Train a SVM classifier using the patterns in Y along with their respective labels.

Prediction of a document $\bar{d}$

1. Get the VSM representation $\vec{x} = w_T(\bar{d})$.
2. Compute the corresponding feature vector $\vec{f} = feat(\vec{x})$ using the trained GHSOM.
3. Predict the polarity of $\bar{d}$ by classifying the pattern f using the trained SVM.

where u_i is the i-th leaf unit of the GHSOM and $0 < i \leq k$. All the training patterns are mapped to obtain a set of corresponding feature vectors in F. This new set of patterns, along with their respective labels, constitutes the training data of a SVM classifier. Once the training phase ends, the classifier is able to assign a positive or negative label to each of its input patterns. In our experiments we evaluate the performances achieved by our model trying an SVM with both a linear and a radial basis function kernels. The linear kernel is used to evaluate the ability of the proposed model to generate a non-linear feature representation of the input patterns in a new space where the points of different classes are linearly separable. The radial basis function kernel is adopted to obtain a non-linear separating plane. In Algorithm 1 we summarize the steps involved in our approach.

3.1 Short Texts Representation

In this section we describe how the short documents are represented in a VSM using a bag-of-words approach. Let D be the set of all documents and V be a vector space with a number of dimensions equals to the number of terms extracted from the corpus. Using an encoding function, we assign to each document $d \in D$ a vector $v_d \in V$, where $v_d(i) \in [0, 1]$ is the weight assigned to the i-th term of the dictionary for the document d. In our experiments we compare the results achieved by our model using five different encoding functions that are presented in the following paragraphs.

Binary Term Frequency. It produces a simple and sparse representation of a short document. Such representation lacks of representative power but acts as an information bottleneck when provided as input to a classifier. Given a term t and a document d, Equation 2 is used to compute the value of each weight.

$$binary\_score(d,t) = \begin{cases} 1 & \text{if } t \in d \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

TF-IDF. It is a well-known method usually employed to compute the weights in a VSM. Using Equation 3, the weight assigned to a document d is proportional to the frequency of the term t in d (called tf) and it is inversely proportional to the frequency of t in the corpus D (called df).

$$TF \cdot IDF(d,t) = tf(d,t) \cdot \log(\frac{|D|}{df(D,t)}) \quad (3)$$

In our experiments we compare the results obtained using the TF-IDF term weighting approach applied both to unigrams and unigrams plus bigrams.

Specific against Generic and One against All. In Equation 4 we present a generic way to assign a weight to each term t of a document d.

$$score(t, sc, gc) = 1 - \frac{1}{log_2(2 + \frac{F_{t,sc} \cdot D_{t,sc}}{F_{t,gc}})} \quad (4)$$

sc and gc are two sets of documents representing the specific corpus and the generic corpus respectively. $F_{t,sc}$ and $F_{t,gc}$ are the frequencies of the term t in sc and gc respectively. The number of documents in sc containing the term t is defined as $D_{t,sc}$.

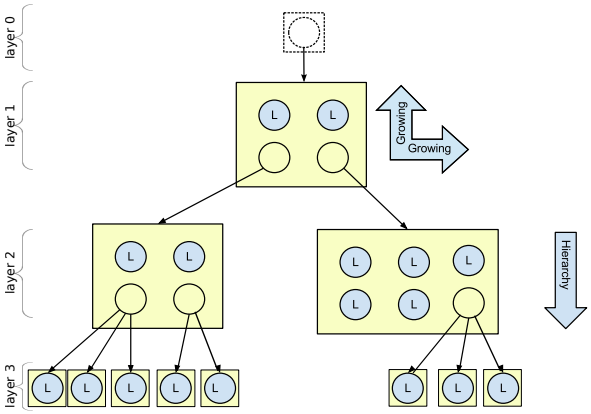

Figure 2: An example showing a GHSOM model. Each layer in the hierarchical structure is composed by several independent SOMs; the units with high mqe are expanded to form a new SOM in their subsequent layers; the units L that represent an homogeneous set of data do not require any expansion.

The weight assigned to each term t in d by Equation 4 is proportional to $F_{t,sc}$ and inversely proportional to $F_{t,gc}$. Therefore, the value of the $score$ function is close to 0 when t does not appear in gc (e.g., when t is a domain-specific term) and it increases according to the value of $\frac{F_{t,sc}}{F_{t,gc}}$. When $t \notin gc$, $score(t, sc, gc) = 1$.

Using Equation 4, two weighting strategies may be defined: (i) the *Specific against Generic* (SaG), where sc is the set of positive-oriented documents and gc is the set of negative-oriented documents, (ii) the *One against All* (OaA), where sc is the set of all the short documents in the corpus and gc is a set of short documents semantically unrelated to the ones in sc.

3.2 GHSOM

In this section we describe the Growing Hierarchical Self-Organizing Map (GHSOM) (Rauber et al., 2002) model.

The GHSOM model is an evolution of the Self Organizing Map (SOM) (Kohonen, 2001) model. The latest is an unsupervised neural network composed by a two dimensional grid of neurons. The aim of a SOM is to learn a quantized representation of the training patterns in their space by adjusting the weights associated to each neuron in order to fit the distribution of the input data. By doing so, a SOM operates a sort of clusterization of the input data, where the weight vector assigned to each neuron is a centroid.

In Figure 2 we show an example of GHSOM, it consists of a set of SOMs organized in a hierarchical structure that is built by an iterative procedure. This procedure starts from a single map and, when convenient, increases the size of the current map by adding rows and columns of neurons or by expanding a single neuron in another SOM. The criterion employed to modify the topology of a GHSOM is based on the quantization error and two parameters τ_1 and τ_2; these parameters adjust the propensity of the structure to grow in width (new rows/columns are added to the SOMs) and in depth (new SOMs are added) respectively. The mean quantization error *mqe* is a measure of the quality of each SOM; the greater the *mqe*, the higher the approximation level. The quantization error can be computed for a single unit and for a whole map using Equations 5 and 6 respectively.

$$mqe_i = \frac{1}{|C_i|} \sum_{x_j \in C_i} \|m_i - x_j\| \quad (5)$$

$$mqe_M = \frac{1}{|M|} \sum_{i \in M} mqe_i \quad (6)$$

Let u_i be the neuron of a SOM M, m_i be the weight vector of u_i and C_i be the set of the input vectors associated to u_i.

The training process begins with the creation of an initial map constituted by only one unit whose weight vector is computed as the mean of all the training vectors. This map constitutes the layer 0 of the GHSOM; we define mqe_0 as its mean quantization error. In the subsequent layer, a new SOM M_1 is created and trained using the standard SOM training algorithm (Kohonen, 2001). After a fixed set of iterations, the mean square error mqe_{M_1} is computed and the unit u_e having the maximum square error is identified by computing $e = \text{argmax}_i \{mqe_i\}$. Depending both on the dissimilarity of its neightboring units and τ_1, a new row or column is inserted at the coordinates of the unit u_e. Note that M_1 is allowed to grow while the following condition holds:

$$mqe_{M_1} \geq (\tau_1 \cdot mqe_{M_0}) \quad (7)$$

When Equation 7 is no longer satisfied, the units of M_1 having high mqe may add a new SOM in the next layer of the GHSOM. The parameter τ_2 is used to control whether a unit should be expanded in a new SOM. A unit $u_i \in M_1$ is subject to hierarchical expansion if $mqe_i \geq \tau_2 \cdot mqe_0$.

The described procedure is recursively repeated by iteratively expanding the SOMs both in depth and width. Note that each map in a layer is trained using only the training patterns clustered by its parent unit. The training process of a map ends when no further expansions are allowed.

4 Experiments

In this section we present the results obtained by performing an extensive experimental analysis of the proposed model. The main goal of such experiments is to determine: (i) how the parameters of our model affect its performances, (ii) the magnitude of the contribution of the GHSOM and SVM in the proposed model, (iii) how our method performs in comparison with other approaches.

All our experiments are carried out using the Customer Review Dataset (Hu and Liu, 2004). The dataset is composed by several annotated reviews associated with 5 different products; each review consists of a set of short phrases whose length do not averagely exceed 30 words. All the phrases are independently annotated, thus they can be treated as short documents; moreover, their polarities can be predicted independently from the reviews they belong to. The Customer Review Dataset is composed by a total of 1095 positive and 663 negative phrases; in our experiments we balance it by removing 432 positive phrases.

We evaluate the performances achieved by the proposed model using the *F-measure* defined as in Equation 8.

$$F_1 = 2 \cdot p \cdot r/(p+r) \tag{8}$$

p and r represent precision and recall values respectively. In all our experiments, the parameters τ_1 and τ_2 are chosen using k-fold cross-validation with $k = 5$.

Baseline. In the first part of our experiments, we measure the results achieved by our model using the 4 encodings described in Section 3.1; the vector representations generated by those encodings are classified using an SVM with both a linear and a radial basis function kernel. As shown in Table 1, the results obtained using the linear and non-linear kernels are similar. In fact, the vector space has a great dimensionality, therefore mapping the data into an higher dimensional non-linear space do not improve the classification performances. Note that this first part of the experiments is crucial for the subsequent phases because the use

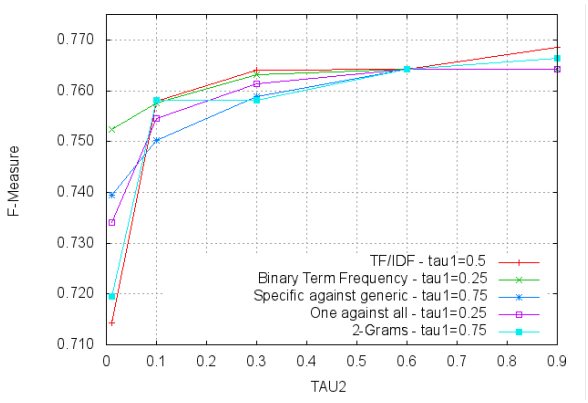

Figure 3: F-measure values achieved by a trained GHSOM for the Customer Review Dataset, while varying the parameter τ_2. For each of the five encodings of Sections 3.1 and 3.2, the optimal parameter τ_1 was found using a k-fold cross-validation technique with $k = 5$.

of an unsupervised feature learning approach is meant to obtain a new non-linear encoding and it is important to know if this unsupervised encoding is able to outperform the results obtained using a non-linear SVM kernel on the same input vectors.

GHSOM analysis. In this second part of our experiments, we analyse the distribution of the documents in the clusters produced by a trained GHSOM. Given a trained GHSOM, we assign a polarity to each of its leaf units. Let u_i be a leaf unit in the map M generated by an expansion of the unit u_{par} belonging to the previous layer. We define $P = P_{pos} \cup P_{neg}$ as the set of training patterns clustered by the unit u_i. The polarity assigned to u_i is computed using Equation 9.

$$pol(u_i) = \begin{cases} pos & \text{if } |P_{pos}| > |P_{neg}| \\ neg & \text{if } |P_{neg}| > |P_{pos}| \\ pol(u_{par}) & \text{otherwise} \end{cases} \tag{9}$$

It is possible to exploit the GHSOM as a clustering algorithm: each leaf unit is a centroid in the input patterns space and each unseen document is assigned the polarity of its closest centroid. Given an unseen document $\bar{d}$, we compute its closest leaf unit $u_{\bar{d}}$ as described in Section 3.2 and its polarity as $pol(u_{\bar{d}})$.

The results obtained using this simple clusterization algorithm are presented in Table 1; we observe a general improvement in respect to the classification results obtained using the baseline approaches. In Figure 3 we present the development

Encoding	SVM linear	SVM rbf	GHSOM	full (linear)	full (rbf)
Binary term frequency	0.52	0.56	0.75	0.81	0.87
TF-IDF unigrams	0.55	0.57	0.76	0.76	0.86
TF-IDF 2-grams	0.60	0.62	0.76	0.78	0.85
Specific against generic	0.54	0.76	0.76	0.76	0.88
One against all	0.56	0.56	0.77	0.81	0.90

Table 1: F-measure values obtained by different stages of the proposed model for the Customer Review Dataset. The columns labelled with *SVM linear* and *SVM rbf* show the baseline results; the column labelled with *GHSOM* shows the results obtained by directly using a GHSOM as classifier; the last two columns show the results achieved by the final model using a linear and a radial basis function kernels.

Method	F-measure
FBS	0.83
OPINE	0.87
Opinion Observer	0.91
Our method	0.90

Table 2: Classification results in comparison with other recent works for the Customer Review Dataset.

of the *F-measure* while varying the parameter τ_2; the value assigned to τ_1 is determined using k-fold cross-validation as previously defined. It is possible to observe that, as the GHSOM grows in depth, the classification results obtained using the 5 different encodings improve. We argue that this is due to the fact that, as the number of leaf units increases, the centroids in the vector space become more specialized and precise.

Sparse encoding classification. In our final experiments we measure the results we obtain when the sparse encoding generated by the trained GHSOM, described in Section 3.2, is given as input to both a linear and a non-linear SVM classifiers. Such results are presented in Table 1. They prove that: (i) the combination of a GHSOM and a non-linear SVM classifier performs better than the baseline approaches, (ii) the encoding generated by the GHSOM defines a vector space that is better (in terms of separability) than the ones defined by the encodings presented in Section 3.1. Note that the vectors generated by the function $feat$, described in Section 3, are not linearly separable; in fact, a non-linear classifier, trained using the encoding generated by the GHSOM, performs better than a linear one.

Results. In Table 2 we provide an experimental comparison between our approach and some of the models presented in literature: the Feature-Based Summarization (FBS) (Hu and Liu, 2004), the OPINE (Popescu and Etzioni, 2005) and the Opinion Observer (Ding et al, 2008). It is important to point out that in our method we classify the short documents as either positive or negative, while the other 3 methods infer a sentiment orientation about features of the products. Moreover, we slightly modified the Customer Review Dataset by performing the following steps: (i) we discarded the neutral tagged phrases, (ii) we balanced the dataset by removing 432 positive phrases. However, the results reported in Tables 1 and 2 are obtained using k-fold cross-validation; therefore, we argue that the comparison we provide is meaningful. Our results prove that the proposed method can pose a challenge to the others without exploiting any prior information related to the specific domain.

5 Conclusion

The method presented in this paper is able to generate a sparse encoding of short documents in an unsupervised manner, without using any prior information related to the context of the problem. In our experiments we proved that a properly trained Growing Hierarchical Self-Organizing Map, used as clustering algorithm and applied to several bag-of-words approaches, provides robust results. Moreover, excellent performances can be achieved when the output of such model is provided as input to a Support Vector Machine classifier; this proves the suitability of feature learning algorithms in the field of *sentiment analysis*. Our solution presents some interesting advantages: (i) it does not depend on the language, (ii) it does not require any lexicon of opinion-bearing words nor idioms, (iii) it is domain-independet, meaning that it may be applied to different contexts without further modifications.

References

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. *Latent Dirichlet Allocation*. Journal of Machine Learning Research, 3, pages 993-1022.

Corinna Cortes and Vladimir Vapnik. 1995. *Support-Vector Networks*. Machine Learning , 20(3), pages 273-297.

Xiaowen Ding, Bing Liu and Philip S. Yu. 2008. *A Holistic Lexicon-Based Appraoch to Opinion Mining*. Proceedings of First ACM International Conference on Web Search and Data Mining (WSDM).

Vasileios Hatzivassiloglou and Kathleen R. McKeown. 1997. *Predicting the semantic orientation of adjectives*. Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL).

Minqing Hu and Bing Liu. 2004. *Mining and summarizing customer reviews*. Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD).

Hiroshi Kanayama and Tetsuya Nasukawa. 2006. *Fully automatic lexicon expansion for domain-oriented sentiment analysis*. Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP).

Teuvo Kohonen. 2001. *Self-Organizing Maps*.

Lun-Wei Ku, Ting-Hao Huang and Hsin-Hsi Chen. 2011. *Predicting Opinion Dependency Relations for Opinion Analysis*. Proceedings of 5th International Joint Conference on Natural Language Processing (IJCNLP).

Shoushan Li and Shengfeng Ju, Guodong Zhou, and Xiaojun Li. 2012. *Active learning for imbalanced sentiment classification*. Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP).

Shoushan Li, Zhongqing Wang, Guodong Zhou and Sophia Yat Mei Lee. 2011. *Semi-supervised learning for imbalanced sentiment classification*. Proceedings of the 22th International Joint Conference on Artificial Intelligence (IJCAI).

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng and Christopher Potts. 2011. *Learning word vectors for sentiment analysis*. Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL).

Tetsuji Nakagawa, Kentaro Inui and Sadao Kurohashi. 2010. *Dependency tree-based sentiment classification using CRFs with hidden variables*. Human Language Technologies (HLT).

Bo Pang and Shivakumar Vaithyanathan. 2002. *Thumbs Up? Sentiment Classification Using Machine Learning Techniques*. Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP).

Bo Pang and Lillian Lee. 2008. *Opinion mining and sentiment analysis*. Foundations and Trends in Information Retrieval, 2(1-2), pages 1-135.

Ana-Maria Popescu and Oren Etzioni. 2005. *Extracting product features and opinions from reviews*. Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT/EMNLP).

Andreas Rauber, Dieter Merkl and Michael Dittenbach. 2002. *The Growing Hierarchical Self-Organizing Map: Exploratory Analysis of High-Dimensional Data*. IEEE Transactions on Neural Networks, 13, pages 1331-1341.

Richard Socher, Jeffrey Pennington, Eric H. Huang, Andrew Y. Ng and Christopher D. Manning. 2011. *Semi-supervised recursive autoencoders for predicting sentiment distributions*. Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP).

Peter D. Turney and Michael L. Littman. 2002. *Unsupervised Learning of Semantic Orientation from a Hundred-Billion-Word Corpus*. Technical Report EGB-1094, National Research Council Canada.

Miaomiao Wen and Yunfang Wu. 2011. *Mining the Sentiment Expectation of Nouns Using Bootstrapping Method*. Proceedings of 5th International Joint Conference on Natural Language Processing (IJCNLP).

Theresa Wilson, Janyce Wiebe and Rebecca Hwa. 2004. *Just how mad are you? finding strong and weak opinion clauses*. Proceedings of the 19th national conference on Artifical intelligence (AAAI).

Li Zhuang, Feng Jing and Xiao-Yan Zhu. 2006. *Movie review mining and summarization*. Proceedings of the 15th ACM international Conference on Information and Knowledge Management (CIKM).