Sentiment Learning on Product Reviews via Sentiment Ontology Tree

Wei Wei
Department of Computer and Information Science
Norwegian University of Science and Technology
wwei@idi.ntnu.no

Jon Atle Gulla
Department of Computer and Information Science
Norwegian University of Science and Technology
jag@idi.ntnu.no

Abstract

Existing works on sentiment analysis on product reviews suffer from the following limitations: (1) The knowledge of hierarchical relationships of products attributes is not fully utilized. (2) Reviews or sentences mentioning several attributes associated with complicated sentiments are not dealt with very well. In this paper, we propose a novel HL-SOT approach to labeling a product's attributes and their associated sentiments in product reviews by a Hierarchical Learning (HL) process with a defined Sentiment Ontology Tree (SOT). The empirical analysis against a human-labeled data set demonstrates promising and reasonable performance of the proposed HL-SOT approach. While this paper is mainly on sentiment analysis on reviews of one product, our proposed HL-SOT approach is easily generalized to labeling a mix of reviews of more than one products.

1 Introduction

As the internet reaches almost every corner of this world, more and more people write reviews and share opinions on the World Wide Web. The user-generated opinion-rich reviews will not only help other users make better judgements but they are also useful resources for manufacturers of products to keep track and manage customer opinions. However, as the number of product reviews grows, it becomes difficult for a user to manually learn the panorama of an interesting topic from existing online information. Faced with this problem, research works, e.g., (Hu and Liu, 2004; Liu et al., 2005; Lu et al., 2009), of sentiment analysis on product reviews were proposed and have become a popular research topic at the crossroads of information retrieval and computational linguistics.

Carrying out sentiment analysis on product reviews is not a trivial task. Although there have already been a lot of publications investigating on similar issues, among which the representatives are (Turney, 2002; Dave et al., 2003; Hu and Liu, 2004; Liu et al., 2005; Popescu and Etzioni, 2005; Zhuang et al., 2006; Lu and Zhai, 2008; Titov and McDonald, 2008; Zhou and Chaovalit, 2008; Lu et al., 2009), there is still room for improvement on tackling this problem. When we look into the details of each example of product reviews, we find that there are some intrinsic properties that existing previous works have not addressed in much detail.

First of all, product reviews constitute domain-specific knowledge. The product's attributes mentioned in reviews might have some relationships between each other. For example, for a digital camera, comments on image quality are usually mentioned. However, a sentence like "40D handles noise very well up to ISO 800", also refers to image quality of the camera 40D. Here we say "noise" is a sub-attribute factor of "image quality". We argue that the hierarchical relationship between a product's attributes can be useful knowledge if it can be formulated and utilized in product reviews analysis. Secondly, Vocabularies used in product reviews tend to be highly overlapping. Especially, for same attribute, usually same words or synonyms are involved to refer to them and to describe sentiment on them. We believe that labeling existing product reviews with attributes and corresponding sentiment forms an effective training resource to perform sentiment analysis. Thirdly, sentiments expressed in a review or even in a sentence might be opposite on different attributes and not every attributes mentioned are with sentiments. For example, it is common to find a fragment of a review as follows:

Example 1: *"...I am very impressed with this camera except for its a bit heavy weight especially with*

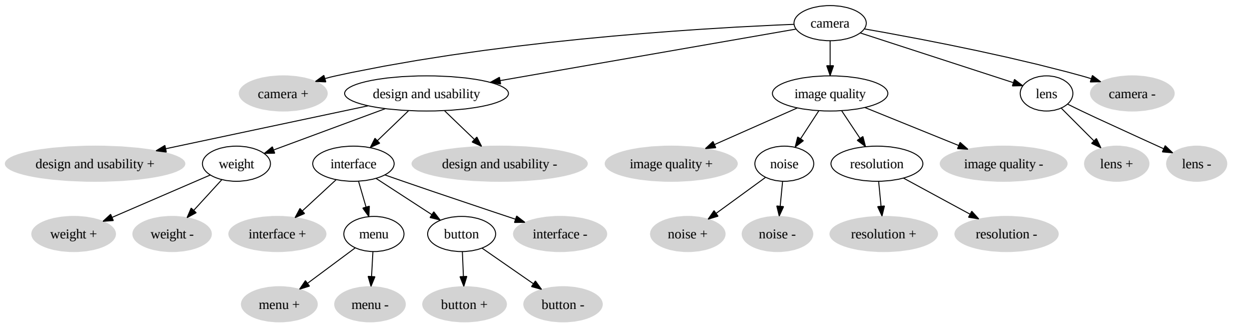


Figure 1: an example of part of a SOT for digital camera

extra lenses attached. It has many buttons and two main dials. The first dial is thumb dial, located near shutter button. The second one is the big round dial located at the back of the camera..."
In this example, the first sentence gives positive comment on the camera as well as a complaint on its heavy weight. Even if the words "lenses" appears in the review, it is not fair to say the customer expresses any sentiment on lens. The second sentence and the rest introduce the camera's buttons and dials. It's also not feasible to try to get any sentiment from these contents. We argue that when performing sentiment analysis on reviews, such as in the Example 1, more attention is needed to distinguish between attributes that are mentioned with and without sentiment.

In this paper, we study the problem of sentiment analysis on product reviews through a novel method, called the HL-SOT approach, namely Hierarchical Learning (HL) with Sentiment Ontology Tree (SOT). By sentiment analysis on product reviews we aim to fulfill two tasks, i.e., labeling a target text[1] with: 1) the product's attributes (attributes identification task), and 2) their corresponding sentiments mentioned therein (sentiment annotation task). The result of this kind of labeling process is quite useful because it makes it possible for a user to search reviews on particular attributes of a product. For example, when considering to buy a digital camera, a prospective user who cares more about image quality probably wants to find comments on the camera's image quality in other users' reviews. SOT is a tree-like ontology structure that formulates the relationships between a product's attributes. For example, Fig. 1 is a SOT for a digital camera[2]. The root node of the SOT is a camera itself. Each of the non-leaf nodes (white nodes) of the SOT represents an attribute of a camera[3]. All leaf nodes (gray nodes) of the SOT represent sentiment (positive/negative) nodes respectively associated with their parent nodes. A formal definition on SOT is presented in Section 3.1. With the proposed concept of SOT, we manage to formulate the two tasks of the sentiment analysis to be a hierarchical classification problem. We further propose a specific hierarchical learning algorithm, called HL-SOT algorithm, which is developed based on generalizing an online-learning algorithm H-RLS (Cesa-Bianchi et al., 2006). The HL-SOT algorithm has the same property as the H-RLS algorithm that allows multiple-path labeling (input target text can be labeled with nodes belonging to more than one path in the SOT) and partial-path labeling (the input target text can be labeled with nodes belonging to a path that does not end on a leaf). This property makes the approach well suited for the situation where complicated sentiments on different attributes are expressed in one target text. Unlike the H-RLS algorithm , the HL-SOT algorithm enables each classifier to separately learn its own specific threshold. The proposed HL-SOT approach is empirically analyzed against a human-labeled data set. The experimental results demonstrate promising and reasonable performance of our approach.

This paper makes the following contributions:

- To the best of our knowledge, with the proposed concept of SOT, the proposed HL-SOT approach is the first work to formulate the tasks of sentiment analysis to be a hierarchical classification problem.

- A specific hierarchical learning algorithm is

[1]Each product review to be analyzed is called target text in the following of this paper.

[2]Due to the space limitation, not all attributes of a digital camera are enumerated in this SOT; m+/m- means positive/negative sentiment associated with an attribute m.

[3]A product itself can be treated as an overall attribute of the product.

further proposed to achieve tasks of sentiment analysis in one hierarchical classification process.

- The proposed HL-SOT approach can be generalized to make it possible to perform sentiment analysis on target texts that are a mix of reviews of different products, whereas existing works mainly focus on analyzing reviews of only one type of product.

The remainder of the paper is organized as follows. In Section 2, we provide an overview of related work on sentiment analysis. Section 3 presents our work on sentiment analysis with HL-SOT approach. The empirical analysis and the results are presented in Section 4, followed by the conclusions, discussions, and future work in Section 5.

2 Related Work

The task of sentiment analysis on product reviews was originally performed to extract overall sentiment from the target texts. However, in (Turney, 2002), as the difficulty shown in the experiments, the whole sentiment of a document is not necessarily the sum of its parts. Then there came up with research works shifting focus from overall document sentiment to sentiment analysis based on product attributes (Hu and Liu, 2004; Popescu and Etzioni, 2005; Ding and Liu, 2007; Liu et al., 2005).

Document overall sentiment analysis is to summarize the overall sentiment in the document. Research works related to document overall sentiment analysis mainly rely on two finer levels sentiment annotation: *word-level sentiment annotation* and *phrase-level sentiment annotation*. The *word-level sentiment annotation* is to utilize the polarity annotation of words in each sentence and summarize the overall sentiment of each sentiment-bearing word to infer the overall sentiment within the text (Hatzivassiloglou and Wiebe, 2000; Andreevskaia and Bergler, 2006; Esuli and Sebastiani, 2005; Esuli and Sebastiani, 2006; Hatzivassiloglou and McKeown, 1997; Kamps et al., 2004; Devitt and Ahmad, 2007; Yu and Hatzivassiloglou, 2003). The *phrase-level sentiment annotation* focuses sentiment annotation on phrases not words with concerning that atomic units of expression is not individual words but rather appraisal groups (Whitelaw et al., 2005). In (Wilson et al., 2005), the concepts of *prior polarity* and *contextual polarity* were proposed. This paper presented a system that is able to automatically identify the *contextual polarity* for a large subset of sentiment expressions. In (Turney, 2002), an unsupervised learning algorithm was proposed to classify reviews as recommended or not recommended by averaging sentiment annotation of phrases in reviews that contain adjectives or adverbs. However, the performances of these works are not good enough for sentiment analysis on product reviews, where sentiment on each attribute of a product could be so complicated that it is unable to be expressed by overall document sentiment.

Attributes-based sentiment analysis is to analyze sentiment based on each attribute of a product. In (Hu and Liu, 2004), mining product features was proposed together with sentiment polarity annotation for each opinion sentence. In that work, sentiment analysis was performed on product attributes level. In (Liu et al., 2005), a system with framework for analyzing and comparing consumer opinions of competing products was proposed. The system made users be able to clearly see the strengths and weaknesses of each product in the minds of consumers in terms of various product features. In (Popescu and Etzioni, 2005), Popescu and Etzioni not only analyzed polarity of opinions regarding product features but also ranked opinions based on their strength. In (Liu et al., 2007), Liu et al. proposed Sentiment-PLSA that analyzed blog entries and viewed them as a document generated by a number of hidden sentiment factors. These sentiment factors may also be factors based on product attributes. In (Lu and Zhai, 2008), Lu et al. proposed a semi-supervised topic models to solve the problem of opinion integration based on the topic of a product's attributes. The work in (Titov and McDonald, 2008) presented a multi-grain topic model for extracting the ratable attributes from product reviews. In (Lu et al., 2009), the problem of rated attributes summary was studied with a goal of generating ratings for major aspects so that a user could gain different perspectives towards a target entity. All these research works concentrated on attribute-based sentiment analysis. However, the main difference with our work is that they did not sufficiently utilize the hierarchical relationships among a product attributes. Although a method of ontology-supported polarity mining, which also involved ontology to tackle the sentiment analysis problem, was proposed in (Zhou and Chaovalit, 2008), that work studied polarity mining by machine learning techniques that still suffered from a problem of ignoring dependencies among attributes within an ontology's hierarchy. In the contrast, our work solves the sentiment analysis problem as a hierarchical classification problem that fully utilizes the hierarchy of the SOT during training and classification process.

3 The HL-SOT Approach

In this section, we first propose a formal definition on SOT. Then we formulate the HL-SOT approach. In this novel approach, tasks of sentiment analysis are to be achieved in a hierarchical classification process.

3.1 Sentiment Ontology Tree

As we discussed in Section 1, the hierarchial relationships among a product's attributes might help improve the performance of attribute-based sentiment analysis. We propose to use a tree-like ontology structure SOT, i.e., Sentiment Ontology Tree, to formulate relationships among a product's attributes. Here,we give a formal definition on what a SOT is.

Definition 1 *[SOT] SOT is an abbreviation for Sentiment Ontology Tree that is a tree-like ontology structure $T(v, v^+, v^-, \mathbb{T})$. v is the root node of T which represents an attribute of a given product. v^+ is a positive sentiment leaf node associated with the attribute v. v^- is a negative sentiment leaf node associated with the attribute v. $\mathbb{T}$ is a set of subtrees. Each element of $\mathbb{T}$ is also a SOT $T'(v', v'^+, v'^-, \mathbb{T}')$ which represents a sub-attribute of its parent attribute node.*

By the Definition 1, we define a root of a SOT to represent an attribute of a product. The SOT's two leaf child nodes are sentiment (positive/negative) nodes associated with the root attribute. The SOT recursively contains a set of sub-SOTs where each root of a sub-SOT is a non-leaf child node of the root of the SOT and represent a sub-attribute belonging to its parent attribute. This definition successfully describes the hierarchical relationships among all the attributes of a product. For example, in Fig. 1 the root node of the SOT for a digital camera is its general overview attribute. Comments on a digital camera's general overview attribute appearing in a review might be like "this camera is great". The "camera" SOT has two sentiment leaf child nodes as well as three non-leaf child nodes which are respectively root nodes of sub-SOTs for sub-attributes "design and usability", "image quality", and "lens". These sub-attributes SOTs recursively repeat until each node in the SOT does not have any more non-leaf child node, which means the corresponding attributes do not have any sub-attributes, e.g., the attribute node "button" in Fig. 1.

3.2 Sentiment Analysis with SOT

In this subsection, we present the HL-SOT approach. With the defined SOT, the problem of sentiment analysis is able to be formulated to be a hierarchial classification problem. Then a specific hierarchical learning algorithm is further proposed to solve the formulated problem.

3.2.1 Problem Formulation

In the proposed HL-SOT approach, each target text is to be indexed by a unit-norm vector $x \in \mathcal{X}, \mathcal{X} = \mathbb{R}^d$. Let $\mathscr{Y} = \{1, ..., N\}$ denote the finite set of nodes in SOT. Let $y = \{y_1, ..., y_N\} \in \{0, 1\}^N$ be a label vector to a target text x, where $\forall i \in \mathscr{Y}$:

$$y_i = \begin{cases} 1, & \text{if } x \text{ is labeled by the classifier of node } i, \\ 0, & \text{if } x \text{ is not labeled by the classifier of node } i. \end{cases}$$

A label vector $y \in \{0, 1\}^N$ is said to respect SOT if and only if y satisfies $\forall i \in \mathscr{Y}, \forall j \in \mathcal{A}(i)$: if $y_i = 1$ then $y_j = 1$, where $\mathcal{A}(i)$ represents a set ancestor nodes of i, i.e.,$\mathcal{A}(i) = \{x|ancestor(i, x)\}$. Let $\mathcal{Y}$ denote a set of label vectors that respect SOT. Then the tasks of sentiment analysis can be formulated to be the goal of a hierarchical classification that is to learn a function $f : \mathcal{X} \to \mathcal{Y}$, that is able to label each target text $x \in \mathcal{X}$ with classifier of each node and generating with x a label vector $y \in \mathcal{Y}$ that respects SOT. The requirement of a generated label vector $y \in \mathcal{Y}$ ensures that a target text is to be labeled with a node only if its parent attribute node is labeled with the target text. For example, in Fig. 1 a review is to be labeled with "image quality +" requires that the review should be successively labeled as related to "camera" and "image quality". This is reasonable and consistent with intuition, because if a review cannot be identified to be related to a camera, it is not safe to infer that the review is commenting a camera's image quality with positive sentiment.

3.2.2 HL-SOT Algorithm

The algorithm H-RLS studied in (Cesa-Bianchi et al., 2006) solved a similar hierarchical classification problem as we formulated above. However, the H-RLS algorithm was designed as an online-learning algorithm which is not suitable to be applied directly in our problem setting. Moreover, the algorithm H-RLS defined the same value as the threshold of each node classifier. We argue that if the threshold values could be learned separately for each classifiers, the performance of classification process would be improved. Therefore we propose a specific hierarchical learning algorithm, named HL-SOT algorithm, that is able to train each node classifier in a batch-learning setting and allows separately learning for the threshold of each node classifier.

Defining the f function Let $w_1, ..., w_N$ be weight vectors that define linear-threshold classifiers of each node in SOT. Let $W = (w_1, ..., w_N)^\top$ be an $N \times d$ matrix called weight matrix. Here we generalize the work in (Cesa-Bianchi et al., 2006) and define the hierarchical classification function f as:

$$\hat{y} = f(x) = g(W \cdot x),$$

where $x \in \mathcal{X}, \hat{y} \in \mathcal{Y}$. Let $z = W \cdot x$. Then the function $\hat{y} = g(z)$ on an N-dimensional vector z defines:
$\forall i = 1, ..., N :$

$$\hat{y}_i = \begin{cases} \mathfrak{B}(z_i \geq \theta_i), & \text{if } i \text{ is a root node in SOT} \\ & \text{or } y_j = 1 \text{ for } j = \mathcal{P}(i), \\ 0, & \text{else} \end{cases}$$

where $\mathcal{P}(i)$ is the parent node of i in SOT and $\mathfrak{B}(S)$ is a boolean function which is 1 if and only if the statement S is true. Then the hierarchical classification function f is parameterized by the weight matrix $W = (w_1, ..., w_N)^\top$ and threshold vector $\theta = (\theta_1, ..., \theta_N)^\top$. The hierarchical learning algorithm HL-SOT is proposed for learning the parameters of W and θ.

Parameters Learning for f function Let D denote the training data set: $D = \{(r, l) | r \in \mathcal{X}, l \in \mathcal{Y}\}$. In the HL-SOT learning process, the weight matrix W is firstly initialized to be a 0 matrix, where each row vector w_i is a 0 vector. The threshold vector is initialized to be a 0 vector. Each instance in the training set D goes into the training process. When a new instance r_t is observed, each row vector $w_{i,t}$ of the weight matrix W_t is updated by a regularized least squares estimator given by:

$$w_{i,t} = (I + S_{i,Q(i,t-1)} S_{i,Q(i,t-1)}^\top + r_t r_t^\top)^{-1} \times S_{i,Q(i,t-1)} (l_{i,i_1}, l_{i,i_2}, ..., l_{i,i_{Q(i,t-1)}})^\top \quad (1)$$

where I is a $d \times d$ identity matrix, $Q(i, t-1)$ denotes the number of times the parent of node i observes a positive label before observing the instance r_t, $S_{i,Q(i,t-1)} = [r_{i_1}, ..., r_{i_{Q(i,t-1)}}]$ is a $d \times Q(i, t-1)$ matrix whose columns are the instances $r_{i_1}, ..., r_{i_{Q(i,t-1)}}$, and $(l_{i,i_1}, l_{i,i_2}, ..., l_{i,i_{Q(i,t-1)}})^\top$ is a $Q(i, t-1)$-dimensional vector of the corresponding labels observed by node i. The Formula 1 restricts that the weight vector $w_{i,t}$ of the classifier i is only updated on the examples that are positive for its parent node. Then the label vector $\hat{y}_{r_t}$ is computed for the instance r_t, before the real label vector l_{r_t} is observed. Then the current threshold vector θ_t is updated by:

$$\theta_{t+1} = \theta_t + \epsilon(\hat{y}_{r_t} - l_{r_t}), \quad (2)$$

where ϵ is a small positive real number that denotes a corrective step for correcting the current threshold vector θ_t. To illustrate the idea behind the Formula 2, let $y'_t = \hat{y}_{r_t} - l_{r_t}$. Let $y'_{i,t}$ denote an element of the vector y'_t. The Formula 2 correct the current threshold $\theta_{i,t}$ for the classifier i in the following way:

- If $y'_{i,t} = 0$, it means the classifier i made a proper classification for the current instance r_t. Then the current threshold θ_i does not need to be adjusted.

- If $y'_{i,t} = 1$, it means the classifier i made an improper classification by mistakenly identifying the attribute i of the training instance r_t that should have not been identified. This indicates the value of θ_i is not big enough to serve as a threshold so that the attribute i in this case can be filtered out by the classifier i. Therefore, the current threshold θ_i will be adjusted to be larger by ϵ.

- If $y'_{i,t} = -1$, it means the classifier i made an improper classification by failing to identify the attribute i of the training instance r_t that should have been identified. This indicates the value of θ_i is not small enough to serve as a threshold so that the attribute i in this case

Algorithm 1 Hierarchical Learning Algorithm HL-SOT

```
    INITIALIZATION:
1:  Each vector $w_{i,1}, i = 1, ..., N$ of weight matrix $W_1$ is set to be 0 vector
2:  Threshold vector $\theta_1$ is set to be 0 vector
    BEGIN
3:  for $t = 1, ..., |D|$ do
4:      Observe instance $r_t \in \mathcal{X}$
5:      for $i = 1, ...N$ do
6:          Update each row $w_{i,t}$ of weight matrix $W_t$ by Formula 1
7:      end for
8:      Compute $\hat{y}_{r_t} = f(r_t) = g(W_t \cdot r_t)$
9:      Observe label vector $l_{r_t} \in \mathcal{Y}$ of the instance $r_t$
10:     Update threshold vector $\theta_t$ by Formula 2
11: end for
    END
```

can be recognized by the classifier i. Therefore, the current threshold θ_i will be adjusted to be smaller by ϵ.

The hierarchial learning algorithm HL-SOT is presented as in Algorithm 1. The HL-SOT algorithm enables each classifier to have its own specific threshold value and allows this threshold value can be separately learned and corrected through the training process. It is not only a batch-learning setting of the H-RLS algorithm but also a generalization to the latter. If we set the algorithm HL-SOT's parameter ϵ to be 0, the HL-SOT becomes the H-RLS algorithm in a batch-learning setting.

4 Empirical Analysis

In this section, we conduct systematic experiments to perform empirical analysis on our proposed HL-SOT approach against a human-labeled data set. In order to encode each text in the data set by a d-dimensional vector $x \in \mathbb{R}^d$, we first remove all the stop words and then select the top d frequency terms appearing in the data set to construct the index term space. Our experiments are intended to address the following questions:(1) whether utilizing the hierarchical relationships among labels help to improve the accuracy of the classification? (2) whether the introduction of separately learning threshold for each classifier help to improve the accuracy of the classification? (3) how does the corrective step ϵ impact the performance of the proposed approach?(4)how does the dimensionality d of index terms space impact the proposed approach's computing efficiency and accuracy?

4.1 Data Set Preparation

The data set contains 1446 snippets of customer reviews on digital cameras that are collected from a customer review website[4]. We manually construct a SOT for the product of digital cameras. The constructed SOT (e.g., Fig. 1) contains 105 nodes that include 35 non-leaf nodes representing attributes of the digital camera and 70 leaf nodes representing associated sentiments with attribute nodes. Then we label all the snippets with corresponding labels of nodes in the constructed SOT complying with the rule that a target text is to be labeled with a node only if its parent attribute node is labeled with the target text. We randomly divide the labeled data set into five folds so that each fold at least contains one example snippets labeled by each node in the SOT. For each experiment setting, we run 5 experiments to perform cross-fold evaluation by randomly picking three folds as the training set and the other two folds as the testing set. All the testing results are averages over 5 running of experiments.

4.2 Evaluation Metrics

Since the proposed HL-SOT approach is a hierarchical classification process, we use three classic loss functions for measuring classification performance. They are the One-error Loss (O-Loss) function, the Symmetric Loss (S-Loss) function, and the Hierarchical Loss (H-Loss) function:

- One-error loss (O-Loss) function is defined as:
$$L_O(\hat{y}, l) = \mathfrak{B}(\exists i : \hat{y}_i \neq l_i),$$
where $\hat{y}$ is the prediction label vector and l is the true label vector; $\mathfrak{B}$ is the boolean function as defined in Section 3.2.2.

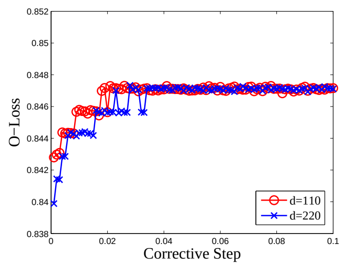
- Symmetric loss (S-Loss) function is defined as:
$$L_S(\hat{y}, l) = \sum_{i=1}^{N} \mathfrak{B}(\hat{y}_i \neq l_i),$$

- Hierarchical loss (H-Loss) function is defined as:
$$L_H(\hat{y}, l) = \sum_{i=1}^{N} \mathfrak{B}(\hat{y}_i \neq l_i \wedge \forall j \in \mathcal{A}(i), \hat{y}_j = l_j),$$

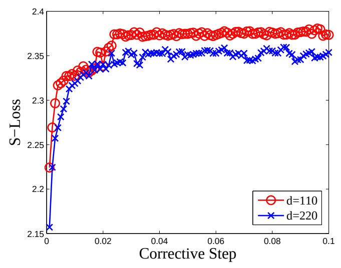
[4]http://www.consumerreview.com/

Table 1: Performance Comparisons (A Smaller Loss Value Means a Better Performance)

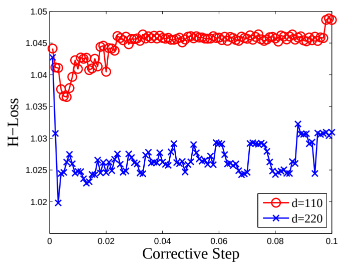
Metrics	Dimensinality=110			Dimensinality=220		
	H-RLS	HL-flat	HL-SOT	H-RLS	HL-flat	HL-SOT
O-Loss	0.9812	0.8772	**0.8443**	0.9783	0.8591	**0.8428**
S-Loss	8.5516	2.8921	**2.3190**	7.8623	2.8449	**2.2812**
H-Loss	3.2479	1.1383	**1.0366**	3.1029	1.1298	**1.0247**



(a) O-Loss



(b) S-Loss



(c) H-Loss

Figure 2: Impact of Corrective Step ϵ

where $\mathcal{A}$ denotes a set of nodes that are ancestors of node i in SOT.

Unlike the O-Loss function and the S-Loss function, the H-Loss function captures the intuition that loss should only be charged on a node whenever a classification mistake is made on a node of SOT but no more should be charged for any additional mistake occurring in the subtree of that node. It measures the discrepancy between the prediction labels and the true labels with consideration on the SOT structure defined over the labels. In our experiments, the recorded loss function values for each experiment running are computed by averaging the loss function values of each testing snippets in the testing set.

4.3 Performance Comparison

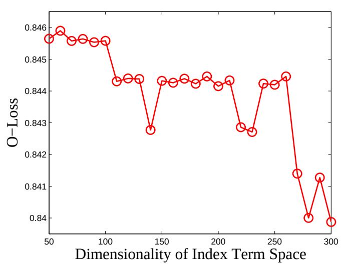
In order to answer the questions (1), (2) in the beginning of this section, we compare our HL-SOT approach with the following two baseline approaches:

- HL-flat: The HL-flat approach involves an algorithm that is a "flat" version of HL-SOT algorithm by ignoring the hierarchical relationships among labels when each classifier is trained. In the training process of HL-flat, the algorithm reflexes the restriction in the HL-SOT algorithm that requires the weight vector $w_{i,t}$ of the classifier i is only updated on the examples that are positive for its parent node.
- H-RLS: The H-RLS approach is implemented by applying the H-RLS algorithm studied in (Cesa-Bianchi et al., 2006). Unlike our proposed HL-SOT algorithm that enables the threshold values to be learned separately for each classifiers in the training process, the H-RLS algorithm only uses an identical threshold values for each classifiers in the classification process.

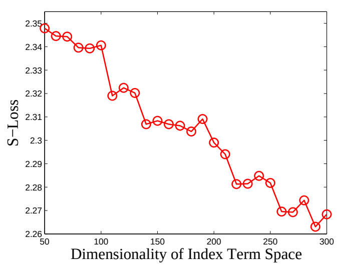
Experiments are conducted on the performance comparison between the proposed HL-SOT approach with HL-flat approach and the H-RLS approach. The dimensionality d of the index term space is set to be 110 and 220. The corrective step ϵ is set to be 0.005. The experimental results are summarized in Table 1. From Table 1, we can observe that the HL-SOT approach generally beats the H-RLS approach and HL-flat approach on O-Loss, S-Loss, and H-Loss respectively. The H-RLS performs worse than the HL-flat and the HL-SOT, which indicates that the introduction of separately learning threshold for each classifier did improve the accuracy of the classification. The HL-SOT approach performs better than the HL-flat, which demonstrates the effectiveness of utilizing the hierarchical relationships among labels.

4.4 Impact of Corrective Step ϵ

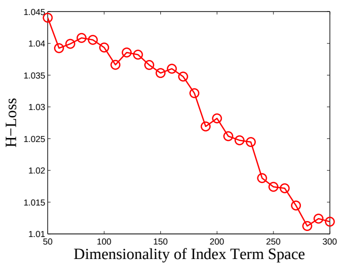
The parameter ϵ in the proposed HL-SOT approach controls the corrective step of the classifiers' thresholds when any mistake is observed in the training process. If the corrective step ϵ is set too large, it might cause the algorithm to be too



(a) O-Loss



(b) S-Loss



(c) H-Loss

Figure 3: Impact of Dimensionality d of Index Term Space ($\epsilon = 0.005$)

sensitive to each observed mistake. On the contrary, if the corrective step is set too small, it might cause the algorithm not sensitive enough to the observed mistakes. Hence, the corrective step ϵ is a factor that might impact the performance of the proposed approach. Fig. 2 demonstrates the impact of ϵ on O-Loss, S-Loss, and H-Loss. The dimensionality of index term space d is set to be 110 and 220. The value of ϵ is set to vary from 0.001 to 0.1 with each step of 0.001. Fig. 2 shows that the parameter ϵ impacts the classification performance significantly. As the value of ϵ increase, the O-Loss, S-Loss, and H-Loss generally increase (performance decrease). In Fig. 2c it is obviously detected that the H-Loss decreases a little (performance increase) at first before it increases (performance decrease) with further increase of the value of ϵ. This indicates that a finer-grained value of ϵ will not necessarily result in a better performance on the H-loss. However, a fine-grained corrective step generally makes a better performance than a coarse-grained corrective step.

4.5 Impact of Dimensionality d of Index Term Space

In the proposed HL-SOT approach, the dimensionality d of the index term space controls the number of terms to be indexed. If d is set too small, important useful terms will be missed that will limit the performance of the approach. However, if d is set too large, the computing efficiency will be decreased. Fig. 3 shows the impacts of the parameter d respectively on O-Loss, S-Loss, and H-Loss, where d varies from 50 to 300 with each step of 10 and the ϵ is set to be 0.005. From Fig. 3, we observe that as the d increases the O-Loss, S-Loss, and H-Loss generally decrease (performance increase). This means that when more terms are indexed better performance can be achieved by the HL-SOT approach. However,

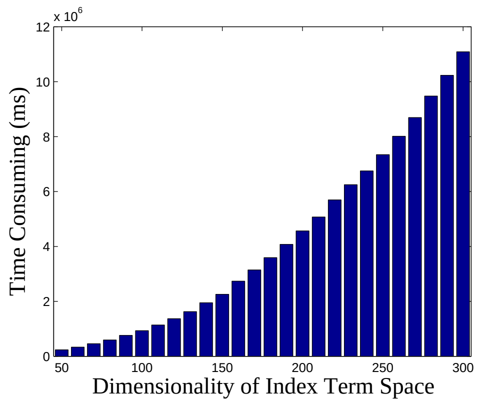


Figure 4: Time Consuming Impacted by d

considering the computing efficiency impacted by d, Fig. 4 shows that the computational complexity of our approach is non-linear increased with d's growing, which indicates that indexing more terms will improve the accuracy of our proposed approach although this is paid by decreasing the computing efficiency.

5 Conclusions, Discussions and Future Work

In this paper, we propose a novel and effective approach to sentiment analysis on product reviews. In our proposed HL-SOT approach, we define SOT to formulate the knowledge of hierarchical relationships among a product's attributes and tackle the problem of sentiment analysis in a hierarchical classification process with the proposed algorithm. The empirical analysis on a human-labeled data set demonstrates the promising results of our proposed approach. The performance comparison shows that the proposed HL-SOT approach outperforms two baselines: the HL-flat and the H-RLS approach. This confirms two intuitive motivations based on which our approach is proposed: 1) separately learning threshold values for each classifier improve the classification accuracy; 2) knowledge of hierarchical relationships of labels improve the approach's performance. The experiments on analyzing the impact of parameter ϵ indicate that a fine-grained corrective step generally makes a better performance than a coarse-grained corrective step. The experiments on analyzing the impact of the dimensionality d show that indexing more terms will improve the accuracy of our proposed approach while the computing efficiency will be greatly decreased.

The focus of this paper is on analyzing review texts of one product. However, the framework of our proposed approach can be generalized to deal with a mix of review texts of more than one products. In this generalization for sentiment analysis on multiple products reviews, a "big" SOT is constructed and the SOT for each product reviews is a sub-tree of the "big" SOT. The sentiment analysis on multiple products reviews can be performed the same way the HL-SOT approach is applied on single product reviews and can be tackled in a hierarchical classification process with the "big" SOT.

This paper is motivated by the fact that the relationships among a product's attributes could be a useful knowledge for mining product review texts. The SOT is defined to formulate this knowledge in the proposed approach. However, what attributes to be included in a product's SOT and how to structure these attributes in the SOT is an effort of human beings. The sizes and structures of SOTs constructed by different individuals may vary. How the classification performance will be affected by variances of the generated SOTs is worthy of study. In addition, an automatic method to learn a product's attributes and the structure of SOT from existing product review texts will greatly benefit the efficiency of the proposed approach. We plan to investigate on these issues in our future work.

Acknowledgments

The authors would like to thank the anonymous reviewers for many helpful comments on the manuscript. This work is funded by the Research Council of Norway under the VERDIKT research programme (Project No.: 183337).

References

Alina Andreevskaia and Sabine Bergler. 2006. Mining wordnet for a fuzzy sentiment: Sentiment tag extraction from wordnet glosses. In *Proceedings of 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL'06)*, Trento, Italy.

Nicolò Cesa-Bianchi, Claudio Gentile, and Luca Zaniboni. 2006. Incremental algorithms for hierarchical classification. *Journal of Machine Learning Research (JMLR)*, 7:31–54.

Kushal Dave, Steve Lawrence, and David M. Pennock. 2003. Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In *Proceedings of 12nd International World Wide Web Conference (WWW'03)*, Budapest, Hungary.

Ann Devitt and Khurshid Ahmad. 2007. Sentiment polarity identification in financial news: A cohesion-based approach. In *Proceedings of 45th Annual Meeting of the Association for Computational Linguistics (ACL'07)*, Prague, Czech Republic.

Xiaowen Ding and Bing Liu. 2007. The utility of linguistic rules in opinion mining. In *Proceedings of 30th Annual International ACM Special Interest Group on Information Retrieval Conference (SIGIR'07)*, Amsterdam, The Netherlands.

Andrea Esuli and Fabrizio Sebastiani. 2005. Determining the semantic orientation of terms through gloss classification. In *Proceedings of 14th ACM Conference on Information and Knowledge Management (CIKM'05)*, Bremen, Germany.

Andrea Esuli and Fabrizio Sebastiani. 2006. Sentiwordnet: A publicly available lexical resource for opinion mining. In *Proceedings of 5th International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy.

Vasileios Hatzivassiloglou and Kathleen R. McKeown. 1997. Predicting the semantic orientation of adjectives. In *Proceedings of 35th Annual Meeting of the Association for Computational Linguistics (ACL'97)*, Madrid, Spain.

Vasileios Hatzivassiloglou and Janyce M. Wiebe. 2000. Effects of adjective orientation and gradability on sentence subjectivity. In *Proceedings of 18th International Conference on Computational Linguistics (COLING'00)*, Saarbrüken, Germany.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of 10th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'04)*, Seattle, USA.

Jaap Kamps, Maarten Marx, R. ort. Mokken, and Maarten de Rijke. 2004. Using WordNet to measure semantic orientation of adjectives. In *Proceedings of 4th International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal.

Bing Liu, Minqing Hu, and Junsheng Cheng. 2005. Opinion observer: analyzing and comparing opinions on the web. In *Proceedings of 14th International World Wide Web Conference (WWW'05)*, Chiba, Japan.

Yang Liu, Xiangji Huang, Aijun An, and Xiaohui Yu. 2007. ARSA: a sentiment-aware model for predicting sales performance using blogs. In *Proceedings of the 30th Annual International ACM Special Interest Group on Information Retrieval Conference (SIGIR'07)*, Amsterdam, The Netherlands.

Yue Lu and Chengxiang Zhai. 2008. Opinion integration through semi-supervised topic modeling. In *Proceedings of 17th International World Wide Web Conference (WWW'08)*, Beijing, China.

Yue Lu, ChengXiang Zhai, and Neel Sundaresan. 2009. Rated aspect summarization of short comments. In *Proceedings of 18th International World Wide Web Conference (WWW'09)*, Madrid, Spain.

Ana-Maria Popescu and Oren Etzioni. 2005. Extracting product features and opinions from reviews. In *Proceedings of Human Language Technology Conference and Empirical Methods in Natural Language Processing Conference (HLT/EMNLP'05)*, Vancouver, Canada.

Ivan Titov and Ryan T. McDonald. 2008. Modeling online reviews with multi-grain topic models. In *Proceedings of 17th International World Wide Web Conference (WWW'08)*, Beijing, China.

Peter D. Turney. 2002. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics (ACL'02)*, Philadelphia, USA.

Casey Whitelaw, Navendu Garg, and Shlomo Argamon. 2005. Using appraisal taxonomies for sentiment analysis. In *Proceedings of 14th ACM Conference on Information and Knowledge Management (CIKM'05)*, Bremen, Germany.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of Human Language Technology Conference and Empirical Methods in Natural Language Processing Conference (HLT/EMNLP'05)*, Vancouver, Canada.

Hong Yu and Vasileios Hatzivassiloglou. 2003. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of 8th Conference on Empirical Methods in Natural Language Processing (EMNLP'03)*, Sapporo, Japan.

Lina Zhou and Pimwadee Chaovalit. 2008. Ontology-supported polarity mining. *Journal of the American Society for Information Science and Technology (JASIST)*, 59(1):98–110.

Li Zhuang, Feng Jing, and Xiao-Yan Zhu. 2006. Movie review mining and summarization. In *Proceedings of the 15th ACM International Conference on Information and knowledge management (CIKM'06)*, Arlington, USA.