

# Enhancing the HL-SOT Approach to Sentiment Analysis via a Localized Feature Selection Framework

**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

In this paper, we propose a Localized Feature Selection (LFS) framework tailored to the HL-SOT approach to sentiment analysis. Within the proposed LFS framework, each node classifier of the HL-SOT approach is able to perform classification on target texts in a locally customized index term space. Extensive empirical analysis against a human-labeled data set demonstrates that with the proposed LFS framework the classification performance of the HL-SOT approach is enhanced with computational efficiency being greatly gained. To find the best feature selection algorithm that caters to the proposed LFS framework, five classic feature selection algorithms are comparatively studied, which indicates that the TS, DF, and MI algorithms achieve generally better performances than the CHI and IG algorithms. Among the five studied algorithms, the TS algorithm is best to be employed by the proposed LFS framework.

## 1 Introduction

With tens and thousands of review texts being generated online, it becomes increasingly challenging for an individual to exhaustively collect and study the online reviews. Therefore, research on automatic sentiment analysis on review texts has emerged as a popular topic at the crossroads of information retrieval and computational linguistics.

Sentiment analysis on product reviews aims at extracting sentiment information from texts. It includes two tasks, i.e., labeling a target text<sup>1</sup> with

<sup>1</sup>Each product review to be analyzed is called target text

1) the product's attributes it mentions (attributes identification task), and 2) the corresponding sentiments mentioned therein (sentiment annotation task). Recently, Wei and Gulla proposed the HL-SOT approach (Wei and Gulla, 2010), i.e., Hierarchical Learning (HL) with Sentiment Ontology Tree (SOT), that is able to achieve the two tasks in one hierarchical classification process. In the HL-SOT approach, each target text is encoded by a vector in a globally unified  $d$ -dimensional index term space and is respectively labeled by different nodes<sup>2</sup> of SOT in a hierarchical manner. Although the HL-SOT approach is reported with promising classification performance on tasks of sentiment analysis, its computational efficiency, especially as  $d$  increases, becomes very low. Furthermore, as  $d$  increases it will have more chance to index noisy term into the globally unified index term space so that the classification performance of the HL-SOT approach might be depressed. Hence, we argue that if a locally customized index term space could be constructed for each node respectively, both the computational efficiency and the classification performance of the HL-SOT approach would be improved.

In this paper, we propose a Localized Feature Selection (LFS) framework tailored to the HL-SOT approach. The rationale of the proposed LFS framework draws on the following two observations. Firstly, a feature term that is relevant to a node is usually irrelevant to nodes which stay at another branch of SOT. For example, “ergonomics” might be a feature term for the node “design and usability” (see Fig. 1) but it is irrelevant to the node “image quality”. Secondly, a feature ter-

in the following of this paper.

<sup>2</sup>If specified otherwise in the following of this paper the term “node” refers to the classifier of the node.

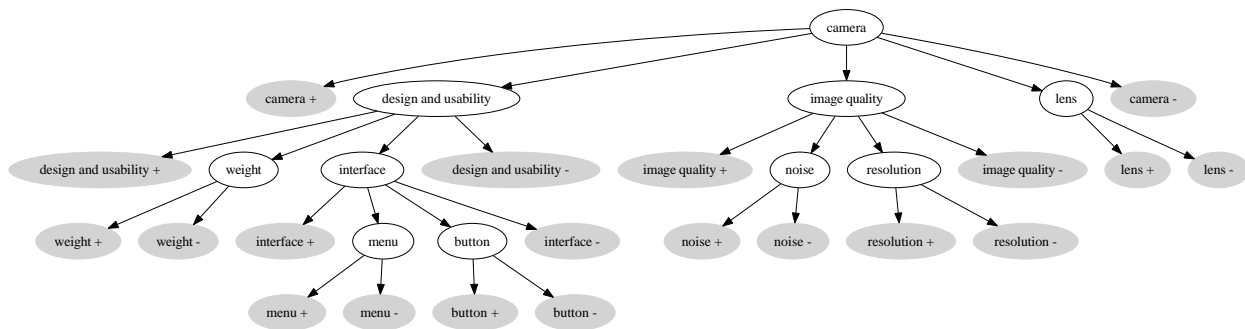


Figure 1: an example of part of a SOT for digital camera

m might become insignificant for child nodes of  $i$  even if the feature term is significant to  $i$ . For example, for a sentence commenting on a digital camera like “40D handles noise very well”, terms such as “noise” and “well” are significant feature terms for the node “noise”. However, the term “noise” becomes insignificant for its child nodes “noise +” and “noise -”, since the hierarchical classification characteristic of the HL-SOT approach that a node only processes target texts which are labeled as true by its parent node ensures that each target text handled by the nodes “noise +” and “noise -” is already classified as related to “noise”.

In the proposed LFS framework, the concept of “local hierarchy” is defined and introduced as delimitation of local scope of nodes. The localized feature selection process is conducted for each node within its local scope to generate the customized index term space for the node. The proposed LFS framework is empirically analyzed on a human-labeled data set. The experimental results show that with the proposed LFS framework the classification performance of the HL-SOT approach is enhanced and the computational efficiency is significantly improved. To test which is the best to be employed by the proposed LFS framework, we further comparatively study five classic feature selection algorithms respectively based on document frequency (DF) (Manning et al., 2008), mutual information (MI) (Manning et al., 2008; Church and Hanks, 1990),  $\chi^2$ -statistic (CHI) (Manning et al., 2008), information gain (IG) (Mitchell, 1997), and term strength (TS) (Wilbur and Sirotkin, 1992). The comparatively experimental results suggest that the TS, DF, and MI algorithms achieve generally better performance than the CHI and IG algorithms. Among the five employed algorithms, the TS algorithm is the best to be employed by the proposed LFS

framework. This paper makes the following contributions:

- We propose a LFS framework to enhance the classification performance and improve the computational efficiency of the HL-SOT approach;
- We conduct a comparative study on five feature selection algorithms that can be employed in the proposed LFS.

The remainder of the paper is organized as follows. In section 2, we discuss an overview of related work on sentiment analysis. In section 3, we review the HL-SOT approach proposed in (Wei and Gulla, 2010). In section 4, we present the proposed LFS framework. The empirical analysis and the results are presented in section 5. Finally, we conclude the paper and discuss the future work in section 6.

## 2 Related Work

Research on sentiment analysis was originally performed to extract overall sentiments from target texts. However, as shown in the experiments in (Turney, 2002), the whole sentiment of a document is not necessarily the sum of its parts. Recent work has shifted the 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, which relies on two finer levels of 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 approaches are not satisfactory 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 at the 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. In a most recent research work (Wei and Gulla, 2010), Wei and Gulla proposed the HL-SOT approach that sufficiently utilizes the hierarchical relationships among a product attributes and solves the sentiment analysis problem in a hierarchical classification process. However, the HL-SOT approach proposed in (Wei and Gulla, 2010) uses a globally unified index term space to encode target texts for different nodes which is deemed to limit the performance of the HL-SOT approach. Therefore, the LFS framework proposed in this paper aims at overcoming the weakness of the HL-SOT approach and consequently improving its performance by generating a locally customized index term space for each node.

### 3 The HL-SOT Approach Review

In the HL-SOT approach (Wei and Gulla, 2010), each target text is indexed by a vector  $x \in \mathcal{X}$ ,  $\mathcal{X} = \mathbb{R}^d$ . Weight vectors  $w_i (1 \leq i \leq N)$  define linear-threshold classifiers of each node  $i$  in SOT so that the target text  $x$  is labeled true by node  $i$  if  $x$  is labeled true by  $i$ ’s parent node and  $w_i \cdot x \geq \theta_i$ . The parameters  $w_i$  and  $\theta_i$  are learned from the training data set:  $D = \{(r, l) | r \in \mathcal{X}, l \in \mathcal{Y}\}$ , where  $\mathcal{Y}$  denotes the set of label vectors. In the training process, when a new instance  $r_t$  is observed, each row vector  $w_{i,t}$  is updated by a regularized least squares estimator given by:

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

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  $r_t$ ,  $S_{i,Q(i,t-1)} = [r_{i_1}, \dots, r_{i_{Q(i,t-1)}}]$  is a  $d \times Q(i, t - 1)$  matrix whose columns are the instances  $r_{i_1}, \dots, r_{i_{Q(i,t-1)}}$ , and  $(l_{i,i_1}, l_{i,i_2}, \dots, l_{i,i_{Q(i,t-1)}})^\top$  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  $\theta_t$  is updated by:

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

where  $\epsilon$  is a small positive real number that denotes a corrective step for the current threshold vector  $\theta_t$ . After the training process for each node of SOT, each target text is to be labeled by each node  $i$  parameterized by the weight vector  $w_i$  and the threshold  $\theta_i$  in the hierarchical classification process.

## 4 The Localized Feature Selection

In this section, we propose the LFS framework to generate a locally customized index term space for each node of SOT respectively. We first discuss why localized feature selection is needed for the HL-SOT approach. Then we define the concept of local hierarchy of SOT to introduce the local feature selection scope of a node, followed by a presentation on the local hierarchy based feature selection process.

### 4.1 Why Localized Feature Selection for the HL-SOT

One deficiency of the HL-SOT approach is that it uses a globally unified index term space to index target texts, which cannot efficiently encode feature information required by each local individual node of SOT. When we look into the detailed classification process of each node of SOT, we observe the following two types of phenomena. Firstly, SOT organizes domain knowledge in a tree like structure. Within a particular domain knowledge represented by SOT, nodes that stay in different branches of SOT represent independent different attributes in that domain. In this way, feature terms (e.g., the term “ergonomics”) that are relevant to a node (e.g., the node “design and usability”) might be irrelevant to other nodes (e.g., the node “image quality”) that stay at another branches of SOT; Secondly, the HL-SOT approach labels each target text in a hierarchical order which ensures that each target text that comes to be handled by a node has already been labeled as true by its parent node. Due to this characteristic, feature terms (e.g., the term “noise”) that are significant to a node  $i$  (e.g., the node “noise”) might become a trivial term for  $i$ 's child nodes (e.g., the nodes “noise +” and “noise -”). Therefore, the purpose of the localized feature selection is to filter out irrelevant terms that are insignificant to each individual node and build a locally customized index term space for the node so that the performance of the node can be improved.

### 4.2 Local Feature Selection Scope for a Node

In order to select locally customized feature terms for each individual node, we need to define a suitable scope, called local feature selection scope<sup>3</sup>, within which the feature selection process can be effectively conducted for the node. Since the HL-SOT approach is a hierarchical classification process, before we introduce the local scope for a node we first give a formal definition on local hierarchy of SOT.

**Definition 1** [*Local Hierarchy*] A local hierarchy  $\Delta_u$  of SOT is defined to be formed by all the child nodes of  $u$  in SOT, where the node  $u$  must be a non-leaf node of the SOT.

By the Definition 1, we say all the child nodes of  $u$  are on the same local hierarchy under  $u$  which is denoted by  $\Delta_u$ . For examples, in Fig. 2 nodes “camera +”, “design and usability”, “image quality”, “lens”, “camera -” are deemed on the same local hierarchy under the node “camera” and nodes “weight +”, “weight -” are deemed on the same local hierarchy under the node “weight”, etc. In the hierarchical labeling process of the HL-SOT approach, after a target text is labeled as true by a node  $i$  it will go further to the local hierarchy under  $i$  and is to be labeled by all nodes on the local hierarchy  $\Delta_i$ . For a target text the labeling processes of nodes on  $\Delta_i$  locally can be considered as a multi-label classification process where each node is a local label. Therefore, the measurement for selecting terms as features should be calculated among nodes on the same hierarchy. Hence, the local scope for a node is defined within the local hierarchy which the node is on.

### 4.3 Local Hierarchy Based Feature Selection

In the proposed LFS framework, local feature selection for a node  $i$  of SOT is performed within the *local scope* of the node  $i$ . Since nodes on the same local hierarchy share the same local scope, local feature selection process for all nodes of SOT is achieved in local hierarchy based manner. Specifically, for the feature selection process on a local hierarchy  $\Delta$ , let  $c_1, c_2, \dots, c_K$  denote the  $K$  nodes on  $\Delta$ . Let  $D$  denote the training data set for the HL-SOT approach. Let  $D_{c_k}$  denote the set of instances in  $D$  that contains the label of the node  $c_k$  ( $1 \leq k \leq K$ ). Let  $D_\Delta$  denote the training corpus for the local hierarchy  $\Delta$  which is the set of

<sup>3</sup>In this paper, we also call it “local scope” for short.

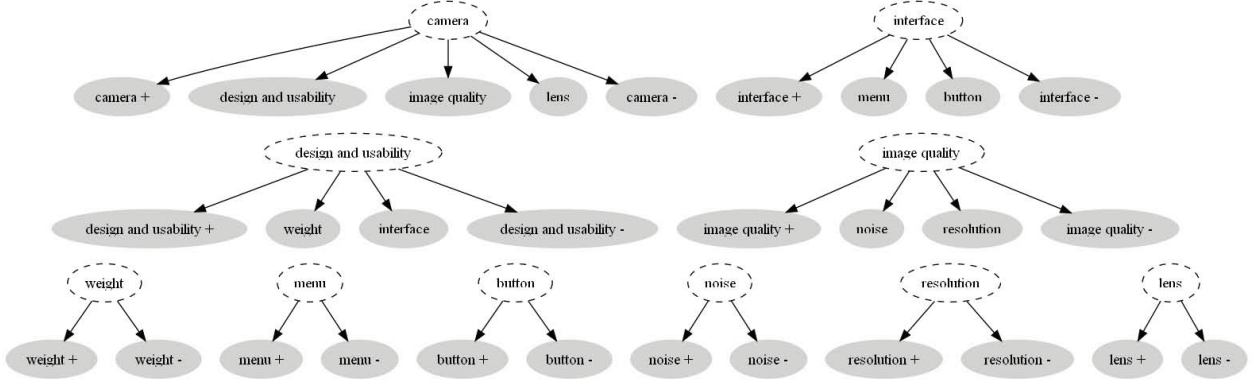


Figure 2: All local hierarchies of the example SOT: the grey nodes sharing the same parent node in dashed line are called on the same local hierarchy under the parent node

all instances in the training data set  $D$  that contain any label of nodes on the local hierarchy  $\Delta$ :  $D_\Delta = \bigcup_{k=1}^K D_{c_k}$ . Let  $V_{c_k}$  denote the set of all the vocabularies that appears in  $D_{c_k}$ . Let  $s_{c_k}(w)$  denote the term score that measures the suitability of  $w$  as a feature for node  $c_k$ . Let  $F_{c_k}$  denote the set of feature terms selected for  $c_k$ . Let  $d_{c_k}$  denote the number of features to be selected in  $F_{c_k}$ . A local feature selection process for nodes on the local hierarchy  $\Delta$  is described in Algorithm 1.

---

**Algorithm 1** Localized Feature Selection Algorithm

---

**DATA INITIALIZATION:**  
1: **for** each node  $c_k$  on  $\Delta$  **do**  
2:   Establish  $D_{c_k}$  containing instances being labeled true by  $c_k$ ;  
3:   Establish the vocabulary set  $V_{c_k}$ ;  
4:   Remove stop words from  $V_{c_k}$ ;  
5: **end for**  
6: Establish the training corpus:  $D_\Delta = \bigcup_{k=1}^K D_{c_k}$ ;  
**BEGIN**  
7: **for** each node  $c_k$  on  $\Delta$  **do**  
8:   **for** each term  $w \in V_{c_k}$  **do**  
9:     with training corpus  $D_\Delta$  and data instance set  $D_{c_k}$ ;  
10:     Calculate  $s_{c_k}(w)$  with a specified feature selection algorithm;  
11:   **end for**  
12:   Establish feature space  $F_{c_k}$  with top  $d_{c_k}$  terms from  $V_{c_k}$ ;  
13: **end for**  
**END**

---

In the data initialization phase of the Algorithm 1, the data instance set  $D_{c_k}$  and vocabulary set  $V_{c_k}$  for each node on the local hierarchy  $\Delta$  as well as the training corpus  $D_\Delta$  are established. In a local feature selection process, a term score  $s_{c_k}(w)$  for each term  $w \in V_{c_k}$  can be calculated by a specified feature selection algorithm, taking  $D_\Delta$  as the training corpus and  $D_{c_k}$  as the data instance set in the class  $c_k$ . The local feature selection process can employ any specific feature selection algorithm to calculate the term scores. After all terms in  $V_{c_k}$  are calculated, those terms with top  $d_{c_k}$  scores are selected to establish the feature space  $F_{c_k}$  for

the node  $c_k$ . Since the number of terms in  $V_{c_k}$  varies from node to node, in order to produce a rational dimensionality  $d_{c_k}$  for the established feature space  $F_{c_k}$ , we introduce a feature selection rate, denoted by  $\gamma$ , to control  $d_{c_k}$  for each node  $c_k$ , i.e.,  $d_{c_k} = \lceil |V_{c_k}| \times \gamma \rceil$ .

After local feature selection processes for all the nodes of SOT are accomplished, a locally customized index term space  $F_{c_k}$  for each node  $c_k$  is established. Each target text will be respectively indexed by a customized vector  $x_{c_k} \in \mathcal{X}_{c_k}$  ( $\mathcal{X}_{c_k} = \mathbb{R}^{d_{c_k}}$ ) when it goes through the hierarchical classification process of the HL-SOT approach. In next section, we will present the empirical analysis on evaluating the proposed LFS framework.

## 5 Empirical Analysis

In this section, we conduct extensive experiments to empirically analyze the proposed LFS framework. Our experiments are intended to address the following questions: (1) can the classification performance of the HL-SOT approach be improved with the LFS framework; (2) how much computational efficiency can be gained for the HL-SOT to be implemented in the LFS framework; (3) how are the comparative performances produced by different feature selection algorithms when employed in the proposed LFS framework.

### 5.1 Data Set Preparation

We construct our data set based on the digital camera review data set used in the HL-SOT approach (Wei and Gulla, 2010). In total, the constructed data set contains 1500 snippets of customer reviews on digital cameras, where 35 attributes of a digital camera are mentioned in the

review data. We build an ontology structure to organize the mentioned attributes and label each review text with correspondent attributes as well as sentiments, which complying the rule that if a review text is assigned with a label of a node then it is assigned with a label of the parent node. We randomly divide the labeled data set into five folds so that each fold at least contains one example instance labeled by each attribute node. To catch the statistical significance of experimental results, we perform 5 cross-fold evaluation by using four folds as training data and the other one fold as testing data. All the experimental results presented in this section are averaged over 5 runs of each experiment.

## 5.2 Evaluation Metrics

Since the existing HL-SOT approach is a hierarchical classification process, we use the same three classic loss functions (Wei and Gulla, 2010) for measuring classification performance. They are respectively the One-error Loss (O-Loss) function, the Symmetric Loss (S-Loss) function, and the Hierarchical Loss (H-Loss) function<sup>4</sup>:

- One-error loss (O-Loss) function is defined as:

$$L_O(\hat{y}, l) = \mathfrak{B}(\exists i : \hat{y}_i \neq l_i), \quad (3)$$

where  $\hat{y}$  is the prediction label vector and  $l$  is the true label vector;  $\mathfrak{B}(S)$  is a boolean function which is 1 if and only if the statement  $S$  is true, otherwise it is 0.

- Symmetric loss (S-Loss) function is defined as:

$$L_S(\hat{y}, l) = \sum_{i=1}^N \mathfrak{B}(\hat{y}_i \neq l_i), \quad (4)$$

- Hierarchical loss (H-Loss) function is defined as:

$$L_H(\hat{y}, l) = \sum_{i=1}^N \mathfrak{B}(\hat{y}_i \neq l_i \wedge \forall j \in \mathcal{A}(i), \hat{y}_j = l_j), \quad (5)$$

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

## 5.3 Performance Comparison

In this section, we conduct experiments to show performance improvement from the proposed LFS framework. The performance considered here include both classification performance and computational efficiency. We use the existing HL-SOT approach as a baseline. Since the HL-SOT

<sup>4</sup>Since the three loss functions are respectively well-defined by each formula and self-explained by their names, due to the space limitation, we do not present more explanation.

approach used terms' document frequencies (DF) (Manning et al., 2008) algorithm to select features to build the globally unified index term space, employing the same DF feature selection algorithm we apply the proposed LFS framework on the HL-SOT approach and call the implemented method "DF-SOT". The only difference between HL-SOT and DF-SOT is the index term space for each node of SOT, i.e., in the HL-SOT all the nodes using the globally unified index term space while in the DF-SOT each node respectively using a locally customized index term space. In this way, the performance difference between the two methods will indicate the effect of the proposed LFS framework.

### 5.3.1 Comparison on Classification Performance

We conduct experiments to investigate whether the classification performance of the HL-SOT can be improved when it is implemented with the LFS framework. Fig. 3 presents the experimental results of classification accuracies between HL-SOT and DF-SOT. In the experiments, the dimensionality  $d$  of the globally unified index term space of the HL-SOT approach is set to 270, which is large enough for the HL-SOT approach to reach its best performance level. The feature selection rate  $\gamma$  for the locally customized index term space of the DF-SOT approach is set to 0.2 and 0.3, which brings respectively 80% and 70% vocabulary reduction. The value of the corrective step  $\epsilon$  is set to varying from 0.005 to 0.05 with each step of 0.005 so that each running approach can achieve its best performance with a certain value of  $\epsilon$ . From Fig. 3, we can observe that when  $\gamma = 0.2$  the DF-SOT approach reaches its best performance with 0.6953 ( $\epsilon = 0.02$ ) on O-Loss, 1.5516 ( $\epsilon = 0.045$ ) on S-Loss, and 1.0578 ( $\epsilon = 0.04$ ) on H-Loss, and that when  $\gamma = 0.3$  the DF-SOT approach reaches its best performance with 0.6953 ( $\epsilon = 0.015$ ) on O-Loss, 1.5531 ( $\epsilon = 0.02$ ) on S-Loss, and 1.0547 ( $\epsilon = 0.025$ ) on H-Loss, which outperforms the best performance of the HL-SOT approach on O-Loss 0.6984 ( $\epsilon = 0.025$ ), on S-Loss 1.6188 ( $\epsilon = 0.025$ ), and on H-Loss 1.0969 ( $\epsilon = 0.05$ ). This indicates that with the proposed LFS framework, compared with the HL-SOT approach, the DF-SOT approach generally improves the classification performance.

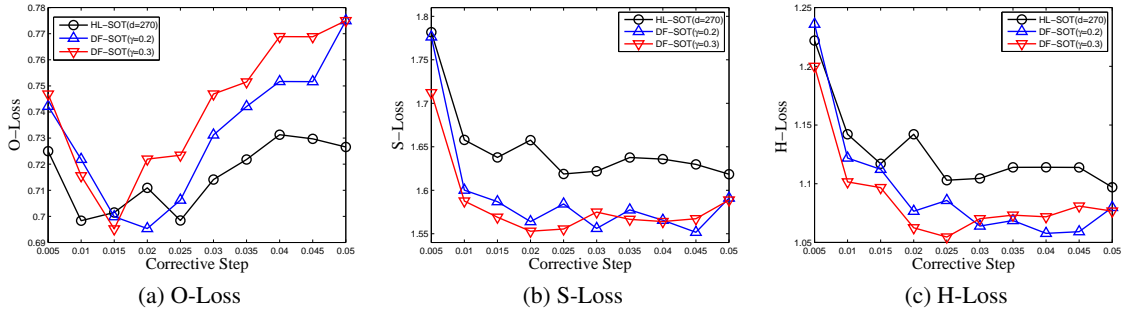


Figure 3: Classification Performance (A Smaller Loss Value Means Better Classification Performance)

### 5.3.2 Comparison on Computational Efficiency

We conduct further experiments to analyze computational efficiency gained through the proposed LFS framework. All the experiments are conducted on a normal personal computer containing an Intel Pentium D CPU (2.4 GHz, Dual Core) and 4G memory. Fig. 4 summarizes the computational time consumed by experiment runs respectively for HL-SOT ( $d = 270$ ) and DF-SOT ( $\gamma = 0.2$  and  $\gamma = 0.3$ ). From Fig. 4, we can observe that the HL-SOT approach consumes 15917695 ms to finish an experimental run, although the DF-SOT approach only takes respectively 2.29% (with  $\gamma = 0.2$ ) and 4.91% (with  $\gamma = 0.3$ ) of computational time as the existing HL-SOT approach consumes and achieves even better classification performance than the HL-SOT approach (see Fig.3). This confirms that much computational efficiency can be gained for the HL-SOT approach to be implemented in the LFS framework while better classification performance is ensured. Since the computational complexity of each node classifier of DF-SOT is the same as HL-SOT, the computational efficiency gained from the proposed LFS framework should be attributed to the dimension reduction of the index term space.

### 5.4 Comparative Study on Feature Selection Algorithms

The proposed LFS framework for the HL-SOT approach can employ various feature selection algorithms to select local features for each individual node. In this section, we conduct intensive experiments to comparatively study five classic feature selection algorithms employed within the LFS framework. The five employed feature selection algorithms are respectively document frequency

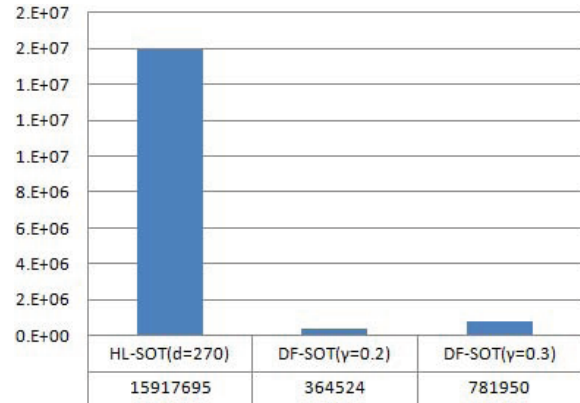


Figure 4: Time Consuming (ms)

(DF) (Manning et al., 2008) based feature selection algorithm, mutual information (MI) (Manning et al., 2008; Church and Hanks, 1990) based feature selection algorithm,  $\chi^2$ -statistic (CHI) (Manning et al., 2008) based feature selection algorithm, information gain (IG) (Mitchell, 1997) based feature selection algorithm as well as term strength (TS) (Wilbur and Sirotkin, 1992) based feature selection algorithm<sup>5</sup>.

In the experiments, the feature selection rate  $\gamma$  is set to 0.2 and 0.3 respectively. The value of the corrective step  $\epsilon$  varies from 0.005 to 0.05 with each step of 0.005. The experimental results are summarized in Fig. 5. From Fig. 5 it is observed that DF, MI, and TS feature selection algorithms achieve generally better performances than CHI and IG feature selection algorithms when they are employed in the proposed LFS framework. Specifically, the TS algorithm is generally the best among the five employed algorithms while the DF algorithm can also achieve as

<sup>5</sup>Due to the space limitation, details of the studied feature selection algorithms are not reviewed here. The mechanism of each algorithm can be read in the related references.



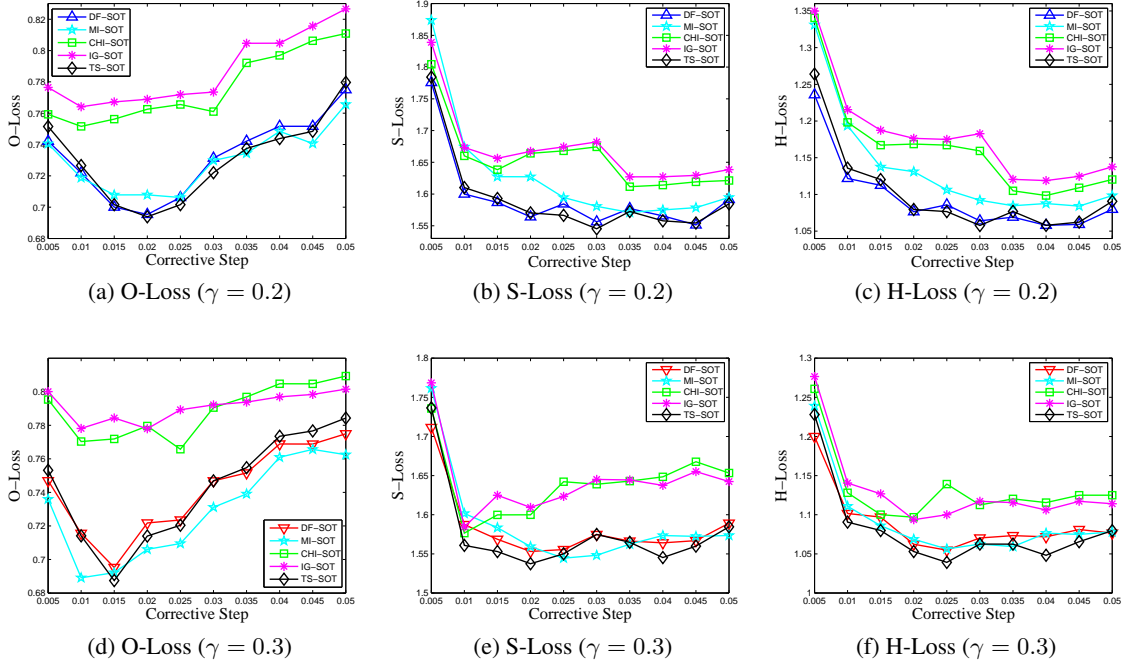


Figure 5: Comparative Performances on the Employed Feature Selection Algorithms

comparable good performance as the TS algorithm does. This is due to that both the TS and the DF algorithms favor high frequency terms and vocabularies used in customer reviews on a specific product are usually overlapping. When  $\gamma = 0.3$ , it can be also observed that the MI algorithm achieves as comparable good performance as the TS algorithm does. This is because, in customer reviews, although some vocabularies are rarely used they always occur as significant features in some specific categories. For example, “ergonomics” is a rare term but almost always appears in the class of “design and usability”. Therefore, the MI algorithm can also achieve relatively better performance through favoring rare terms that always co-occur with specific classes.

## 6 Conclusions and Future Work

In this paper, we propose a LFS framework tailored to the HL-SOT approach to sentiment analysis. In the proposed LFS framework, significant feature terms of each node can be selected to construct the locally customized index term space for the node so that the classification performance and computational efficiency of the existing HL-SOT approach are improved. The effectiveness of the proposed LFS is validated against a human-labeled data set. Further comparative study on

five employed feature selection algorithms within the proposed LFS framework indicates that the TS, DF, and MI algorithms achieve generally better performance than the CHI and IG algorithms. Among the five employed algorithms, the TS algorithm is the best to be employed by the proposed LFS framework.

Although the proposed LFS framework shows its effectiveness of improving on the HL-SOT approach, its improvement on the classification performance is not so obvious compared with its much improvement on computational efficiency. Due to the limited number of instances in the training data set, the classification performance still suffers from the problem that unobserved terms appear in testing cases. This problem is inherently raised by the bag-of-words model. A concept-based indexing scheme that can infer concepts of unobserved terms might alleviate the problem. We plan to investigate on this issue in the future work.

## Acknowledgments

The authors would like to thank the anonymous reviewers for the 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*.
- Kenneth Ward Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29.
- 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*.
- Xiaowen Ding and Bing Liu. 2007. The utility of linguistic rules in opinion mining. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and development in Information Retrieval*.
- 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*.
- 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*.
- 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*.
- 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*.
- Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of 10th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.
- 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*.
- 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*.
- 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 SIGIR Conference on Research and development in Information Retrieval*.
- Yue Lu and Chengxiang Zhai. 2008. Opinion integration through semi-supervised topic modeling. In *Proceedings of 17th International World Wide Web Conference*.
- Yue Lu, ChengXiang Zhai, and Neel Sundaresan. 2009. Rated aspect summarization of short comments. In *Proceedings of 18th International World Wide Web Conference*.
- Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schutze, 2008. *Introduction to Information Retrieval*, chapter 13, pages 271–278. Cambridge University Press.
- Tom Mitchell. 1997. *Machine Learning*. McGraw-Hill.
- 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*.
- Ivan Titov and Ryan T. McDonald. 2008. Modeling online reviews with multi-grain topic models. In *Proceedings of 17th International World Wide Web Conference*.
- 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*.
- Wei Wei and Jon Atle Gulla. 2010. Sentiment learning on product reviews via sentiment ontology tree. In *Proceedings of 48th Annual Meeting of the Association for Computational Linguistics*.
- 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*.
- J. W. Wilbur and K. Sirotkin. 1992. The automatic identification of stop words. *Journal of the American Society for Information Science*, 18:45–55.
- 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*.
- 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*.