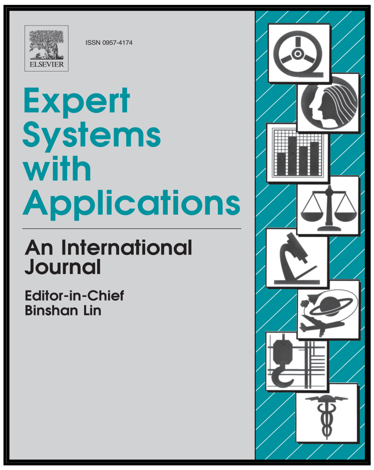
Accepted Manuscript



An improved semi-supervised dimensionality reduction using feature weighting: Application to sentiment analysis

Kyoungok Kim

PII: S0957-4174(18)30316-6
DOI: 10.1016/j.eswa.2018.05.023
Reference: ESWA 11976

To appear in: *Expert Systems With Applications*

Received date: 28 February 2017
Revised date: 18 May 2018
Accepted date: 19 May 2018

Please cite this article as: Kyoungok Kim, An improved semi-supervised dimensionality reduction using feature weighting: Application to sentiment analysis, *Expert Systems With Applications* (2018), doi: 10.1016/j.eswa.2018.05.023

Highlights

- A semi-supervised feature extraction combines with feature weighting is proposed.
- Feature weighting considers both co-occurrence of terms and label of documents.
- The polarity scores defined in SentiWordNet are reflected in the feature weights.
- Six datasets are used to validate the enhanced performance of the proposed method.

An improved semi-supervised dimensionality reduction using feature weighting: Application to sentiment analysis

Kyoungok Kim[a,*]

[a]Information Technology Management Programme, International Fusion School, Seoul National University of Science & Technology (SeoulTech), 232 Gongreungno, Nowon-gu, Seoul, 01811, Republic of Korea

Abstract

Analyzing a large number of documents for sentiment analysis entails huge complexity and cost. To alleviate this burden, dimensionality reduction has been applied to documents as a preprocessing step. Among dimensionality reduction algorithms, compared with feature selection, feature extraction can reduce information loss and achieve a higher discriminating power in sentiment classification. However, feature extraction suffers from lack of interpretability and many nonlinear extraction methods, which generally outperform linear methods, are not applicable for sentiment classification because of the characteristics that only provide corresponding low-dimensional coordinates without mapping. Therefore, this research proposes an improved semi-supervised dimensionality reduction framework that simultaneously preserves the advantages of feature extraction and addresses the drawbacks for sentiment classification. The proposed framework is mainly based on linear feature extraction providing mapping and feature weight-

*Corresponding author: Tel)+82-2-970-7286

Email addresses: kyoungok.kim@seoultech.ac.kr (Kyoungok Kim)

ing is applied before feature extraction. Feature weighting and extraction are conducted in a semi-supervised manner so that both label information and structural information of data can be considered. The superiority of both feature weighting and feature extraction was verified by conducting extensive experiments in six benchmark datasets.

Key words: Semi-supervised dimensionality reduction, Feature weighting, Feature extraction, Sentiment analysis, Natural Language Processing (NLP)

---

1. Introduction

Sentiment analysis is an emerging technique in textual data analytics that can be used for identifying subjective opinions related to specific topics in documents (Pang and Lee, 2008; Medhat et al., 2014). This technique has been widely applied in customer reviews (Maas et al., 2011; Mukherjee and Bhattacharyya, 2012), social network services (Thelwall et al., 2011; Ortigosa et al., 2014), news (Godbole and Srinivasaiah, 2007; Li et al., 2014), and blogs (Godbole and Srinivasaiah, 2007; Boldrini et al., 2012) and its application has been recently extended to other fields, including recommendation systems (Tatemura, 2000), business intelligence (Mishne and Glance, 2005), politics (Cardie et al., 2006), and marketing (Feldman, 2013). With the rapid development of IT, the Internet has become a very important source of customer opinion, attitude, feeling, and emotion. On the Internet, new documents are constantly being written and delivered to others in real time. Therefore, an expert system that collects peoples opinions that are stored in documents and detects the sentiment of these documents in real time must be built.

In order to construct an intelligent system for sentiment classification, several machine learning algorithms, such as decision tree, support vector machine (SVM), naïve Bayes (NB), k-nearest neighbor (k-NN), can be applied to a corpus (Pang et al., 2002; Chaovalit and Zhou, 2005; Medhat et al., 2014). However, similar to other text mining applications, machine learning techniques for sentiment analysis suffer from the high complexity of data analysis. Given that text data are unstructured, the relevant features, including individual words or word n-grams, are representative of the features used in text mining. By converting text data into a sequence of terms to a set of n-grams, it can be embedded in a vector space. Text documents that are represented as vectors of identifiers are usually high-dimensional, and such high dimensionality greatly limits their classification.

Dimensionality reduction techniques are usually applied to high-dimensional data to prevent their degradation and to enhance the utility of prediction models. Dimensionality reduction can also reduce the number of redundant features, facilitate the interpretation of features, reveal latent information, and minimize the cost and time for building a predictive model (Tenenbaum, 1998; Schölkopf et al., 1998; Roweis and Saul, 2000). Two approaches, namely, feature selection and feature extraction, are used to reduce the number of features under consideration.

Feature selection refers to the process of selecting a subset of relevant features. To solve classification problems, feature selection tries to find the most relevant feature set with a high discriminating power. Given the simple concept and easy interpretation of attributes, feature selection has been widely used in sentiment analysis. Feature selection methods have been classified into three types, namely, filter, wrapper, and embedded methods (Guyon and Elisseeff, 2003). Filter methods perform statistical tests to distinguish the more important and relevant features from the less relevant ones, wrapper methods find the best feature set by using trained models, and embedded methods select the features during the model training. Therefore, feature selection depends on label information to select excellent features. However, this technique cannot address the sparsity of the input data matrix, which is frequently observed in text mining applications, including sentiment analysis

Compared with feature selection, feature extraction reduces the number of variables by transforming a large number of attributes into a reduced set of features. Feature extraction attempts to obtain a meaningful low-dimensional representation of high-dimensional data. In other words, feature extraction can guarantee a lesser information loss and a higher discriminating power than feature selection (Hira and Gillies, 2015).

However, some problems have hindered the application of feature extraction in sentiment analysis. First, many feature extraction algorithms that have been classified as nonlinear methods cannot perform mapping from a high-dimensional space to a low-dimensional space, thereby prohibiting the training of a practically usable classification model and resulting in a loss of data interpretability (Hira and Gillies, 2015). An expert system that can automatically determine the sentiment of documents must obtain a low-dimensional representation of new documents during the test phase. In addition, unlike most feature selection methods, feature extraction methods are usually unsupervised, that is, label information cannot be utilized during the dimension reduction process. When applying machine learning algorithms for sentiment analysis, those documents with label information must be used to train a predictive model. This same information can also help feature extraction methods create excellent features.

Therefore, this paper proposes a semi-supervised dimensionality reduction framework that can enhance the performance of sentiment classification models by addressing the existing drawbacks of feature extraction. The main research problem focused in this research is how to build the process to reduce the dimensionality of text documents using feature extraction while preserving the advantages of feature extraction and overcoming the drawbacks of feature extraction for a sentiment classification system. Three key approaches are used to achieve this aim. First, feature weighting is applied to the term-document matrix to improve the feature extraction performance. Second, feature weighing and feature extraction are applied in semi-supervised ways, which means that both the relations among terms and documents and the partial label information are used to extract excellent features for sentiment analysis. Third, a feature extraction method that provides the mapping function from the original space to the low-dimensional space is utilized. To be more specific in the proposed framework, the semi-supervised maximum margin criterion (SS-MMC) is used in the proposed framework because research primarily aims to propose an expert system that can determine the sentiment of documents in many application domains. SS-MMC is a semi-supervised linear feature extraction algorithm that provides the mapping function from an original space to a low-dimensional space and guarantees the interpretability of the data (Song et al., 2008). Moreover, to prevent the weights of features from over-fitting during semi-supervised feature weighting, the general polarity values of terms provided by SentiWordNet (Godbole and Srinivasaiah, 2007; Baccianella et al., 2010) are utilized to determine the weights of terms. In order to reflect these polarity values in the proposed feature weighting approach, the lexicon-based feature selection process using SentiWordNet is introduced to remove those features that lack subjectivity before the feature extraction with feature weighting.

The remainder of this paper is organized as follows. Section 2 reviews the recent literature related to sentiment analysis and dimensionality reduction techniques whereas Section 3 explains the proposed semi-supervised dimensionality reduction framework. The datasets used in the experiments, experimental procedures and results are presented in Section 4. Next, I make in-depth discussion in Section 5. Finally, Section 6 describes conclusions with a brief summary and future directions.

2. Related literature

2.1. Sentiment analysis

Studies on sentiment classification have applied many machine algorithms to achieve an improved classification performance. Pang et al. (2002) provided a primitive study in sentiment classification using NB, maximum entropy classifier (ME) and SVM. In this study, 2,000 movie reviews with 1,000 positive and 1,000 negative reviews (Cornell dataset) were used and several different approaches were applied to generate feature sets. SVM gave the best results. The best accuracy was 82.9% achieved by SVM with unigrams. Boiy and Moens (2009) also compares NB, ME and SVM for sentiment analysis, but this study applied machine learning algorithms on not only the dataset used in Pang et al. (2002) but also crawled web texts in English, Dutch and French from several major blog sites (750 sentences for each language). This study achieved 87.4% accuracy for Cornell dataset. In Lane et al. (2012), different classifiers, such as NB, k-NN, SVM, and decision trees, were compared on two datasets generated by a media analysis company for sentiment classification task. Two datasets consist of 3,394, and 3,432 documents, respectively and these datasets were highly imbalanced between positive and negative classes. NB outperformed all the other classifiers for sentiment classification and the best records by NB was 76.6% and 80.0% for two datasets. Moraes et al. (2013) mainly focused on the comparison between SVM and artificial neural network (ANN) on sentiment classification task using four different datasets including Cornell dataset. This study compared SVM and ANN depending on the number of selected features and when the number of features was 1,000, SVM and ANN achieved 86% and 85.2% accuracies, respectively.

To further improve the classification performance, some studies have investigated the effectiveness of ensemble methods for sentiment classification. Tsutsumi et al. (2007) combined SVM, ME and scoring based on a score calculation process of word polarity on Cornell dataset and showed that the ensemble models were better than single models. Popular ensemble methods, such as bagging, boosting, and random subspace, were evaluated in Whitehead and Yaeger (2010), which used SVM as a base learner. In this study, five product review datasets[1] were used in evaluation and bagging with random subspace usually showed the best performance and average accuracy for five datasets by bagging with random subspace was 86.37%. Meanwhile, Wang et al. (2014) extended the candidate base learners to NB, k-NN, SVM, decision trees and ME and evaluated the contribution of the ensemble algorithms on 10 datasets. It also proved that the ensemble methods could obtain better results than the base learners and random subspace typically obtained the highest classification accuracy. In this study, random space using SVM as a base learner achieved 81.94% accuracy while single SVM ob-

[1] All datasets contains less than 1,500 reviews.

tained 79.11% accuracy.

Similar to machine learning approaches, lexicon-based methods have been widely studied. Taboada et al. (2011) proposed the method that automatically creates sentiment lexicons and developed sentiment orientation calculator. Based on sentiment orientation, documents were classified into positive and negative classes. This work used four different datasets. For Cornell dataset, 76.37% accuracy was obtained when all sentiment words were used, negation, irrealis were considered, and weighting was applied by intensifiers. In Heerschop et al. (2011), sentiment was classified based on document scores that were computed by adding the polarities of subjective words. This work proposed a discourse parsing module and the document scores were later modified according to the position of subjective words in these documents. The effect of this module was evaluated using 1,000 movie reviews extracted from Cornell dataset and the best accuracy by the proposed procedure was 72.0% accuracy. In Moreo et al. (2012), a lexicon-based, comments-oriented news sentiment analyzer and a system using a taxonomy lexicon were specifically designed for news analysis. The system was evaluated on a set of 500 news articles obtained from various news media and overall accuracy was 89%.

2.2. *Feature selection for sentiment analysis*

Feature selection aims to find the best relevant subset of features that can be used for the model construction. Feature selection methods can be divided into filtering, wrapper, and embedded methods. Filtering methods evaluate the effectiveness of features by applying various statistical tests, such as information gain, mutual information, chi-square statistics, and Z-score (Tan and Zhang, 2008; Khan et al., 2016; Ye et al., 2009; Kummer and Savoy, 2012; Kalaivani and Shunmuganathan, 2015). Although filter-based feature selection methods are simple, fast, and independent of any machine learning algorithms, they suffer from several problems, such as their lack of consideration for the relations among features and their interactions with learning algorithms. Wrapper methods determine an optimal feature set by training and evaluating different subsets of features (Chen et al., 2012; Siti Rohaidah Ahmad et al., 2015). Heuristic search methods are widely adopted to solve certain problems in generating different subsets of features, such as the extremely high complexity in terms of computation time and cost that occurs when checking all subsets of features. Unlike filter-based methods, wrapper methods consider the relationship among the model selections and the subsets of selected features. In this case, wrapper methods can lead to a higher performance improvement in the trained models compared with the filtering approach. Nevertheless, wrapper methods tend to overfit, are highly dependent on machine learning algorithms, and show high complexity in terms of computation time and cost. In embedded methods, the selection of a subset of features is built in the machine learning algorithms (Guyon and Elisseeff, 2003). These methods have a lower time complexity than wrapper methods and also consider the interactions with machine learning algorithms.

Apart from these aforementioned methods, another approach can be used in feature selection for sentiment analysis. This approach manually selects the features based on various sentiment lexicons, such as Bing Liu's opinion lexicon (Hu and Liu, 2004), multi-perspective question answering (MPQA) subjectivity lexicon (Wiebe et al., 2005), and SentiWordNet lexicon (Godbole and Srinivasaiah, 2007; Baccianella et al., 2010), all of which provide the annotated subjectivity of words. Bing Lius opinion lexicon and the MPQA lexicon only classify words into positive and negative words, while the SentiWordNet lexicon provides negative and positive scores for sentiment synsets, which are part of speech-tagged synonym sets. Using the SentiWordNet lexicon, the word polarities can be expressed as real numbers, thereby facilitating the feature selection process. Although the approaches based on these lexicons are simple to use, they do not consider the characteristics of different domains.

Table 1 summarizes the various feature selection methods used for sentiment analysis.

2.3. Feature extraction for sentiment analysis

Feature extraction differs from feature selection such that feature extraction methods generate new features that are not included in the original feature set. Therefore, feature extraction may lead to a further reduction in dimensionality without degrading performance and alleviating the sparsity problem. Nevertheless, not all feature extraction methods can be applied for feature reduction in sentiment classification because many feature extraction methods categorized as non-linear methods do not provide the mapping from the original space to the low-dimensional space, which poses a critical problem in building a sentiment classification system. Therefore, linear feature extraction methods, such as principal component analysis (PCA) and latent semantic analysis (LSA), are applied to build a predictive moded or dimensionality reduction methods examined in the machine learning literature, such as t-distributed stochastic neighbor embedding (t-SNE), have also been employed to reduce the dimensionality of text data for visualization (Huang et al., 2005; Cai et al., 2005; Mao et al., 2010; Jun et al., 2014).

Although many dimensionality reduction techniques are unsupervised, semi-

Table 1: Feature selection methods for sentiment analysis

Ref	Technique	Advantage	Disadvantage
Wang et al. (2011)	Filter using fisher's discriminant ratio	easy to implement, fast computing, independent of learning algorithms	not consider the relations among features and the interaction with learning algorithms
Manek et al. (2017)	Filter using Gini index		
Wang et al. (2007); Abbasi et al. (2008); Agarwal and Mittal (2013b); Khan et al. (2016)	Filter using mutual information or point-wise mutual information		
Agarwal and Mittal (2013b)	Filter using rough set attribute reduction		
Khan et al. (2016)	Filter using chi-square		
Agarwal and Mittal (2013a)	Filter using minimum redundancy maximum relevancy		
Abbasi et al. (2008); Zhu et al. (2010)	Wrapper using genetic algorithm	consider the relations among features and the interaction with learning algorithms and usually obtain the better models than the filter methods	high complexity in computing time and cost, easily over-fitted, dependent on the learning algorithms
Gupta et al. (2015); Basari et al. (2013); Akhtar et al. (2017)	Wrapper using particle swarm optimization		
Ahmad et al. (2017)	Wrapper using ant colony optimization		
Whitelaw et al. (2005); O'Keefe and Koprinska (2009); Mudinas et al. (2012); Sharma and Dey (2012); Hamdan et al. (2015)	Feature selection based on lexicons	simple to use, reflect linguistic knowledge	select only fixed features, cannot differentiate different domains

Table 2: Feature extraction methods for sentiment analysis

Ref	Technique	Advantage	Disadvantage
Mao et al. (2010); Jun et al. (2014); Poria et al. (2015); Zainuddin et al. (2016)	PCA	easy to implement, fast, provide the mapping, remove data sparsity	only reflect input data structure, based on the linearity assumption
Jun et al. (2014)	LSA		
Li et al. (2009)	non-negative matrix factorization		
Mao et al. (2010)	t-SNE	good reduction performance in very low dimensional space such as 2D	
Kim and Lee (2014)	semi-supervised laplacian eigenmaps	reflect partial label information and non-linear data structure	not provide the mapping

supervised feature extraction algorithms have been investigated in order to consider certain types of prior knowledge. One of these algorithms adopts predefined low-dimensional representations of some data points (Yang et al., 2006). However, prior knowledge of low-dimensional representations is rarely considered in real applications. Another semi-supervised dimensionality reduction algorithm directly utilizes label information to construct a weighted graph (Song et al., 2008; Kim and Lee, 2014). However, these algorithms do not distinguish the importance of terms in feature extraction. In Nie et al. (2013), label information was introduced as a loss function that minimizes losses when predicting targets as well as structural embedding errors that require iterative optimization procedures.

Table 2 summarizes the various feature extraction methods used for sentiment analysis.

3. The proposed semi-supervised dimensionality reduction framework

The overall experimental procedures including the proposed semi-supervised dimensionality reduction framework are illustrated in Figure 1. Here, the major steps of the proposed framework are briefly explained before the detailed explanation. The first step of the proposed framework is to create a term-frequency matrix with respect to the given corpus. After preprocessing, term frequencies of terms for every document are calculated. At this step, SentiWordNet (Godbole and Srinivasaiah, 2007; Baccianella et al., 2010) is used to exclude the relatively irrelevant features for sentiment analysis by only retaining those terms that are included in SentiWordNet. SentiWordNet distinguishes POS of terms, so POS tagging should be performed in preprocessing. In the second step, feature weighting is conducted before feature extraction. To do this, the feature weights are computed using certain statistical methods, such as point-wise mutual information (PMI) and Fishers discriminant ratio (FDR) and the term frequency matrix is multiplied by the calculated feature weights. While previous research has usually utilized such methods to select features (Asghar et al., 2014), the proposed framework does not remove additional features using the feature weights because some useful information may be discarded through a feature selection process. In the final step, feature extraction is applied to the weighted term frequency matrix by weights calculated in the second step. In the second and fourth steps, the weight calculation and the feature extraction are conducted in a semi-supervised manner.

The details are described in following sections and preprocessing and evaluation procedures are explained in Section 4.

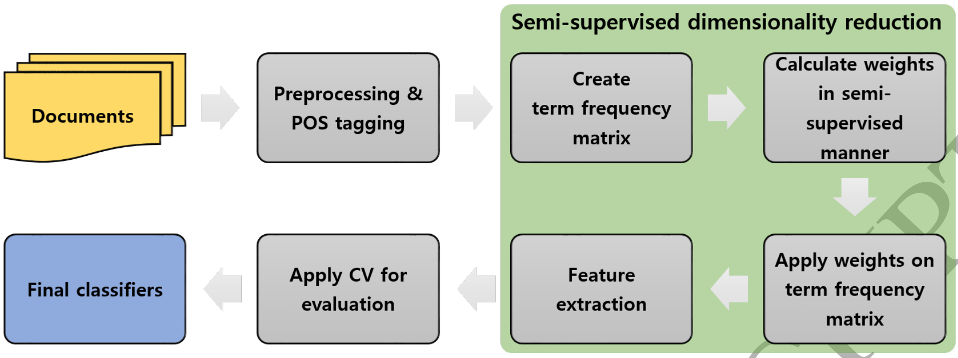


Figure 1: The experimental procedures including the proposed semi-supervised dimensionality reduction framework

3.1. Semi-supervised feature weighting

The key concept of the proposed feature weighting method is to utilize label information to compute better weights of features as well as the graph Laplacian of the graph that depicts the relations among terms. In particular, the weights of words are considered when calculating the similarity between documents. In this study, the weights of words are assumed to be proportional to the absolute word polarities because those words with a strong sentiment might be more effective in determining the overall sentiment of documents.

When the polarities of some portions of words are known in advance, the unknown polarities of the remaining words can be calculated in a semi-supervised manner. To this end, a graph that reflects the relations between words is constructed, and then the unknown polarities are inferred by using the graph Laplacian. The objective function that needs to be minimized to obtain the unknown target is defined as follows (Belkin et al., 2004; Zha et al., 2009).

$$J = \sum_{i=1}^{d} \sum_{j=1}^{d} e_{ij}(p_i - p_j)^2 \tag{1}$$

where d is the number of words that are observed in the set of documents and e_{ij} is the weight of the edge that connects the ith word and the jth word. Here, p_i denotes the polarity of the ith word, which is the target of the constructed graph. The main principle of the objective function is that two strongly connected words have similar polarities.

The matrix $\mathbf{A}$, called an adjacency matrix, is a $d \times d$ matrix which elements indicate the weights of edges. Then, the objective function can be rewritten as

$$J = \sum_{i=1}^{d} \sum_{j=1}^{d} e_{ij}(p_i - p_j)^2 = \mathbf{p}^T \mathbf{L} \mathbf{p} \tag{2}$$

where the graph Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{A}$ and $\mathbf{D}$ is a diagonal matrix whose element $D_{ii} = \sum_j e_{ij}$. $\mathbf{p}$ is the vector consisting of the polarities of d words (Belkin and Niyogi, 2003). In particular, the normalized graph Laplacian is defined as $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}}$ instead of the ordinary graph Laplacian.

In the case that the polarities of some words are fixed, $\mathbf{p}$ is divided into two parts, $\mathbf{p} = [\ \mathbf{p}^l \quad \mathbf{p}^u\ ]^T$ where l represents labeled words and u represents unlabeled words. With this division, $\mathbf{L}$ can be decomposed into $\mathbf{L} = \begin{bmatrix} \mathbf{L}_{11} & \mathbf{L}_{12} \\ \mathbf{L}_{21} & \mathbf{L}_{22} \end{bmatrix}$ where $d_l \times d_l$ matrix, $\mathbf{L}_{11}$, corresponds to d_l labeled words and $d_u \times d_u$ matrix, $\mathbf{L}_{22}$, corresponds to d_u unlabeled words. Therefore, the objective function is reformulated as follows.

$$J = \mathbf{p}^{l^T} \mathbf{L}_{11} \mathbf{p}^l + 2\mathbf{p}^{u^T} \mathbf{L}_{21} \mathbf{p}^l + \mathbf{p}^{u^T} \mathbf{L}_{22} \mathbf{p}^u \tag{3}$$

Because $\mathbf{p}^{l^T} \mathbf{L}_{11} \mathbf{p}^l$ is fixed, the objective function is simplified further.

$$\text{argmin}_{\mathbf{p}^u} J = \text{argmin}_{\mathbf{p}^u} 2\mathbf{p}^{u^T} \mathbf{L}_{21} \mathbf{p}^l + \mathbf{p}^{u^T} \mathbf{L}_{22} \mathbf{p}^u \tag{4}$$

The solution of the equation (4), $\mathbf{p}^u$, can be easily obtained by solving a linear system, $\mathbf{L}_{22} \mathbf{p}^u = -\mathbf{L}_{21} \mathbf{p}^l$.

To calculate the word polarities without referring to the polarity scores through the aforementioned procedure, two things are required.

1. A similarity measure between words, which is necessary for calculating the graph Laplacian, $\mathbf{L}$.
2. Words with known polarities, which are the elements of $\mathbf{p}^l$.

3.1.1. Deciding the weights of edges

The weights of edges in a graph, which nodes correspond to subjective words, must reflect the similarity between words. As one of the most widely used similarity measures for words, co-occurrence is usually calculated by counting how frequently two words are used within the same linguistic unit, such as an n-gram, phrase, or sentence. For example, when the linguistic unit is set to a sentence, the co-occurrence of words i and j is defined as the number of sentences that use both of these words throughout the entire corpus. To calculate the co-occurrence of these words, all sentences that use both words i and j are checked and counted regardless of the documents in the corpus. A greater co-occurrence indicates a greater similarity between two words (Momtazi et al., 2010).

In this paper, a paragraph is used for the window size. A document is too long to calculate co-occurrence because sentiment over a document is sometimes irregular, and a sentence is too short to calculate the co-occurrence of subjective words. To calculate the weights of the edges for the given corpus, the paragraph-wise co-occurrence was obtained for every term used to build a sentiment classifier.

3.1.2. Calculation of polarities of words using labeled documents

In this study, two measures are introduced to determine the polarities of sentiment-based words. The first measure is point-wise mutual information (PMI) (Bouma, 2009; Varela et al., 2013) and the second measure is Fisher's discriminant ratio (FDR) (Wang et al., 2011). These two measures are two of the most widely used criteria for feature selection. PMI measures the association between two events and is defined as follows:

$$\text{PMI}(x, y) = \log_2 \frac{p(x, y)}{p(x)p(y)} = \log_2 \frac{p(x|y)}{p(x)} \tag{5}$$

For the proposed method, x is replaced by t representing words, and y is replaced by s representing the sentiment of the words. A binary sentiment annotation, namely, positive (+1) and negative (1), is used in all experiments. Based on equation (5), the PMI of subjective words in labeled documents can be calculated as follows:

$$\text{PMI}(t, s) = \log_2 \frac{A \times n_l}{(A + B) \times (A + C)} \tag{6}$$

where n_l is the total number of labeled documents, A is the number of documents labeled as s containing word t, B is the number of documents not labeled as s containing t, and C is the number of documents labeled as s without t. Because s can be +1 or -1, there are two values of PMI related with word t. The polarity of word t, $p_{t,label}$ (l means that polarity of a term is calculated from labeled documents), is calculated by

$$p_{t,label} = \text{PMI}(t, +1) - \text{PMI}(t, -1) \tag{7}$$

The main drawback of PMI is that this method does not account for the frequency of a word that is generally used throughout a document and may miscalculate the polarities of rarely used words. Therefore, the $\log$ of document frequency considering only the labeled documents, $df_{t,label}$ was multiplied by p_t to reflect uncertainty as follows:

$$p_{t,label} = \log(df_{t,label})\{\text{PMI}(t, +1) - \text{PMI}(t, -1)\} \tag{8}$$

For example, if a term, t is appeared in 30 positive documents and 65 negative documents and the numbers of positively and negatively labeled documents are both 100, PMI$(t, +1) = \log_2(30 \times 200)/\{(30 + 65)(30 + 70)\}$ and PMI$(t, -1) = \log_2(65 \times 200)/\{(65 + 30)(65 + 35)\}$. From PMI$(t, +1)$ and PMI$(t, -1)$, p_t can be calculated by $\log(30 + 65)\{\text{PMI}(t, +1) - \text{PMI}(t, -1)\}$ based on equation (8).

Another measure used in this study, FDR, is defined as follows:

$$\text{FDR}(\mathbf{x}) = \frac{(m_1 - m_2)^2}{S_1^2 + S_2^2} \tag{9}$$

where $\mathbf{x}$ is a vector of a feature and m_i and S_i^2 $(i = 1, 2)$ are the mean and the with-in class scatter of class i. FDR considers with-in class scatters to rank the separability of features because large with-in class scatters can cause a large mean difference by coincidence. A larger FDR implies that the feature provides better separation of the two classes.

FDR can be reformulated to calculate the polarities of words by considering term frequencies in each document. Let $f_{t,i}$ be a term frequency t in the document i, and then m_s and S_s^2 (s represents two different sentiment labels, +1 and -1) can be given as

$$m_s = \frac{\sum_{i \in C_s} f_{t,i}}{n_s} \tag{10}$$

$$S_s^2 = \sum_{i \in C_s} (f_{t,i} - m_s)^2 \tag{11}$$

where C_s is the set of documents with label s and n_s is the size of C_s.

For example, if the average term frequencies in the positive and negative classes are 2 and 10 for a term, t and the with-in class scatters of two classes are 3 and 5, FDR$(t) = (2 - 10)^2/(3 + 5)$.

In both cases using PMI and FDR, the calculated polarities of words were normalized using min-max normalization to confine the polarities in the range of

[-1, 1] with the equation as follows:

$$p_{t,l}^{'} = 2 \cdot \frac{p_t - \min_j p_j}{\max_j p_j - \min_j p_j} - 1, \quad j \in \{1, 2, ..., d\} \tag{12}$$

where d is the total number of terms in the initial term frequency matrix. Absolute values of the final polarities were used as weights of words. Only the labeled documents were utilized to obtain p_t; therefore, the polarities of some words that only appeared in unlabeled documents were estimated by solving equation (4).

3.1.3. Calculation of polarities of words using SentiWordNet

To obtain better feature weights, the word polarities defined in SentiWordNet are set as the baseline of the word weights. When a specific set of documents is utilized to decide the weights of terms, over-fitting may occur and subsequently lead to poor performance on a test dataset. Given that SentiWordNet provides positive and negative scores for synsets in general contexts, the polarity information from SentiWordNet prevents the weights of terms from over-fitting.

The polarities of synsets are calculated as follows:

$$p_{t,swn} = ss_{pos} - ss_{neg} \tag{13}$$

where ss represent the score of a synset defined in SentiWordNet. However, a term with a specific part-of-speech (POS) can appear multiple times with a different sense usage and be scored in SentiWordNet. In such cases, the polarities of synsets described in SentiWordNet are calculated by PMI and FDR, as similar to the polarities of terms based on labeled documents.

For example, a synset 'average_a' are appeared three times in SentiWordNet and positive and negative scores are described in Table 3.

To calculate PMI, A, B, C and n_l defined in equation (6) should be defined. For PMI, labels of synsets are obtained by comparing positive and negative scores.

	1	2	3
ss_{pos}	0.375	0.000	0.250
ss_{neg}	0.125	0.250	0.625
label	+1	-1	+1

Table 3: Scores of synset 'average_a'

If a positive score is greater than a negative score, the label is defined as +1. Otherwise, the label is set as -1. Then, let A is the sum of positive scores of synsets with +1, B is the sum of positive scores of sysnsets with +1, C is the sum of negative scores of synsets with +1, and n_l is set as the number of synsets for PMI$(t, +1)$. A, B, and C for PMI$(t, -1)$ are oppositely defined to A, B, and C for PMI$(t, +1)$. For PMI(average_a, $+1$), $A = 0.625$, $B = 0.000$, $C = 0.750$, and $n_l = 3$. Unlike $p_{t,label}$, multiplication of $\log$ of document frequency is omitted in calculations of $p_{t,swn}$ by PMI.

For FDR, calculation is simpler than PMI. **x** is set as $(0.375, 0, 0.25, 0.125, 0.25, 0.625)$ and labels of each element are defined as $(+1, +1, +1, -1, -1, -1)$. From these two vectors, m_s and S_s ($s \in +1, -1$) are easily calculated. For example, $m_{+1} = (0.375 + 0 + 0.25)/3$.

After calculation of $p_{t,swn}$, $p_{t,swn}$ is normalized to $p^{'}_{t,swn}$ using min-max normalization.

3.1.4. Feature weight calculation

The proposed feature weighting method calculates the final weights of features by hybridizing the statistical and lexicon-based approaches. As discussed in Section 3.1.2 and Section 3.1.3, the two different word polarities of a term are computed by using PMI and FDR, and by SentiWordNet, respectively.

From these polarities, feature weights are obtained by hybridizing of $p_{t,label}$

and $p_{t,swn}$. The parameter β is used to control the influence of the polarity defined by SentiWordNet on the final weights. To avoid negative weights, absolute values of $p_{t,label}$ and $p_{t,swn}$ are used in calculation and let $w_{t,label} = |p'_{t,label}|$ and $w_{t,swn} = |p'_{t,swn}|$. A weight of a term t, w_t is calculated as follows:

$$w_t = (1 - \beta) \cdot w_{t,label} + \beta \cdot w_{t,swn} \tag{14}$$

These weights are multiplied by the corresponding columns of the term frequency matrix calculated from the preprocessed corpus.

3.2. Semi-supervised feature extraction by using feature weighting

Feature weighting is merely to emphasize more important features and the weighted term frequency matrix is still high-dimensional. In order to further reduce dimensionality, semi-supervised feature extraction is applied on the weighted term frequency matrix after calculating the feature weights in the proposed framework.

Unlike commonly used unsupervised feature extraction algorithms, semi-supervised feature extraction algorithms utilize partially labeled data that are common in real-world problems (Song et al., 2008; Kim and Lee, 2014; Zhao et al., 2014). Introducing the semi-supervised concept can improve the performance of existing dimensionality reduction methods because this concept generally tries to separate data points with different labels as much as possible.

In the proposed framework, SS-MMC was utilized to reduce dimensionality. SS-MMC considers the structural information of the data by using the graph Laplacian of the graph that depicts relations between documents, as well as the label information of the label information of the training data, similar to MMC, to extract better features when partially annotated documents are available (Song et al., 2008). Given that SS-MMC provides a loading matrix similar to PCA and MMC, new documents can be easily mapped into a low-dimensional space after the learning process. In this case, the new data points that are obtained after dimensionality reduction can be used for sentiment classification.

MMC, the base of SS-MMC, calculates projection vectors by solving the following optimization problem (Li et al., 2006).

$$\mathbf{W} = \text{argmin}_{\mathbf{W}^T\mathbf{W}=\mathbf{I}} \text{tr}(\mathbf{W}^T(\mathbf{S}_b - \lambda \mathbf{S}_w)\mathbf{W}) \quad (15)$$

where $\mathbf{W}$ consists of m column vectors with the length d (the original dimensionality of data) and these column vectors are projection vectors, which means that m is the dimensionality of document vectors to be transformed. $\mathbf{S}_b$ is the between-class scatter matrix and $\mathbf{S}_w$ is the within-class scatter matrix. $\mathbf{S}_b$ and $\mathbf{S}_w$ require label information to compute, so MMC is classified as a supervised algorithm. In addition, SS-MMC considers structure information when partially annotated data is available, and the objective function is defined as follows:

$$\mathbf{W} = \text{argmin}_{\mathbf{W}^T\mathbf{W}=\mathbf{I}} \text{tr}(\mathbf{W}^T(\mathbf{S}_b - \lambda_1 \mathbf{S}_w - \lambda_2 \mathbf{X}^T\mathbf{L}\mathbf{X})\mathbf{W}) \quad (16)$$

where $\mathbf{L}$ represents graph Laplacian obtained from the given dataset and $\mathbf{X}$ is an $n \times d$ weighted term frequency matrix including both labeled and unlabeled documents calculated in Section 3.1.4. Here, n is the total number of documents including both labeled and unlabeled documents, and d is the original dimensionality of data. To get the low-dimensional embeddings by SS-MMC, a graph, G is required to calculate $\mathbf{L}$. The common way to construct a graph is to use the k-nearest neighbor graph. In this study, a symmetric k-nearest neighbor graph is used and an adjacency matrix of this graph, A is defined as follows:

$$a_{ij} = \begin{cases} \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right), & \text{if } \mathbf{x}_i \in N_k(\mathbf{x}_j) \text{ or } \mathbf{x}_j \in N_k(\mathbf{x}_i) \\ 0, & \text{otherwise} \end{cases} \tag{17}$$

where $\mathbf{x}$ is vector representation of a document (a row of the weighted term frequency matrix $\mathbf{X}$) and $N_k(\mathbf{x})$ is the k-nearest neighbor set of $\mathbf{x}$. This graph G reflects the similarity between data, so the objective function of SS-MMC can consider the structural information of data.

SS-MMC was used in this study, but any other semi-supervised feature extraction algorithms that provide a mapping function to a low-dimensional space can be incorporated in the proposed framework to enhance its performance.

4. Experiments

The whole experimental procedures are illustrated in Figure 1 motioned in Section 3. While Section 3 explains the proposed framework, this section, first, describes preprocessing and evaluation procedures before presenting the experimental results.

4.1. Data preparation

Some benchmark datasets that are frequently used in the field of sentiment analysis were employed in the experiments to evaluate the proposed algorithms. A total of six datasets that are available online and have been extensively studied were eventually selected. All of these datasets were annotated as binary targets, either positive or negative.

The first dataset was the Large Movie Review dataset, which included 50,000 movie reviews, among which 25,000 reviews were positively annotated and the rest were classified as negative[2]. The second dataset, the Cornell Movie Review dataset, included 2,000 movie reviews, among which 1,000 were positive and 1,000 were negative[3]. The other four datasets were obtained from the multi-domain sentiment dataset, which included reviews of products from different categories in Amazon.com[4]. Four categories, namely, Book, DVD, Electronics, and Kitchen, were selected for the experiments, with each category including 2,000 reviews. Similar to the Large Movie Review and Cornell Movie Review datasets, these four datasets had a 50/50 sentiment ratio.

These six datasets were preprocessed before applying the proposed algorithm to enhance their suitability for the sentiment analysis and to improve the performance of the classifiers. AutoMap[5], a text mining tool developed by CASOS at Carnegie Mellon, and NLTK 3.0[6], a natural language toolkit for Python, were used for the preprocessing. The following procedures were applied on the data.

- Spelling correction: Common spelling mistakes were corrected by using AutoMap
- British English to American English: Some words that were spelled in British English were converted to American English by using AutoMap to improve performance
- Expand abbreviations and contractions: Common abbreviations and contractions were expanded by using AutoMap

[2]http://ai.stanford.edu/ amaas/data/sentiment/
[3]http://www.cs.cornell.edu/people/pabo/movie-review-data/
[4]https://www.cs.jhu.edu/ mdredze/datasets/sentiment/index2.html
[5]http://www.casos.cs.cmu.edu/projects/automap/
[6]http://www.nltk.org/

- Remove stop words, URLs, symbols, and punctuations: The punctuations, symbols (e.g., currency symbols), and stop words (a set of commonly used words in any language) that are usually irrelevant in sentiment analysis must be removed to improve performance. The list of stop words published by NLTK was used in the experiments

- Convert lowercase: All characters were converted to lowercase

- Lemmatization: Lemmatization refers to the process of grouping together the different inflected forms of a word in order for them to be analyzed as the same term. For example, in English, the verb 'to walk' may appear as 'walk', 'walks', 'walked', or 'walking'. Through lemmatization, all of these words are converted to their base form, 'walk', which is considered the lemma in this work. Lemmatization was performed by using NLTK, which lemmatizer is based on the build-in morphy function in WordNet

- POS tagging: POS tagging was performed by using NLTK. Among all POS taggers provided by NLTK, the Universal POS tagger was selected because it uses the same tagset as SentiWordNet

To obtain highly effective features for the sentiment analysis, only those terms with a POS that also existed in SentiWordNet were retained after the preprocessing. Using lexical words as candidate input features prevents the number of features from exceeding a certain upper bound (i.e., the total number of words in SentiWordNet). Given that SentiWordNet has four POS tags, namely, adjective, adverb, noun, and verb, all the words with other POS tags, including pronoun and conjunction, were removed from the final feature set. After the initial feature selection, six sets of documents were converted to term frequency matrices where each column corresponded to a term with a certain POS.

As described in Section 3, the raw datasets are transformed to term frequency matrices after the preprocessing steps and only terms defined in SentiWordNet are kept. Then, the semi-supervised feature extraction following feature weighing is conducted. In this paper, for SS-MMC, a randomized singular value decomposition (SVD) is introduced instead of the original SVD algorithm for the Large Movie Review dataset, to reduce computation time and resources (Martinsson et al., 2011). Then, the performance of the classifiers trained in the original space, and the low-dimensional spaces with different dimensions, are compared with each other. Support vector machine (SVM) using a radial basis function (rbf) kernel is used to train sentiment classification models because many previous studies had proved that SVM worked best for sentiment analysis in many different domains.

In the experimental procedures, some steps require parameter selection. First of all, for SS-MMC, k and σ to obtain a graph G from data as defined in equation (17) and λ_1 and λ_2 used in the objective function of SS-MMC should be determined in advance. Through preliminary experiments, it was proved that the performance of dimensionality reduction was not very sensitive to the setting of λ_1. Therefore, λ_1 is set to 1, which was found to be a fairly good choice in the preliminary experiments. For λ_2, it is important to have a balance between $\mathbf{S}_b - \lambda_1 \mathbf{S}_w$ and $\mathbf{X}^T\mathbf{L}\mathbf{X}$. In addition, Secondly, parameters C to control the balance between the margin and misclassification errors, and γ, the parameter of an rbf kernel, should be determined for SVM. In this study, σ, λ_2, C, and γ were selected in the range of $[10^{-4}, 10^4]$ with log scale and k was selected among $[5, 10, 20, 50]$.

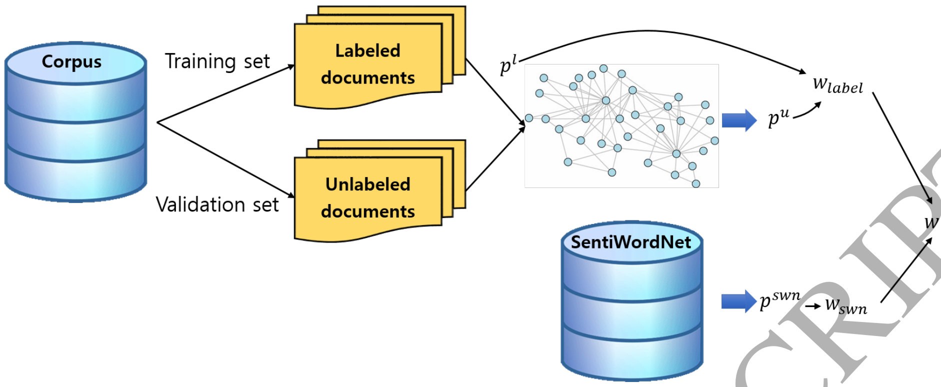


Figure 2: Feature weight calculation

All pairs of $(k, \sigma, \lambda_2, C, \gamma)$ were evaluated through 10-fold cross-validation. In this paper, cross-validation was repeated 10 times and the results were averaged out for comparison. The optimal setting of parameters is only reported in the following section.

One thing I want to mention again is that the main purpose of the proposed algorithm is to calculate the weights of features in a semi-supervised manner. Therefore, a training set is considered as labeled documents and validation set is considered as unlabeled documents during cross-validation, which is depicted in Figure 2.

To evaluate the effectiveness of the proposed framework, four kinds of evaluation measures, accuracy, precision, recall and the F1-measure were adopted in this paper.

4.2. Results

Tables 4~9 and Tables 10~15 summarize the average classification results of the 10-fold cross-validation for each dataset when the weights of features are calculated by using PMI and FDR, respectively. The standard deviations of the experimental results are enclosed in brackets.

To validate the effectiveness of the weights that were computed in a semi-supervised manner, the classification results results that used the weights computed based on SentiWordNet (denoted as w_{swn}) and the classification results results that used the weights obtained by Equation (14) (denoted as the best w) were compared in the same table. In the experiments, β was set in the range of [0,1] with 0.2 intervals. Similar to the other parameters for SVM, the best β was selected via cross-validation because the final weights depend on β to control the impact of w_{swn}. In these tables, Acc., Prec., Rec., and F1 represent accuracy, precision, recall, and F1-measure, respectively, and Orig. means the results obtained from the original term frequency matrices, without dimensionality reduction by SS-MMC.

Overall, the best weights obtained by hybridizing w_{swn} and w_{label} outperformed w_{swn} for all datasets regardless of whether these weights were processed by PMI or FDR. Only slight differences were observed in the performance rankings measured by the four metrics. In most instances, the case with the best accuracy also achieved the best precision, recall, and F1-measure.

The evaluation results of the trained classifiers without weighting are presented in Table 16. As can be seen in this table, applying the weights before learning SVM improved the classification performance for most of the six datasets. However, such improvement was marginal when w_{swn} was applied for the feature extraction because w_{swn} only considers the typical subjectivity of terms regardless of the context in which these terms are used. Moreover, the effectiveness of feature weighting usually increases as the reduced dimensionality decreases, thereby suggesting that feature weighting reduces the information loss during feature extraction.

The best weights were mostly obtained when β was equal to 0.8 regardless of whether PMI or FDR was applied for feature weighting. Therefore, those weights that consider domain-specific information can significantly improve the performance of classifiers for the sentiment analysis. However, in most cases, w_{best} was not observed at $\beta = 1$, thereby suggesting that the polarity scores of terms in SentiWordNet were not impractical in the feature weighting. This observation might be because SentiWordNet provides the general polarities of subjective terms and reduces the overfitted weights in specific domains.

In terms of dimensionality, the best models were trained when the reduced dimensionality was 500 although the best performance in some cases was reported at a dimensionality of 300. Even though 300 or 500 is much smaller than the original dimensionality of the six datasets, the classifiers trained in low-dimensional spaces showed better classification results, which could be attributed to the inherent characteristics and sparseness of term frequency matrices.

As another advantage, dimensionality reduction shortens the computation time when training and testing a classification model. In the training and testing phases, SVM requires the computation of the kernel matrix, and the computation time depends on the number of data points and attributes in the training data. Given that text data usually have high original dimensionality, building an SVM classifier in the original space is a burdensome task. Dimensionality reduction also requires additional computation time. However, the randomized SVD algorithm reduces the computation cost of SS-MMC and subsequently shortens the computation time.

Table 4: Classification results using $w_{pmi,swn}$ and the best w_{pmi} for Large Movie Review

Dim	w_{swn}				best w				
	Acc.	Prec.	Rec.	F1	β	Acc.	Prec.	Rec.	F1
50	0.7528 (0.0119)	0.7388 (0.0321)	0.7820 (0.0269)	0.7598 (0.0210)	0.8	0.7944 (0.0165)	0.7698 (0.0369)	0.8436 (0.0194)	0.8034 (0.0238)
100	0.7644 (0.0144)	0.7323 (0.0383)	0.8336 (0.0260)	0.7796 (0.0197)	0.8	0.8156 (0.0165)	0.7886 (0.0426)	0.8624 (0.0294)	0.8238 (0.0237)
200	0.7962 (0.0200)	0.7762 (0.0327)	0.8352 (0.0228)	0.8039 (0.0277)	1.0	0.8326 (0.0163)	0.8158 (0.0398)	0.8592 (0.0279)	0.8369 (0.0237)
300	0.8022 (0.0200)	0.7879 (0.0327)	0.8392 (0.0228)	0.8171 (0.0277)	1.0	0.8382 (0.0163)	0.8253 (0.0398)	0.8580 (0.0279)	0.8413 (0.0237)
500	0.8188 (0.0183)	0.7949 (0.0303)	0.8508 (0.0296)	0.8239 (0.0250)	1.0	0.8426 (0.0172)	0.8265 (0.0350)	0.8670 (0.0264)	0.8464 (0.0217)
Orig.	0.7992 (0.0170)	0.7672 (0.0352)	0.8664 (0.0278)	0.8118 (0.0202)	1.0	0.8398 (0.0179)	0.8112 (0.0334)	0.8644 (0.0222)	0.8393 (0.0237)

Table 5: Classification results using $w_{pmi,swn}$ and the best w_{pmi} for Cornell

Dim	w_{swn}				best w				
	Acc.	Prec.	Rec.	F1	β	Acc.	Prec.	Rec.	F1
50	0.7390 (0.0281)	0.7040 (0.0374)	0.7585 (0.0271)	0.7291 (0.0215)	0.8	0.7840 (0.0159)	0.7620 (0.0333)	0.7990 (0.0266)	0.7781 (0.0186)
100	0.7685 (0.0181)	0.7570 (0.03725)	0.7757 (0.0211)	0.7652 (0.0234)	0.8	0.8065 (0.0169)	0.8070 (0.0365)	0.8074 (0.0255)	0.8058 (0.0205)
200	0.7770 (0.0240)	0.7560 (0.0380)	0.7897 (0.0211)	0.7717 (0.0217)	0.8	0.8195 (0.0169)	0.8010 (0.0338)	0.8336 (0.0225)	0.8149 (0.0221)
300	0.7875 (0.0253)	0.7590 (0.0331)	0.8052 (0.0223)	0.7806 (0.0216)	0.8	0.8375 (0.0222)	0.8310 (0.0379)	0.8440 (0.0233)	0.8353 (0.0183)
500	0.7895 (0.0273)	0.7390 (0.0385)	0.8046 (0.0284)	0.7843 (0.0211)	0.8	0.8445 (0.0233)	0.8250 (0.0360)	0.8425 (0.0260)	0.8319 (0.0233)
Orig.	0.8025 (0.0187)	0.7930 (0.0350)	0.8090 (0.0200)	0.8001 (0.0229)	0.8	0.8340 (0.0234)	0.8270 (0.0354)	0.8403 (0.0210)	0.8322 (0.0270)

Table 6: Classification results using $w_{pmi,swn}$ and the best w_{pmi} for Book

Dim	w_{swn}				best w				
	Acc.	Prec.	Rec.	F1	β	Acc.	Prec.	Rec.	F1
50	0.6650 (0.0331)	0.6160 (0.0352)	0.6861 (0.384)	0.6472 (0.0260)	0.8	0.7430 (0.0229)	0.7350 (0.0338)	0.7495 (0.0272)	0.7423 (0.0272)
100	0.6650 (0.0182)	0.5840 (0.0353)	0.6974 (0.0273)	0.6333 (0.0275)	0.8	0.7640 (0.232)	0.7440 (0.0317)	0.7736 (0.0245)	0.7510 (0.0202)
200	0.6970 (0.0257)	0.6750 (0.0328)	0.7074 (0.0287)	0.6891 (0.0248)	1.0	0.7715 (0.0265)	0.7620 (0.0361)	0.7771 (0.0204)	0.7681 (0.0213)
300	0.6990 (0.0327)	0.6910 (0.0347)	0.7038 (0.0367)	0.6963 (0.0320)	1.0	0.7740 (0.0162)	0.7550 (0.0347)	0.7796 (0.0271)	0.7709 (0.0230)
500	0.7160 (0.0347)	0.7040 (0.0312)	0.7232 (0.0373)	0.7119 (0.0326)	1.0	0.7785 (0.0238)	0.7780 (0.0285)	0.7939 (0.0276)	0.7787 (0.0262)
Orig.	0.7295 (0.0322)	0.7300 (0.0374)	0.7306 (0.0276)	0.7294 (0.0231)	0.8	0.7360 (0.0254)	0.7210 (0.0270)	0.7449 (0.0223)	0.7316 (0.0280)

Table 7: Classification results using $w_{pmi,swn}$ and the best w_{pmi} for DVD

Dim	w_{swn}				best w				
	Acc.	Prec.	Rec.	F1	β	Acc.	Prec.	Rec.	F1
50	0.7150 (0.0289)	0.6490 (0.0301)	0.7517 (0.0321)	0.6949 (0.0272)	0.6	0.7405 (0.0155)	0.7280 (0.0261)	0.7473 (0.0193)	0.7369 (0.0191)
100	0.7260 (0.0242)	0.7180 (0.0337)	0.7409 (0.0273)	0.7282 (0.0234)	0.8	0.7545 (0.0174)	0.7570 (0.0283)	0.7541 (0.0188)	0.7548 (0.0207)
200	0.7440 (0.0222)	0.6940 (0.0295)	0.7740 (0.0214)	0.7304 (0.0218)	0.6	0.7760 (0.0201)	0.8010 (0.0247)	0.7633 (0.0159)	0.7814 (0.0201)
300	0.7480 (0.0265)	0.7450 (0.0309)	0.7515 (0.0249)	0.7473 (0.0240)	0.8	0.7775 (0.0192)	0.8100 (0.0214)	0.7612 (0.0199)	0.7843 (0.0160)
500	0.7675 (0.0285)	0.7730 (0.0259)	0.7662 (0.0271)	0.7688 (0.0267)	0.8	0.7835 (0.0144)	0.8050 (0.0227)	0.7589 (0.0237)	0.7805 (0.0216)
Orig.	0.7660 (0.0323)	0.790 (0.0328)	0.7612 (0.0307)	0.7695 (0.0277)	0.8	0.7765 (0.0235)	0.8130 (0.0283)	0.7584 (0.0260)	0.7842 (0.0235)

Table 8: Classification results using $w_{pmi,swn}$ and the best w_{pmi} for Electronics

Dim	w_{swn}				best w				
	Acc.	Prec.	Rec.	F1	β	Acc.	Prec.	Rec.	F1
50	0.7180 (0.0204)	0.6540 (0.0319)	0.7512 (0.0268)	0.6981 (0.0268)	0.8	0.7755 (0.0165)	0.7890 (0.0215)	0.7687 (0.0155)	0.7784 (0.0213)
100	0.7460 (0.0348)	0.7150 (0.0328)	0.7619 (0.0214)	0.7372 (0.0207)	0.8	0.7805 (0.0177)	0.7990 (0.0218)	0.7710 (0.0232)	0.7841 (0.0267)
200	0.7480 (0.0202)	0.7280 (0.0265)	0.7590 (0.0239)	0.7426 (0.0229)	1.0	0.7980 (0.0259)	0.8050 (0.0281)	0.7845 (0.0163)	0.7947 (0.0223)
300	0.7600 (0.0230)	0.7610 (0.0275)	0.7611 (0.0202)	0.7603 (0.0207)	1.0	0.7970 (0.0210)	0.8140 (0.0209)	0.7845 (0.0192)	0.8028 (0.0179)
500	0.7645 (0.0274)	0.7630 (0.0224)	0.7660 (0.0208)	0.7639 (0.0198)	1.0	0.8020 (0.0196)	0.8110 (0.0263)	0.7921 (0.0200)	0.8058 (0.0218)
Orig.	0.7645 (0.0207)	0.7650 (0.0231)	0.7654 (0.0268)	0.7647 (0.0191)	1.0	0.7855 (0.0200)	0.8060 (0.0223)	0.7903 (0.0216)	0.7976 (0.0219)

Table 9: Classification results using $w_{pmi,swn}$ and the best w_{pmi} for Kitchen

Dim	w_{swn}				best w				
	Acc.	Prec.	Rec.	F1	β	Acc.	Prec.	Rec.	F1
50	0.7315 (0.0297)	0.6670 (0.0295)	0.7665 (0.0299)	0.7122 (0.0296)	1.0	0.7710 (0.0213)	0.7050 (0.0250)	0.8120 (0.0242)	0.7639 (0.0201)
100	0.7465 (0.0239)	0.7000 (0.0271)	0.7617 (0.0292)	0.7331 (0.0291)	1.0	0.7890 (0.0129)	0.7450 (0.0192)	0.8167 (0.0173)	0.7711 (0.0185)
200	0.7680 (0.0259)	0.7550 (0.0255)	0.7755 (0.0277)	0.7655 (0.0287)	0.8	0.7975 (0.0230)	0.7450 (0.0203)	0.8315 (0.0227)	0.7853 (0.0176)
300	0.7775 (0.0254)	0.7510 (0.0293)	0.7834 (0.0231)	0.7710 (0.0277)	0.8	0.7990 (0.0145)	0.7440 (0.0249)	0.8345 (0.0190)	0.7862 (0.0160)
500	0.7750 (0.0280)	0.7580 (0.0294)	0.7849 (0.0234)	0.7707 (0.0202)	0.8	0.8040 (0.0165)	0.7570 (0.0207)	0.8353 (0.0122)	0.7936 (0.0160)
Orig.	0.7770 (0.0186)	0.7510 (0.0243)	0.7822 (0.0205)	0.7652 (0.0221)	0.8	0.8000 (0.0200)	0.7520 (0.0247)	0.8314 (0.0172)	0.7892 (0.0232)

Table 10: Classification results using $w_{fisher,swn}$ and the best w_{fisher} for Large Movie Review

Dim	w_{swn}				best w				
	Acc.	Prec.	Rec.	F1	β	Acc.	Prec.	Rec.	F1
50	0.7372 (0.0261)	0.7218 (0.0311)	0.7872 (0.0345)	0.7550 (0.0312)	1.0	0.7926 (0.0219)	0.7761 (0.0321)	0.8224 (0.0251)	0.7986 (0.0251)
100	0.7596 (0.0220)	0.7239 (0.0383)	0.8336 (0.0207)	0.7795 (0.0236)	1.0	0.8142 (0.0265)	0.7872 (0.0326)	0.8536 (0.0294)	0.8206 (0.0237)
200	0.7806 (0.0234)	0.7552 (0.0333)	0.8332 (0.0248)	0.7943 (0.0201)	1.0	0.8306 (0.0263)	0.8151 (0.0365)	0.8552 (0.0279)	0.8343 (0.0237)
300	0.7914 (0.0195)	0.7671 (0.0250)	0.8368 (0.0225)	0.7938 (0.0270)	1.0	0.8335 (0.0222)	0.8159 (0.0365)	0.8652 (0.0280)	0.8349 (0.0285)
500	0.8103 (0.0235)	0.7927 (0.0201)	0.8336 (0.0245)	0.8126 (0.0214)	1.0	0.8408 (0.0233)	0.8237 (0.0250)	0.8672 (0.0217)	0.8436 (0.0264)
Orig.	0.8061 (0.0227)	0.7872 (0.0225)	0.8388 (0.0212)	0.8122 (0.0219)	1.0	0.8352 (0.0279)	0.8148 (0.0334)	0.8672 (0.0222)	0.8403 (0.0237)

Table 11: Classification results using $w_{fisher,swn}$ and the best w_{fisher} for Cornell

Dim	w_{swn}				best w				
	Acc.	Prec.	Rec.	F1	β	Acc.	Prec.	Rec.	F1
50	0.7390 (0.0261)	0.7040 (0.0345)	0.7585 (0.0217)	0.7219 (0.0225)	1.0	0.7935 (0.0162)	0.7890 (0.0336)	0.7993 (0.0268)	0.7921 (0.0192)
100	0.7735 (0.0183)	0.7570 (0.0338)	0.7757 (0.0227)	0.7652 (0.0163)	1.0	0.8140 (0.0210)	0.8100 (0.0353)	0.8184 (0.0269)	0.8123 (0.0253)
200	0.7775 (0.0256)	0.7570 (0.0363)	0.7900 (0.0228)	0.7724 (0.0199)	1.0	0.8260 (0.0231)	0.8170 (0.0282)	0.8346 (0.0250)	0.8233 (0.0292)
300	0.7860 (0.0251)	0.7580 (0.0313)	0.8034 (0.0249)	0.7791 (0.0212)	0.8	0.8260 (0.0285)	0.8220 (0.0325)	0.8308 (0.0264)	0.8245 (0.0218)
500	0.7880 (0.0215)	0.7800 (0.0222)	0.7934 (0.0275)	0.7860 (0.0247)	0.8	0.8375 (0.0252)	0.8210 (0.0325)	0.84450 (0.0237)	0.8317 (0.0223)
Orig.	0.8025 (0.0185)	0.7930 (0.0340)	0.8090 (0.0192)	0.8001 (0.191)	0.8	0.8310 (0.0234)	0.8240 (0.0369)	0.8410 (0.0280)	0.8284 (0.0284)

Table 12: Classification results using $w_{fisher,swn}$ and the best w_{fisher} for Book

Dim	w_{swn}				best w				
	Acc.	Prec.	Rec.	F1	β	Acc.	Prec.	Rec.	F1
50	0.660 (0.0326)	0.6170 (0.0342)	0.6875 (0.0282)	0.6483 (0.0251)	0.8	0.7620 (0.0203)	0.7681 (0.0200)	0.7966 (0.0226)	0.7807 (0.0202)
100	0.6650 (0.0297)	0.5840 (0.0357)	0.6974 (0.0312)	0.6333 (0.0295)	0.8	0.8090 (0.0252)	0.7680 (0.0239)	0.8135 (0.0247)	0.7990 (0.0192)
200	0.6970 (0.0308)	0.6750 (0.0344)	0.7074 (0.0267)	0.6891 (0.0220)	0.8	0.8165 (0.0182)	0.8120 (0.0288)	0.8188 (0.0259)	0.8148 (0.0162)
300	0.7015 (0.0208)	0.69405 (0.0328)	0.7063 (0.0287)	0.6989 (0.0248)	0.8	0.8170 (0.0188)	0.8250 (0.0202)	0.8370 (0.0190)	0.8288 (0.0224)
500	0.7160 (0.0259)	0.7010 (0.0322)	0.7250 (0.0221)	0.7104 (0.0230)	0.8	0.8265 (0.0228)	0.8240 (0.0274)	0.8205 (0.0269)	0.8211 (0.0222)
Orig.	0.7295 (0.0232)	0.7300 (0.0247)	0.7306 (0.0267)	0.7294 (0.0221)	0.8	0.8170 (0.0193)	0.8040 (0.0272)	0.8208 (0.0257)	0.8192 (0.0185)

Table 13: Classification results using $w_{fisher,swn}$ and the best w_{fisher} for DVD

Dim	w_{swn}				best w				
	Acc.	Prec.	Rec.	F1	β	Acc.	Prec.	Rec.	F1
50	0.7150 (0.0285)	0.6490 (0.0378)	0.7517 (0.0323)	0.6949 (0.0261)	0.8	0.7515 (0.029)	0.7810 (0.0256)	0.7389 (0.0271)	0.7590 (0.0202)
100	0.7315 (0.0248)	0.7170 (0.0337)	0.7406 (0.0277)	0.7275 (0.0240)	0.8	0.7605 (0.0109)	0.7890 (0.0250)	0.7477 (0.0189)	0.7674 (0.0093)
200	0.7440 (0.0222)	0.6940 (0.0295)	0.7740 (0.0314)	0.7304 (0.0218)	0.8	0.7725 (0.0169)	0.7820 (0.0302)	0.7678 (0.0239)	0.7742 (0.0184)
300	0.7495 (0.0249)	0.7460 (0.0237)	0.7532 (0.0239)	0.7489 (0.0214)	1.0	0.7735 (0.0202)	0.7840 (0.0206)	0.7681 (0.0259)	0.7756 (0.0182)
500	0.7610 (0.0254)	0.7580 (0.0261)	0.7646 (0.0273)	0.7603 (0.0236)	1.0	0.7885 (0.0173)	0.7950 (0.0204)	0.7707 (0.0254)	0.7820 (0.0199)
Orig.	0.7660 (0.0223)	0.7790 (0.0228)	0.7612 (0.0294)	0.7695 (0.0247)	0.8	0.7780 (0.0221)	0.7970 (0.0219)	0.7687 (0.0211)	0.7820 (0.0204)

Table 14: Classification results using $w_{fisher,swn}$ and the best w_{fisher} for Electronics

Dim	w_{swn}				best w				
	Acc.	Prec.	Rec.	F1	β	Acc.	Prec.	Rec.	F1
50	0.7180 (0.0275)	0.6540 (0.0297)	0.7512 (0.0276)	0.6981 (0.0252)	0.6	0.7695 (0.0195)	0.7850 (0.0253)	0.7626 (0.0282)	0.7725 (0.0204)
100	0.7455 (0.0206)	0.7150 (0.0334)	0.7612 (0.0270)	0.7369 (0.0285)	0.8	0.7830 (0.0277)	0.7940 (0.0218)	0.7780 (0.0232)	0.7850 (0.0267)
200	0.7480 (0.0262)	0.7280 (0.0324)	0.7592 (0.0212)	0.7426 (0.0267)	0.8	0.7825 (0.0226)	0.8010 (0.0291)	0.7737 (0.0251)	0.7861 (0.0263)
300	0.7605 (0.0275)	0.7610 (0.0370)	0.7618 (0.0304)	0.7607 (0.0288)	0.8	0.7855 (0.0217)	0.8075 (0.0216)	0.7836 (0.0237)	0.7965 (0.0233)
500	0.7620 (0.0198)	0.7630 (0.0202)	0.7622 (0.0248)	0.7618 (0.0222)	0.8	0.7985 (0.0104)	0.8110 (0.0272)	0.7872 (0.0196)	0.8033 (0.0101)
Orig.	0.7645 (0.0215)	0.7650 (0.0333)	0.7654 (0.0209)	0.7647 (0.0242)	0.8	0.7895 (0.0198)	0.7780 (0.0285)	0.7964 (0.0229)	0.7840 (0.0210)

Table 15: Classification results using $w_{fisher,swn}$ and the best w_{fisher} for Kitchen

Dim	w_{swn}				best w				
	Acc.	Prec.	Rec.	F1	β	Acc.	Prec.	Rec.	F1
50	0.7215 (0.0285)	0.6570 (0.0284)	0.7565 (0.0275)	0.7022 (0.0266)	1.0	0.7710 (0.0168)	0.7100 (0.0271)	0.8089 (0.0211)	0.7552 (0.0203)
100	0.7370 (0.0230)	0.6900 (0.0271)	0.7624 (0.0279)	0.7235 (0.0286)	1.0	0.7825 (0.0167)	0.7270 (0.0254)	0.8167 (0.0154)	0.7687 (0.0220)
200	0.7580 (0.0290)	0.7450 (0.0310)	0.7654 (0.0275)	0.7543 (0.0225)	1.0	0.7855 (0.0163)	0.7410 (0.0290)	0.8128 (0.0160)	0.7746 (0.0223)
300	0.7680 (0.0258)	0.7420 (0.0286)	0.7833 (0.0208)	0.7615 (0.0202)	0.6	0.7865 (0.0220)	0.7350 (0.0192)	0.8330 (0.0223)	0.7787 (0.0249)
500	0.7625 (0.0258)	0.7430 (0.0202)	0.7747 (0.0208)	0.7574 (0.0186)	0.8	0.7955 (0.0129)	0.7840 (0.0237)	0.8219 (0.0135)	0.7954 (0.0178)
Orig.	0.7700 (0.0186)	0.7510 (0.0243)	0.7822 (0.0205)	0.7652 (0.0221)	0.8	0.7910 (0.0170)	0.7570 (0.0252)	0.8329 (0.079)	0.7854 (0.0221)

Table 16: Classification results without feature weighting

Dim	Large Movie Review				Cornell				Book			
	Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1
50	0.7222 (0.0282)	0.6826 (0.0285)	0.7904 (0.0269)	0.7362 (0.0211)	0.7340 (0.0171)	0.6940 (0.0340)	0.7576 (0.0191)	0.7225 (0.0297)	0.6560 (0.0228)	0.6790 (0.0228)	0.6504 (0.0276)	0.6634 (0.0234)
100	0.7380 (0.0164)	0.7085 (0.0287)	0.7952 (0.0217)	0.7493 (0.0234)	0.7550 (0.0258)	0.7380 (0.0302)	0.7660 (0.0286)	0.7496 (0.0250)	0.6850 (0.0252)	0.6700 (0.0227)	0.6920 (0.0220)	0.6796 (0.0207)
200	0.7570 (0.0273)	0.7444 (0.0260)	0.8244 (0.0262)	0.7698 (0.0209)	0.7885 (0.0214)	0.7690 (0.0272)	0.8020 (0.0270)	0.7839 (0.0256)	0.6890 (0.0280)	0.6900 (0.0222)	0.6912 (0.0221)	0.6894 (0.0210)
300	0.7730 (0.0265)	0.7665 (0.0260)	0.8536 (0.0248)	0.7875 (0.0213)	0.7950 (0.0290)	0.7660 (0.0204)	0.8155 (0.0199)	0.7878 (0.0274)	0.7005 (0.0202)	0.6880 (0.0337)	0.7070 (0.0228)	0.6963 (0.0251)
500	0.7868 (0.0237)	0.7611 (0.0237)	0.8360 (0.0242)	0.7968 (0.0257)	0.7935 (0.0153)	0.7690 (0.0395)	0.8175 (0.0240)	0.7858 (0.0224)	0.7135 (0.0145)	0.7160 (0.0315)	0.7157 (0.0242)	0.7133 (0.0221)
Orig.	0.7872 (0.0183)	0.7728 (0.0223)	0.8364 (0.0168)	0.7985 (0.0252)	0.7990 (0.0202)	0.7900 (0.0206)	0.8050 (0.0196)	0.7968 (0.0238)	0.7200 (0.0168)	0.7240 (0.0329)	0.7217 (0.0286)	0.7198 (0.0245)

Dim	DVD				Electronics				Kitchen			
	Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1
50	0.7090 (0.0198)	0.6590 (0.0264)	0.7336 (0.0271)	0.6931 (0.0266)	0.6990 (0.0291)	0.6640 (0.0302)	0.7142 (0.0252)	0.6878 (0.0267)	0.7275 (0.0252)	0.6520 (0.0284)	0.7708 (0.0263)	0.7059 (0.0252)
100	0.7260 (0.0251)	0.6930 (0.0214)	0.7429 (0.0216)	0.7163 (0.0182)	0.7355 (0.0279)	0.7540 (0.0277)	0.7268 (0.0242)	0.7397 (0.0211)	0.7595 (0.0132)	0.7180 (0.0217)	0.7848 (0.0208)	0.7493 (0.0158)
200	0.7375 (0.0251)	0.7330 (0.0270)	0.7410 (0.0188)	0.7356 (0.0198)	0.7415 (0.0224)	0.7500 (0.0230)	0.7383 (0.0197)	0.7535 (0.0215)	0.7610 (0.0184)	0.7270 (0.0276)	0.7824 (0.0206)	0.7532 (0.0197)
300	0.7450 (0.0233)	0.7590 (0.0243)	0.7382 (0.0180)	0.7476 (0.0193)	0.7485 (0.0257)	0.7640 (0.0220)	0.7408 (0.260)	0.7517 (0.179)	0.7610 (0.0156)	0.7370 (0.0295)	0.7770 (0.0228)	0.7556 (0.0144)
500	0.7440 (0.0198)	0.7410 (0.0254)	0.7505 (0.0252)	0.7402 (0.0193)	0.7505 (0.0255)	0.7630 (0.0270)	0.7456 (0.0233)	0.7534 (0.0232)	0.7615 (0.0184)	0.7400 (0.0243)	0.7746 (0.0214)	0.7565 (0.0209)
Orig.	0.7580 (0.0218)	0.7480 (0.0271)	0.7635 (0.0200)	0.7554 (0.0250)	0.7495 (0.0269)	0.7710 (0.0270)	0.7587 (0.0264)	0.7642 (0.0214)	0.7720 (0.0181)	0.7590 (0.0219)	0.7805 (0.0218)	0.7689 (0.0212)

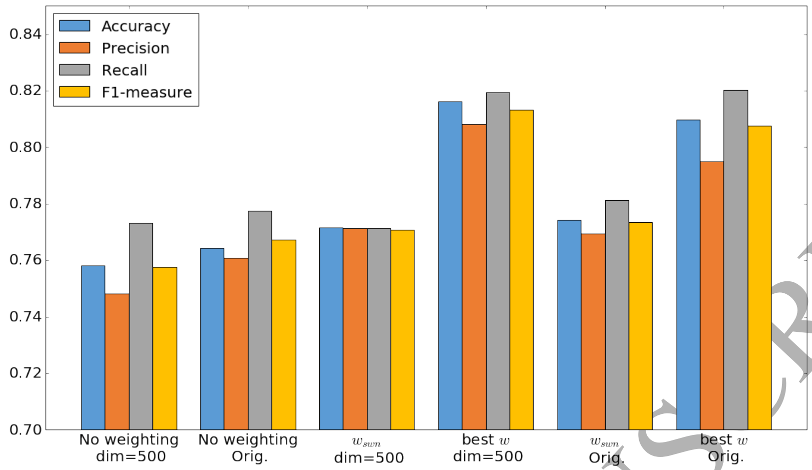


Figure 3: Accuracy, precision, recall and F1-measure comparison (values averaged over 6 datasets)

Figure 3 compares the results for the datasets without feature extraction and for those datasets that include 500 features extracted by the proposed method and averaged over six datasets. The best results from the proposed framework, regardless of the applied feature weighting methods (e.g., PMI and FDR), were considered in averaging. The best performance was achieved when the dimensionality of the datasets was reduced to 500 by using SS-MMC and the best weights of terms. The best weights were usually obtained when w_{label} had a large impact on the final weights; therefore, those weights that consider label information can enhance the performance of classifiers.

The experimental results of other recent studies that apply SVM are presented in Table 17. The proposed framework is fairly competitive compared with those other frameworks proposed in recent studies on sentiment analysis. In Wang et al. (2014) and Singh and Shahid Husain (2014), the accuracies for the Cornell dataset reached 81.94% and 81.15% when SVM was used as a classification model; both of these accuracies were lower than the best accuracies achieved by the proposed

Table 17: Experimental results of other recent research using SVM

Ref	Technique	Dataset	Accuracy (%)
Wang et al. (2014)	Random subspace with SVM using term frequency	Cornell	81.94
Singh and Shahid Husain (2014)	SVM	Cornell	81.15
Zhou et al. (2014)	Transductive SVM	Cornell	68.70
Kalaivani and Shunmuganathan (2015)	SVM using unigram	Book DVD	74.95 76.87
Khan et al. (2016)	SVM with semi-supervised feature weighting	Cornell Large Movie Review Book DVD	81.84 85.78 74.83 77.50
Proposed framework	SVM with feature-weighted SS-MMC (reduced dim = 500)	Cornell Large Movie Review Book DVD	84.45 84.26 82.05 78.35

framework. Wang et al. (2014) obtained this performance using random subspace, one of the ensemble algorithms. In Zhou et al. (2014), transductive SVM achieved a 68.70% accuracy for the Cornell dataset. Kalaivani and Shunmuganathan (2015) evaluated the classification results from different classifiers such as NB, SVM and logistic regression with Bagging. In this paper, the reported accuracies when SVM with unigrams were applied for the Book and DVD datasets were 74.95% and 76.87%, respectively. In Khan et al. (2016), the 10-fold cross-validation accuracies for the Cornell, Large Movie Review, Book, and DVD datasets reached 81.84%, 85.78%, 74.83%, and 77.50%, respectively, without intelligent model selection. Although a better performance was obtained by applying intelligent model selection, the proposed framework obtained higher accuracies in the reduced dimensionality for the Cornell, Book, and DVD datasets. The comparison with other methods verifies that the well-defined weights of features can enhance the sentiment classification performance of feature extraction. In the proposed method, the over-fitting of weights was prevented by the baseline weights and by using SentiWordNet, which can further improve the performance of models.

In addition to evaluation in terms of classification performance, computation time was also measured. The total computing times using the proposed framework can be divided into two parts; 1) dimensionality reduction and 2) SVM training. SVM training is the common process regardless of whether or not dimensionality reduction is conducted, but the training time varies depending on the dimensionality of data. In order to measure the effect of dimensionality reduction, the average computation times for dimensionality reduction and SVM training were separately calculated for each dataset when the reduced dimensionality was set to 300 or 500 and these were compared with the computation times for SVM training using the raw term frequency matrices. As observed in Figure 4, when the number of samples is not large, the time required to reduce the dimensionality of data is larger than computation time reduction in SVM training. On the other hand, when the number of samples is large, computation time reduction in SVM training by reducing dimensionality becomes large.

Table 18 and 19 show the top 10 words with the highest weights as computed by PMI and FDR, respectively. POS was applied to distinguish the terms because similar terms may have different polarities depending on POS. Therefore, 'term#POS' is an independent feature where 'a', 'n', 'r', and 'v' represent the adjective, noun, adverb, and verb for POS tags, respectively.

The average weights for all experiments were used to select the top 10 terms. The selected positive and negative words generally agreed with the typical senti-

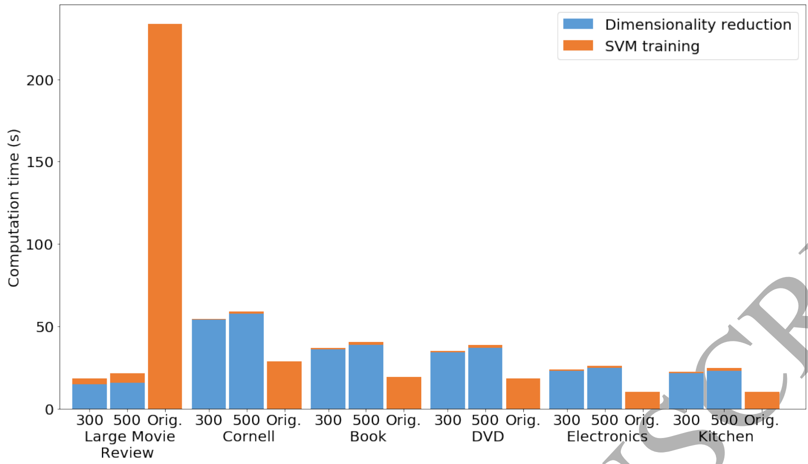


Figure 4: Average computation times (the computation times for Large Moview Review are one-hundredth of the actual)

ment of words, but using several words to express sentiment is only suitable and significant for specific contexts. For the Large Movie Review, Cornell, Book and DVD datasets, those words related to impression or emotion obtained relatively high weights. Meanwhile, the words related to utility or performance in the Electronics and Kitchen datasets greatly outnumbered those in the Book and DVD datasets.

Interestingly, “bad”, “good”, “great”, “worst”, and several other words that are commonly used to show positive or negative sentiments in a universal situation were not always ranked among the top 10 terms. This result might be due to the fact that both statistical methods give more weight to those words that are used to express a certain sentiment. Given that the commonly used words for expressing emotions are widely used in both positive and negative documents, these words can express the opposite sentiment when combined with negation words, such as “no” and “not”.

Table 18: Top 10 words with the highest weights for each category by PMI

Large Movie Review	negative	bad#a, angriness#n, worst#a, distressful#a, waste#n, deface#v, scrawl#n, maliciousness#n, inconvenience#v, awful#a
	positive	unsurpassable#a, gentle#v, curative#a, eminence#n, great#a, measurable#a, propulsive#a, reconstructive#a, levelheaded#a, jolliness#n
Cornell	negative	bad#a, waste#n, worst#a, stupid#a, bullshit#v, reek#v, groan#v, alas#r, irrefutable#a, ridiculous#a
	positive	outstanding#a, courteous#a, elegance#n, incumbent#a, sensational#a, wonderfully#r, excellent#a, dedication#n, wonderful#a, perfectly#r
Book	negative	bad#a, waste#n, poorly#r, claim#n, cad#n, stupid#a, unfortunately#r, canard#n, ridiculous#a, schlock#n
	positive	bliss#n, sensational#a, easy#a, wonderful#a, excellent#a, favorite#a, highly#r, therapeutic#a, superlative#a, enjoyable#a
DVD	negative	waste#n, worst#a, bad#a, horrible#a, malice#n, smell#v, awful#a, regrettably#r, moan#v, schlock#n
	positive	kudos#n, congratulations#n, impressively#r, upright#a, superlative#a, sterling#a, wonderful#a, mummy#n, fortunate#a, excellent#a
Electronics	negative	burden#n, regrettably#r, muddy#v, waste#n, smell#v, notorious#a, terrible#a, spite#n, unfortunate#a, error#n
	positive	excellent#a, dandy#a, kudos#n, highly#r, great#a, courteous#a, superb#a, impressively#r, price#n, upright#a
Kitchen	negative	horrible#a, woefully#r, regrettably#r, outrageous#a, chintzy#a, atrocious#a, scary#a, accumulation#n, insufficient#a, unrealistic#a
	positive	easy#a, greatest#a, superb#a, praise#n, enamored#a, definite#a, mild#a, finer#a, impressively#r, perfect#a

Table 19: Top 10 words with the highest weights for each category by FDR

Large Movie Review	negative	bad#a, unsound#a, waste#n, irony#a, worst#a, awful#a, negative#a, terrible#a, unchallengeable#a, woefully#r
	positive	great#a, impressive#a, life#n, loveliness#n, gentle#a, helpful#a, playable#a, outstanding#a, wonderful#a, good#a
Cornell	negative	bullshit#v, worst#a, reek#v, egregious#a, untraditional#a, undependable#a, stupid#a, bad#a, poorly#r
	positive	life#n, wonderfully#r, dignify#v, great#a, props#n, excellent#a, performance#n, memorable#a, impressively#r, elegance#n
Book	negative	bad#a, waste#n, awful#a, poorly#r, unorganized#a, undeniable#a, wrong#v, disappointing#a, disability#n, compassionate#v
	positive	sensational#a, easy#a, bliss#n, superlative#a, highly#r, reputable#a, childlike#a, wonderful#a, love#n, upright#a, cheerful#a
DVD	negative	abduction#n, hound#n, schlock#n, gloat#n, demerit#n, regrettably#r, resentful#a, disapprove#v, fright#n, scrawl#n
	positive	sensational#a, impressively#r, bliss#n, congratulations#n, great#a, premium#a, altruistic#a, plus#a, finer#a, fortunate#a
Electronics	negative	hound#n, muddy#v, burden#n, regrettably#r, spite#n, woefully#r, notorious#a, bush#a, scuff#n, horrific#a
	positive	work#n, praise#n, mild#a, dandy#a, great#a, impressively#r, thorough#a, price#n, upright#a, highly#r
Kitchen	negative	horrible#a, woefully#r, regrettably#r, outrageous#a, chintzy#a, atrocious#a, scary#a, accumulation#n, insufficient#a, unrealistic#a
	positive	easy#a, greatest#a, superb#a, praise#n, enamored#a, definite#a, mild#a, finer#a, impressively#r, perfect#a

5. Discussion

The proposed framework in this study is mainly based on a semi-supervised feature extraction method with a semi-supervised feature weighting as a pre-processing step for reducing dimensionality in a sentiment classification system, while most of the existing expert systems for sentiment classification apply feature selection instead of feature extraction Wang et al. (2007, 2011); Khan et al. (2016); Akhtar et al. (2017). In fact, the popularity of feature selection methods in sentiment analysis is driven by the limitations of the extant feature extraction methods. Those feature extraction methods that belong to manifold learning aim to explain the original data structure as much as possible with a small number of new features. Therefore, these algorithms are usually unsupervised and inappropriate to be used for the preprocessing step before predictive modeling. Meanwhile, given that feature selection aims to find the best feature set to be used in model construction, the selection procedures are dependent on label information. Moreover, compared with feature extraction methods, feature selection methods require a larger number of features to maintain the performance of models. The proposed framework complements the disadvantages of feature extraction and facilitates the building of an expert system for sentiment classification. In this framework, SS-MMC plays the important roles of providing a mapping from the original space to the reduced space and reflecting both the label information and structural information of the input data.

Given that SS-MMC provides a linear mapping from the original space to the reduced space, the obtained features are actually the linear combination of the original features. Although some features have a very marginal impact on the new features, they all contribute to the new features and it is also possible to rank features based on their contributions. Therefore, SS-MMC can be viewed as the second step of feature weighting after the feature weighting using the graph Laplacian of the graph that depicts the relations among features. The weights of terms in feature weighting step of the proposed framework are calculated in consideration of termlevel relations, but SS-MMC utilizes documentlevel relations and partial label information. Two levels of relations increase the significance of the extracted features for the model training, and the superiority of the proposed method may be ascribed to the lower amount of information loss by feature extraction and to the prevention of overfitting by considering the relations between terms and documents. The general polarity information of sentiment terms as provided by SentiWordNet also partially contributes to the improvement of feature weighting.

The major contributions of this research are outlined as follows:

- The proposed framework builds sentiment classification models with the better features constructed based on feature extraction instead of feature selection. In this way, this framework maintains the high discriminating power of feature extraction in solving classification problems. In addition, the proposed framework utilizes SS-MMC, one of linear feature extraction methods, so the proposed framework for the intelligent sentiment classification system also enhances the interpretability and usability of the trained classifiers.

- Unlike other filter feature selection methods that calculate the importance of features to distinguish the more important and relevant features from the less relevant ones, the proposed framework obtains the weights of features in a semi-supervised manner. In the proposed feature weighting algorithm, the weights of features that only appear in unlabeled documents are inferred from those of features in labeled documents. The polarity scores of features that are defined in SentiWordNet are also utilized to provide the general polarity scores of features regardless of their domains.

- To demonstrate the usefulness and applicability of the proposed semi-supervised feature extraction with the semi-supervised feature weighting for sentiment analysis, I conducted the extensive experiments over the six benchmark datasets. The results showed that the framework using the proposed method improved the performance of sentiment classification.

The proposed framework can be further improved by introducing procedures that update the weights of features and extracted features through the addition of new unlabeled and labeled documents. If feature weights and extracted features can be automatically updated, the proposed framework can take advantage of the benefit that new documents are easily collected from the Internet. In addition, it is relatively easy to retrain the model due to feature update, because the proposed method uses a fewer number of features than the system using feature selection.

However, the proposed framework for the intelligent sentiment classification system also has some weaknesses, which are listed as follows.

- The proposed framework only considers unigrams with limited POS tags. However, previous studies have proven that some phrases have high discriminating power and can indicate a strong sentiment (Pang et al., 2002; Cui et al., 2006; Xia et al., 2011; Yousefpour et al., 2017). For instance, Cui et al. (2006) showed that a discriminating classifier combined with high-order n-grams as features can achieve a comparable, or even a better, performance than other classifiers. Xia et al. (2011) compared different feature sets and found that a feature set with bigrams could enhance the classification accuracy much better than a feature set with unigrams and the joint feature set with unigrams, bigrams and dependency relation features could further improve the performance of sentiment classifiers. In Yousefpour et al. (2017), unigrams, bigrams, and trigrams matched with predetermined POS patterns were extracted to construct the initial feature set, which was then compared with another feature set that only included unigrams. They proved that the feature set with n-grams (n¡4) could improve sentiment classification performance.

- The proposed framework utilizes very basic machine learning techniques. In previous literature, ensemble algorithms could enhance the performance of classification algorithms in sentiment analysis as well as in other application areas (Wang et al., 2014; Fersini et al., 2014; Wan and Gao, 2015; Lochter et al., 2016). For instance, Wang et al. (2014) tested well-known ensemble methods in sentiment analysis, such as bagging, boosting, and random subspace, and revealed that these methods substantially improve the performance of individual base learners for sentiment classification. Fersini et al. (2014) proposed the Bayesian ensemble approach by applying Bayesian model averaging to aggregate the results from several weak learners and Lochter et al. (2016) proposed an ensemble method that averaged the base classifiers that were trained by using several different training sets consisting of different feature sets that were generated by different rules. Both cases proved that the ensemble approach can enhance the performance of classification algorithms in sentiment analysis.

- The proposed framework is only applicable for single-label binary classification (positive or negative). However, Liu et al. (2017) built a framework that classified the sentiment of documents into three to five sentiment classes while Liu and Chen (2015) proposed a multi-label classification-based approach for sentiment analysis that classified the sentiment of microblogs into a set of sentiment labels.

6. Conclusion

To build expert and intelligent systems for sentiment classification, dimensionality reduction is required to avoid the "curse of dimensionality" that frequently occurs in text mining applications when the vector space model is used to convert text documents to vectors of identifiers. Therefore, it is crucial to develop enhanced dimensionality reduction procedures that can be applied to both training and test phases for the systems. Apart from guaranteeing better interpretability, feature extraction can also reduce information loss and achieve a higher discriminating power in sentiment classification. However, feature extraction has attracted limited application in sentiment analysis because many nonlinear feature extraction methods, which generally outperform linear methods, only consider input features and are not applicable for a test phase because of lack of the mapping function.

To address the problems of existing feature extraction methods and create better features, a new semi-supervised dimensionality reduction framework that efficiently utilizes feature extraction was proposed in this paper. In the proposed framework, feature weighting was applied in the proposed framework before conducting feature extraction, and both feature weighting and feature extraction were performed in a semi-supervised manner. The polarities of terms in unlabeled documents were estimated using the graph Laplacian of the graph that captures word-word relations as well as the polarities of terms computed from labeled documents using PMI and FDR. The extensive experiments proved that the weights of features obtained by the proposed method improved the sentiment classification performance regardless of the applied weighting methods (PMI and FDR) or datasets. Moreover, it was also observed that the general polarity scores of terms defined in SentiWordNet improved the feature weighting in many cases and the feature weighting helped SS-MMC create better features.

Possible directions for future research include the following. First, the similarities among the words that are used for weights in the graph of terms can be adjusted in a more sophisticated manner in addition to co-occurrence. Second, a more intensive feature engineering approach must be adopted in the proposed framework. Bigrams, trigrams, or high-order n-grams may further improve the performance of this framework and can be used to validate the effectiveness of different POS patterns in constructing the initial feature set. Third, applying ensemble algorithms can further increase the classification accuracy as proven in some studies.

Acknowledgement

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government (MSIP; Ministry of Science, ICT & Future Planning) (No. 2017R1C1B5014805).

References

Abbasi, A., Chen, H., Salem, A., 2008. Sentiment Analysis in Multiple Languages: Feature Selection for Opinion Classification in Web Forums. ACM Trans. Inf. Syst. 26 (3), 12:1—-12:34.

Agarwal, B., Mittal, N., 2013a. Optimal Feature Selection for Sentiment Analysis. In: Gelbukh, A. (Ed.), Computational Linguistics and Intelligent Text Processing. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 13–24.

Agarwal, B., Mittal, N., 2013b. Sentiment Classification using Rough Set based Hybrid Feature Selection. In: the 4th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis. pp. 115–119.

Ahmad, S. R., Yusop, N. M. M., Bakar, A. A., Yaakub, M. R., 2017. Statistical analysis for validating ACO-KNN algorithm as feature selection in sentiment analysis. Vol. 1891. p. 20018.

Akhtar, M. S., Gupta, D., Ekbal, A., Bhattacharyya, P., 2017. Feature selection and ensemble construction: A two-step method for aspect based sentiment analysis. Knowledge-Based Systems 125, 116–135.

Asghar, M. Z., Khan, A., Ahmad, S., Kundi, F. M., 2014. A Review of Feature Extraction in Sentiment Analysis. J. Basic. Appl. Sci. Res 4 (3), 181–186.

Baccianella, S., Esuli, A., Sebastiani, F., 2010. SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. In: Proceedings of the Seventh Conference on International Language Resources and Evaluation. European Language Resources Association, pp. 2200–2204.

Basari, A. S. H., Hussin, B., Ananta, I. G. P., Zeniarja, J., 2013. Opinion Mining of Movie Review using Hybrid Method of Support Vector Machine and Particle Swarm Optimization. Procedia Engineering 53, 453–462.

Belkin, M., Matveeva, I., Niyogi, P., 2004. Regularization and Semi-supervised Learning on. Colt 2004, 624–638.

Belkin, M., Niyogi, P., 2003. Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. Neural Comput. 15 (6), 1373–1396.

Boiy, E., Moens, M. F., 2009. A machine learning approach to sentiment analysis in multilingual web texts. Information Retrieval 12 (5), 526–558.

Boldrini, E., Balahur, A., Martínez-Barco, P., Montoyo, A., mar 2012. Using EmotiBlog to annotate and analyse subjectivity in the new textual genres. Data Mining and Knowledge Discovery 25 (3), 603–634.

Bouma, G., 2009. Normalized (pointwise) mutual information in collocation extraction. In: International Conference of the German Society for Computational Linguistics and Language Technology. pp. 31–40.

Cai, D., He, X., Han, J., 2005. Document clustering using locality preserving indexing. Knowledge and Data Engineering, IEEE Transactions on 17 (12), 1624–1637.

Cardie, C., Farina, C., Bruce, T., 2006. Using natural language processing to improve eRulemaking: project highlight. In: Proceedings of the 2006 international conference on Digital government research. pp. 177–178.

Chaovalit, P., Zhou, L., 2005. Movie Review Mining: a Comparison between Supervised and Unsupervised Classification Approaches. In: Proceedings of the 38th Annual Hawaii International Conference on System Sciences. pp. 112c–112c.

Chen, L.-F., Su, C.-T., Chen, K.-H., jan 2012. An improved particle swarm optimization for feature selection. Intelligent Data Analysis 16 (2), 167–182.

Cui, H., Mittal, V., Datar, M., 2006. Comparative experiments on sentiment classification for online product reviews. In: AAAI. Vol. 6. pp. 1265–1270.

Feldman, R., Apr. 2013. Techniques and applications for sentiment analysis. Communications of the ACM 56 (4), 82.

Fersini, E., Messina, E., Pozzi, F. A., 2014. Sentiment analysis: Bayesian Ensemble Learning. Decision Support Systems 68, 26–38.

Godbole, N., Srinivasaiah, M., 2007. Large-scale sentiment analysis for news and blogs. Conference on Weblogs and Social Media (ICWSM 2007), 219–222.

Gupta, D. K., Reddy, K. S., Ekbal, A., Others, 2015. Pso-asent: Feature selection using particle swarm optimization for aspect based sentiment analysis. In: International Conference on Applications of Natural Language to Information Systems. Springer, pp. 220–233.

Guyon, I., Elisseeff, A., 2003. An Introduction to Variable and Feature Selection. Journal of Machine Learning Research 3 (Mar), 1157–1182.

Hamdan, H., Bellot, P., Bechet, F., 2015. lsislif: Feature extraction and label weighting for sentiment analysis in twitter. In: Proceedings of the 9th International Workshop on Semantic Evaluation. pp. 568–573.

Heerschop, B., Goossen, F., Hogenboom, A., Frasincar, F., Kaymak, U., de Jong, F., 2011. Polarity Analysis of Texts Using Discourse Structure. In: Proceedings of the 20th ACM International Conference on Information and Knowledge Management. CIKM '11. ACM, New York, NY, USA, pp. 1061–1070.

Hira, Z. M., Gillies, D. F., 2015. A Review of Feature Selection and Feature Extraction Methods Applied on Microarray Data. Advances in bioinformatics 2015, 198363.

Hu, M., Liu, B., 2004. Mining and Summarizing Customer Reviews. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '04. ACM, New York, NY, USA, pp. 168–177.

Huang, S., Ward, M., Rundensteiner, E., july 2005. Exploration of dimensionality reduction for text visualization. In: Proceedings of the 3rd International Conference on Coordinated and Multiple Views in Exploratory Visualization. pp. 63 – 74.

Jun, S., Park, S.-S., Jang, D.-S., jun 2014. Document clustering method using dimension reduction and support vector clustering to overcome sparseness. Expert Systems with Applications 41 (7), 3204–3212.

Kalaivani, P., Shunmuganathan, K. L., 2015. Feature Reduction Based on Genetic Algorithm and Hybrid Model for Opinion Mining. Scientific Programming 2015, 1–15.

Khan, F. H., Qamar, U., Bashir, S., 2016. SWIMS: Semi-supervised subjective feature weighting and intelligent model selection for sentiment analysis. Knowledge-Based Systems 100, 97–111.

Kim, K., Lee, J., 2014. Sentiment visualization and classification via semi-supervised nonlinear dimensionality reduction. Pattern Recognition 47 (2), 758 – 768.

Kummer, O., Savoy, J., 2012. Feature Selection in Sentiment Analysis. in Proc. CORIA, 273–284.

Lane, P. C., Clarke, D., Hender, P., 2012. On developing robust models for favourability analysis: Model choice, feature sets and imbalanced data. Decision Support Systems 53 (4), 712–718.

Li, H., Jiang, T., Zhang, K., jan 2006. Efficient and Robust Feature Extraction by Maximum Margin Criterion. IEEE Transactions on Neural Networks 17 (1), 157–165.

Li, T., Zhang, Y., Sindhwani, V., 2009. A non-negative matrix tri-factorization approach to sentiment classification with lexical prior knowledge. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1. Association for Computational Linguistics, pp. 244–252.

Li, X., Xie, H., Chen, L., Wang, J., Deng, X., oct 2014. News impact on stock price return via sentiment analysis. Knowledge-Based Systems 69, 14–23.

Liu, S. M., Chen, J.-H., 2015. A multi-label classification based approach for sentiment classification. Expert Systems with Applications 42 (3), 1083–1093.

Liu, Y., Bi, J.-W., Fan, Z.-P., 2017. Multi-class sentiment classification: The experimental comparisons of feature selection and machine learning algorithms. Expert Systems with Applications 80, 323–339.

Lochter, J. V., Zanetti, R. F., Reller, D., Almeida, T. A., 2016. Short text opinion detection using ensemble of classifiers and semantic indexing. Expert Systems with Applications 62, 243–249.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., Potts, C., 2011. Learning Word Vectors for Sentiment Analysis. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. pp. 142–150.

Manek, A. S., Shenoy, P. D., Mohan, M. C., Venugopal, K. R., 2017. Aspect term extraction for sentiment analysis in large movie reviews using Gini Index feature selection method and SVM classifier. World Wide Web 20 (2), 135–154.

Mao, Y., Balasubramanian, K., Lebanon, G., 2010. Dimensionality reduction for text using domain knowledge. In: Proceedings of the 23rd International Conference on Computational Linguistics: Posters. Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 801–809.

Martinsson, P.-G., Rokhlin, V., Tygert, M., 2011. A randomized algorithm for the decomposition of matrices. Applied and Computational Harmonic Analysis 30 (1), 47–68.

Medhat, W., Hassan, A., Korashy, H., 2014. Sentiment analysis algorithms and applications: A survey. Ain Shams Engineering Journal 5 (4), 1093–1113.

Mishne, G., Glance, N., 2005. Predicting Movie Sales from Blogger Sentiment. AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs, 155–158.

Momtazi, S., Khudanpur, S., Klakow, D., 2010. A Comparative Study of Word Co-occurrence for Term Clustering in Language Model-based Sentence Retrieval. In: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics. HLT '10. Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 325–328.

Moraes, R., Valiati, J. F., Gavião Neto, W. P., 2013. Document-level sentiment classification: An empirical comparison between SVM and ANN. Expert Systems with Applications 40 (2), 621–633.

Moreo, A., Romero, M., Castro, J., Zurita, J., 2012. Lexicon-based Comments-oriented News Sentiment Analyzer system. Expert Systems with Applications 39 (10), 9166–9180.

Mudinas, A., Zhang, D., Levene, M., 2012. Combining Lexicon and Learning Based Approaches for Concept-level Sentiment Analysis. In: Proceedings of the First International Workshop on Issues of Sentiment Discovery and Opinion Mining. WISDOM '12. ACM, New York, NY, USA, pp. 5:1—-5:8.

Mukherjee, S., Bhattacharyya, P., 2012. Feature Specific Sentiment Analysis for Product Reviews. In: Gelbukh, A. (Ed.), Computational Linguistics and Intelligent Text Processing SE - 39. Vol. 7181 of Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 475–487.

Nie, F., Wang, H., Huang, H., Ding, C., 2013. Adaptive loss minimization for semi-supervised elastic embedding. In: Proceedings of the Twenty-Third international joint conference on Artificial Intelligence. AAAI Press, pp. 1565–1571.

O'Keefe, T., Koprinska, I., 2009. Feature selection and weighting methods in sentiment analysis. Australasian Document Computing Symposium, 67.

Ortigosa, A., Martín, J. M., Carro, R. M., Feb. 2014. Sentiment analysis in Facebook and its application to e-learning. Computers in Human Behavior 31, 527–541.

Pang, B., Lee, L., 2008. Opinion Mining and Sentiment Analysis. Foundations and Trends in Information Retrieval 1 (2), 1–135.

Pang, B., Lee, L., Vaithyanathan, S., 2002. Thumbs Up?: Sentiment Classification Using Machine Learning Techniques. In: Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10. EMNLP '02. Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 79–86.

Poria, S., Cambria, E., Gelbukh, A., 2015. Deep Convolutional Neural Network Textual Features and Multiple Kernel Learning for Utterance-level Multimodal Sentiment Analysis. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. No. September. pp. 2539–2544.

Roweis, S. T., Saul, L. K., Dec. 2000. Nonlinear dimensionality reduction by locally linear embedding. Science (New York, N.Y.) 290 (5500), 2323–6.

Schölkopf, B., Smola, A., Müller, K. R., 1998. Nonlinear Component Analysis as a Kernel Eigenvalue Problem. Neural Computation 10 (5), 1299–1319.

Sharma, A., Dey, S., 2012. Performance investigation of feature selection methods and sentiment lexicons for sentiment analysis. IJCA Special Issue on Advanced Computing and Communication Technologies for HPC Applications 3, 15–20.

Singh, P. K., Shahid Husain, M., 2014. Methodological Study Of Opinion Mining And Sentiment Analysis Techniques. International Journal on Soft Computing 5 (1), 11–21.

Siti Rohaidah Ahmad, Azuraliza Abu Bakar, Mohd Ridzwan Yaakub, 2015. Metaheuristic Algorithms for Feature Selection in Sentiment Analysis. Science and Information Conference 2015, 222–226.

Song, Y., Nie, F., Zhang, C., Xiang, S., Sep. 2008. A unified framework for semi-supervised dimensionality reduction. Pattern Recognition 41 (9), 2789–2799.

Taboada, M., Brooke, J., Tofiloski, M., Voll, K., Stede, M., 2011. Lexicon-Based Methods for Sentiment Analysis. Computational Linguistics 37 (2), 267–307.

Tan, S., Zhang, J., 2008. An empirical study of sentiment analysis for chinese documents. Expert Systems with Applications 34 (4), 2622–2629.

Tatemura, J., 2000. Virtual reviewers for collaborative exploration of movie reviews. In: International Conference on Intelligent User Interfaces, Proceedings IUI. ACM, pp. 272–275.

Tenenbaum, J. B., 1998. Mapping a Manifold of Perceptual Observations. In: Proceedings of the 1997 Conference on Advances in Neural Information Processing Systems 10. NIPS '97. MIT Press, Cambridge, MA, USA, pp. 682–688.

Thelwall, M., Buckley, K., Paltoglou, G., Feb. 2011. Sentiment in Twitter events. Journal of the American Society for Information Science and Technology 62 (2), 406–418.

Tsutsumi, K., Shimada, K., Endo, T., 2007. Movie review classification based on a multiple classifier. In: Proceedings of the 21st Pacific Asia Conference on Language, Information and Computation. pp. 481–488.

Varela, P. L., Martins, A. F., Aguiar, P. M., Figueiredo, M. A., 2013. An empirical study of feature selection for sentiment analysis. In: Proceedings of the 9th conference on Telecommunications, Conftele, Castelo Branco.

Wan, Y., Gao, Q., 2015. An ensemble sentiment classification system of twitter data for airline services analysis. In: Data Mining Workshop (ICDMW), 2015 IEEE International Conference on. IEEE, pp. 1318–1325.

Wang, G., Sun, J., Ma, J., Xu, K., Gu, J., 2014. Sentiment classification: The contribution of ensemble learning. Decision Support Systems 57, 77–93.

Wang, S., Li, D., Song, X., Wei, Y., Li, H., 2011. A feature selection method based on improved fisher's discriminant ratio for text sentiment classification. Expert Systems with Applications 38 (7), 8696–8702.

Wang, S., Wei, Y., Li, D., Zhang, W., Li, W., 2007. A Hybrid Method of Feature Selection for Chinese Text Sentiment Classification. In: Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007). Vol. 3. pp. 435–439.

Whitehead, M., Yaeger, L., 2010. Sentiment Mining Using Ensemble Classification Models. Springer Netherlands, Dordrecht, pp. 509–514.

Whitelaw, C., Garg, N., Argamon, S., 2005. Using Appraisal Groups for Sentiment Analysis. In: Proceedings of the 14th ACM International Conference on Information and Knowledge Management. CIKM '05. ACM, New York, NY, USA, pp. 625–631.

Wiebe, J., Wilson, T., Cardie, C., 2005. Annotating Expressions of Opinions and Emotions in Language. Language Resources and Evaluation 39 (2), 165–210.

Xia, R., Zong, C., Li, S., 2011. Ensemble of feature sets and classification algorithms for sentiment classification. Information Sciences 181 (6), 1138–1152.

Yang, X., Fu, H., Zha, H., Barlow, J., 2006. Semi-supervised Nonlinear Dimensionality Reduction. In: Proceedings of the 23rd International Conference on Machine Learning. ICML '06. ACM, New York, NY, USA, pp. 1065–1072.

Ye, Q., Zhang, Z., Law, R., 2009. Sentiment classification of online reviews to travel destinations by supervised machine learning approaches. Expert Systems with Applications 36 (3), 6527–6535.

Yousefpour, A., Ibrahim, R., Hamed, H. N. A., 2017. Ordinal-based and frequency-based integration of feature selection methods for sentiment analysis. Expert Systems with Applications 75, 80–93.

Zainuddin, N., Selamat, A., Ibrahim, R., 2016. Twitter Feature Selection and Classification Using Support Vector Machine for Aspect-Based Sentiment Analysis. Springer International Publishing, Cham, pp. 269–279.

Zha, Z.-J., Mei, T., Wang, J., Wang, Z., Hua, X.-S., 2009. Graph-based semi-supervised learning with multiple labels. Journal of Visual Communication and Image Representation 20 (2), 97–103.

Zhao, M., Zhang, Z., Chow, T. W., Li, B., 2014. A general soft label based Linear Discriminant Analysis for semi-supervised dimensionality reduction. Neural Networks 55, 83–97.

Zhou, S., Chen, Q., Wang, X., Li, X., 2014. Hybrid Deep Belief Networks for Semi-supervised Sentiment Classification. In: Proceedings of the 25th International Conference on Computational Linguistics. pp. 1341–1349.

Zhu, J., Wang, H., Mao, J., 2010. Sentiment classification using genetic algorithm and Conditional Random Fields. In: 2010 2nd IEEE International Conference on Information Management and Engineering. pp. 193–196.