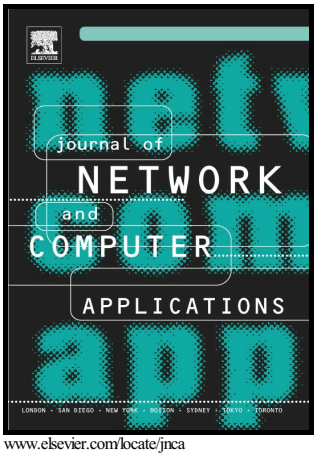
SentiRelated: a Cross-Domain Sentiment Classification Algorithm for Short Texts through Sentiment Related Index

Lei Wang, Jianwei Niu, Houbing Song, Mohammed Atiquzzaman



www.elsevier.com/locate/jnca

PII: S1084-8045(17)30358-2
DOI: https://doi.org/10.1016/j.jnca.2017.11.001
Reference: YJNCA2004

To appear in: *Journal of Network and Computer Applications*

Received date: 28 June 2017
Revised date: 10 September 2017
Accepted date: 3 November 2017

Cite this article as: Lei Wang, Jianwei Niu, Houbing Song and Mohammed Atiquzzaman, SentiRelated: a Cross-Domain Sentiment Classification Algorithm for Short Texts through Sentiment Related Index, *Journal of Network and Computer Applications,* https://doi.org/10.1016/j.jnca.2017.11.001

SentiRelated: a Cross-Domain Sentiment Classification Algorithm for Short Texts through Sentiment Related Index

Lei Wang[a], Jianwei Niu[a,*], Houbing Song[b], Mohammed Atiquzzaman[c]

[a]*State Key Laboratory of Software Development Environment, School of Computer Science and Engineering, Beihang University, Beijing 100191, China*
[b]*Department of Electrical and Computer Engineering, West Virginia University, WV 25136, USA*
[c]*School of Computer Science, University of Oklahoma, Norman, OK 73019, USA*

Abstract

Sentiment classification for short texts, aiming at predicting sentiment polarity of short texts automatically, has attracted more and more attentions due to its wide applications. Traditional supervised classification approaches perform well in predicting the sentiment polarity for a given domain, but the performance decreases drastically when a classifier trained on a specific domain is directly applied to predict the sentiment polarity of another domain because the words used in the trained domain may not appear in the test domain. Moreover, the same word may indicate different sentiment polarities in different domains. In this paper, to bridge the gap between different domains, we create a *Sentiment Related Index (SRI)* to measure the association between different lexical elements in a specific domain with the help of *domain-independent features* as a bridge. Then we propose a novel cross-domain sentiment classification algorithm based on SRI, which is termed *SentiRelated*, to analyze the sentiment polarity for short texts. SentiRelated utilizes SRI to expand feature vectors based on unlabeled data from the target domain. In this way, some important sentiment indicators for the target domain are appended to feature vectors. At last, we validate our SentiRelated algorithm on two typical datasets. The experimental results demonstrate that, compared with state-of-the-art algorithms, our SentiRelated algorithm can improve the per-

*Corresponding author

Email addresses: lei@buaa.edu.cn (Lei Wang), niujianwei@buaa.edu.cn (Jianwei Niu), Houbing.Song@mail.wvu.edu (Houbing Song), atiq@ou.edu (Mohammed Atiquzzaman)

formance of cross-domain sentiment classification for short texts.

Keywords: Sentiment classification, Cross-domain, Short texts

1. Introduction

Sentiment analysis of product or service reviews is becoming more and more important due to the rapid development of online shopping platforms, such as Amazon (*amazon.com*), eBay (*ebay.com*), Tmall (*tmall.com*), and Jingdong (*jd.com*). Nowadays a large number of customers share their reviews on the commodities they have bought through various platforms on the Internet, and people can refer to these reviews before their purchasing. Product reviews are useful for both consumers to make decisions, and sellers to understand public thinkings about their products. As a result, given the ever-growing amount of product reviews available from various platforms on the Internet, mining the sentiment polarity expressed in product reviews has become critical expectations from both end-users and researchers [1]. Sentiment analysis, targeting at classifying product reviews into polarity categories (e.g., positive or negative), has been applied in numerous scenarios, such as market analysis [2, 3], product recommendation [4] and review spam identification [5, 6].

Traditional supervised learning algorithms have been proved effective and widely used in predicting the sentiment polarity for given domains [7]. However, supervised learning algorithms require a large amount of manually labeled data, and it is time-costing and labor-consuming to accomplish the labeling work. Moreover, people trend to use some different words to express sentiment in different domains. Table 1 shows several reviews from two domains: *Hotels* and *Computers*. Some words used frequently in a domain may be hard to be found in another domain. For example, domain-specific words in the *Hotels* domain, such as "spacious" and "filthy", are hard to be found in reviews in the *Computers* domain. Furthermore, some words carrying a specific sentiment polarity in one domain may even indicate a different sentiment polarity in another domain. "Hot" that appears in "The computer is very hot when working" explicitly indicates a negative sentiment. However, when referring to "The shower had great hot water" in the *Hotels* domain, the same word "hot" expresses a positive sentiment. Therefore, the performance of a sentiment classifier trained on a specific domain will drop drastically if the classifier is directly applied to another domain. Thus, it is challenging and highly desirable to investigate cross-domain sentiment classification.

Table 1 Cross-domain sentiment classification examples: positive (+) and negative (−) reviews of *Hotels* and *Computers*

	Hotels	*Computers*
+	Everything was wonderful! The bed was very comfortable. The bathroom was spacious. The shower had great hot water. We had a great time!	This exceeded my expectations. The seller gave outstanding service, and it arrived earlier than expected. It was a economical way to provide a quality product to a child.
+	We enjoyed the Skyrise and the easy access to parking and adventuredome. The hotel is beautiful and the variety of shops were so convenient.	The CPU is powerful. It's fast and with Windows 8.1 updated to let us old codgers use it properly. I would definitely recommend this to anyone who needs a home computer.
−	The overall appearance of the hotel was not clean. The hallways, elevators and stairwells were filthy.	This computer is a piece of junk. I was looking for a good starter computer with normal internet access. The computer is very hot when working.

Currently there are some algorithms proposed to solve the cross-domain sentiment classification problem, such as Spectral Feature Alignment (SFA) [8], Structural Correspondence Learning (SCL) [9], TRIplex Transfer Learning (TriTL) [10] and graph-based algorithms [11, 12]. However, most of these algorithms are not dedicated to short texts. Product reviews from various online shopping platforms belong to short texts in general, and they are much shorter, sparser, and noisier, which demands for a revisit of many fundamental technical problems for cross-domain sentiment classification [13]. The performances of these algorithms will drop significantly when they are applied to these short product reviews.

In this paper, we aim at finding an effective algorithm for the cross-domain sentiment classification problem, especially for these short texts. In our proposed cross-domain sentiment classification algorithm called *SentiRelated*, the cross-domain sentiment classification problem is modeled as a *feature expansion* problem. As we all know, some *domain-independent words* are widely used to express the same sentiment in different domains. Therefore, we can use these domain-independent words to obtain some additional features from the target domain. Then these additional features from the target domain are appended to feature vectors extracted from the source domain. In this way, the gap between the source

and target domains is reduced. To obtain appropriate additional features from short texts, we create a *Sentiment Related Index (SRI)* to measure the association between different lexical elements in a specific domain. The traditional similarity measurement of sentiment between different lexical elements depends on the co-occurrence of the two words. Different from the traditional way, we use the distribution of word occurrence to execute the job. This makes the sentiment related index unbiased toward infrequent features and words. Based on sentiment related index, from massive unlabeled data on the target domain, we then append additional features to the feature vectors in a binary classifier. In our SentiRelated algorithm, we make full use of unlabeled data, which is much cheaper to collect compared with labeled data. Moreover, the proposed algorithm can learn from a great deal of unlabeled data and build a robust cross-domain sentiment classifier. The contributions of this paper are summarized as follows.

- We create a sentiment related index to measure the association between different lexical elements in a specific domain. The experimental results show that sentiment related index is unbiased toward infrequent features and words.
- We propose a method to expand feature vectors based on sentiment related index. Some additional features from the target domain are added to feature vectors, and these additional features can bridge the gap between the source and target domains.
- We conduct a series of experiments to evaluate the performance of our proposed SentiRelated algorithm. We experimentally test the sensitivity of the two parameters in our proposed algorithm. We also study the effectiveness of sentiment related index and using multiple source domains. Besides, we compare the accuracy of our proposed algorithm with several state-of-the-art cross-domain sentiment classification algorithms, and the experimental results demonstrate that our proposed algorithm is effective to analyze the sentiment polarity of short texts.

The rest of this paper is organized as follows. Section 2 introduces the related work in this field. Section 3 describes the dataset used in our experiment and problem description. We propose a cross-domain sentiment classification algorithm in Section 4, and evaluate it in Section 5. We conclude this paper in Section 6.

2. Related Work

Sentiment classification, aiming at predicting sentiment polarity of text data, has become an important field of opinion mining due to its wide applications. The majority of sentiment analysis approaches can fall into two groups: single-domain sentiment classification and cross-domain sentiment classification.

For the single-domain sentiment classification problem, a classifier trained on a specific domain is used to predict the sentiment polarity of the same domain. Many machine learning based approaches have been applied on single-domain sentiment classification, such as the supervised learning based approaches [14] [15], the semi-supervised learning based approaches [16], the deep learning based approaches [17]. For the sentiment classification for short texts, the authors of [18] proposed the Adaptive Recursive Neural Network (AdaRNN) and employed multiple composition functions. In [19], the authors modeled the sentiment classification problem as a learning sentiment-specific word embedding issue and applied neural networks to incorporate the supervision. In general, machine learning based sentiment classification algorithms rely on a corpus of data labeled with their sentiment polarity or sentiment strength. These learning algorithms report the best performance when there are sufficient labeled texts available. However, these approaches are domain dependent. Thus to analyze the sentiment of data from a new domain, the training data of the new domain need to be labeled, and the classifier needs to be rebuilt. It is a challenging issue to adapt a classifier trained on a specific domain to a different domain.

Compared with the single-domain sentiment classification problem, the cross-domain sentiment classification problem focuses on how to apply the labeled data of the source domain to predict the sentiment polarity of a different domain [20, 21]. To solve this problem, the authors of [9] proposed the SCL algorithm which constructs a set of related tasks in order to model the relationship between "pivot features" and "non-pivot features". However, it is quite hard to build a reasonable number of related tasks from the datasets in practice, and the transfer ability of SCL for cross-domain sentiment analysis is limited. Similar to SCL, Spectral feature alignment (SFA) [8] divides features into two groups: domain independent features (pivot features), and domain specific features (non-pivot features). Then a bipartite graph is constructed between the two groups, and spectral clustering is performed on this bipartite graph to create a lower dimensional representation. Finally, a binary logistic regression model is trained in this lower-dimensional space using the labeled data from the source domain. SFA only relies on sufficient labeled data from the source domain, but ignores

unlabeled data, which is much cheaper to collect compared with labeled data. In [22], the authors constructed a Sentiment Sensitive Thesaurus (SST) to measure the similarity of sentiment between two words, and expanded feature vectors with additional related elements to overcome the feature mismatch problem. The accuracy achieved by this algorithm outperformed the accuracies given by SCL and SFA. However, SST is not dedicated to short texts, and the performance of SST will drop when it is applied to predict the sentiment polarity of short texts. In the performance evaluation part of this paper, we will compare our proposed SentiRelated algorithm with SCL, SFA and SST.

3. Problem Description and Datasets

3.1. Problem Description

We define a domain D as a collection of entities that share similar features. For example, different types of products, such as computers, books, or foods, fall in different domains. Our objective is to predict the sentiment polarity expressed in the reviews about a product belonging to a specific domain. In this paper, we only focus on positive and negative sentiment reviews. It is not very hard to extend our proposed algorithm to classify other sentiment reviews, such as neutral sentiment reviews or mixed sentiment reviews.

Let D_{src} and D_{tar} denote the source and target domains respectively. $D^l_{src} = \{(x_{s_1}, y_{s_1}), (x_{s_2}, y_{s_2}), ..., (x_{s_m}, y_{s_m})\}$ is the set of labeled data from the source domain, where $x_{s_1}, x_{s_2}, ..., x_{s_m}$ are m reviews sampled from the source domain, and $y_{s_1}, y_{s_2}, ..., y_{s_m} \in \{+1, -1\}$ are sentiment labels. Here, the sentiment labels $+1$ and -1 denote positive and negative sentiment, respectively. In addition to the labeled instances from the source domain, there also exist some unlabeled instances from both the source and target domains. The set of unlabeled data from the source domain is denoted by D^u_{src}, and the set of unlabeled data from the target domain is denoted by D^u_{tar}. Our task is to learn a cross-domain sentiment binary classifier based on D^l_{src}, D^u_{src} and D^u_{tar} to make prediction on the sentiment label of a review from the target domain.

3.2. Experimental Datasets

In this section, we describe the datasets used in this paper. The first dataset is from [23] and [24], and it has been widely used to evaluate the effectiveness of cross-domain sentiment classification algorithms. It contains a collection of product reviews that consist of three different domains: Computer (Comp), Education (Edu), and Hotel (Hotl). Each review in the dataset is assigned a sentiment

label, $+1$ (positive review) or -1 (negative review), by the authors manually. This dataset is denoted by *RewData*. In this dataset, the average length of each review is larger than 120.

The second dataset is collected by ourself for experimental purpose. We crawled a set of reviews from *Douban*[1]. The reviews from *Douban* contain three domains: Movie (Mov), Music (Mus), and Book (Book). We invited 10 volunteers to manually assign a sentiment polarity label for each review. Let *DoubanData* denote this dataset. In this dataset, the average length of each review is smaller than 25.

The details of these two datasets are listed in Table 2. From Table 2, we can see that compared with the dataset of *RewData*, the average length of *DoubanData* is much shorter.

Moreover, we designed a crawler program to collect a set of unlabeled reviews on these six domains from *Dangdang*[2], *Gome*[3], *Ctrip*[4] and *Douban*. As a result, we collect 63971 unlabeled reviews on domain *Comp*, 84165 reviews on domain *Edu*, 73514 reviews on domain *Hotl*, 58165 reviews on domain *Mov*, 64761 reviews on domain *Mus*, and 82651 reviews on domain *Book*. When collecting reviews from these websites, we use the method similar to [25] to crawl and store crawled data.

All reviews in the datasets are written in Chinese. Therefore, we use *Jieba*[5], a Chinese text segmentation tool, to segment these Chinese reviews.

4. Cross-domain Sentiment Classification

As we can see from Table 1, a challenging problem for a cross-domain sentiment classifier is that the features extracted from the reviews in a specific domain may mismatch the features extracted from a different domain. In this section, we present how we overcome this problem in detail.

4.1. Basic Idea

Users trend to utilize some different words when they express their opinions in different domains, which leads to the gap between different domains. At the

[1] http://www.douban.com/
[2] http://www.dangdang.com
[3] http://www.gome.com.cn/
[4] http://www.ctrip.com/
[5] https://github.com/fxsjy/jieba

Table 2 The datasets we study

Dataset	Domain	Positive	Negative	Total	Average Length
RewData	Comp	$2,714$	$2,791$	$5,505$	121
	Edu	$2,015$	$1,713$	$3,748$	581
	Hotl	$2,395$	$3,518$	$5,913$	317
DoubanData	Mov	$8,316$	$4,701$	$13,017$	23
	Mus	$7,823$	$3,771$	$11,594$	18
	Book	$8,647$	$4,177$	$12,824$	16

same time, the same word may indicate different sentiment polarities in different domains. However, some *domain-independent words*, such as "good" and "bad", are widely used to express the same sentiment in different domains. Therefore, we use these domain-independent words to bridge the gap between different domains.

4.2. Domain-Independent Feature Selection

First we need to establish some strategies to select domain-independent features. As aforementioned, a domain-independent feature ought to be used frequently and indicate the same sentiment in both the source and target domains. In this section, we present how to identify whether a feature is domain-independent or not.

Firstly, we need to choose the features that express the same sentiment in both the source and target domains from all the features that appear in the reviews in both domains. In this step, our work relies on sentiment lexicons that contain a list of positive and negative words. Note that the sentiment of these words is domain-independent. Therefore, we remove the features that do not appear in sentiment lexicons, and the rest are considered as candidates of domain-independent features. Two sentiment lexicons, *HowNet*[6] and *NTUSD*[7], are chosen to generate our candidate features. We collect 8934 sentiment words (4566 positive words and 4368 negative words) from *HowNet* and 11086 sentiment words (2810 positive words and 8276 negative words) from *NTUSD*. After removing duplicates from

[6]http://www.datatang.com/data/12990

[7]http://www.datatang.com/data/44317

Table 3 Sentiment words of lexicons

Lexicon	Positive	Negative	Total
HowNet	$4,566$	$4,368$	$8,934$
NTUSD	$2,810$	$8,276$	$11,086$
OBL	$5,834$	$10,108$	$15,942$

HowNet and *NTUSD*, we obtain a Original Basic Lexicon (*OBL*), which consists of 5834 positive words and 10108 negative words. The details of these lexicons are displayed in Table 3.

Secondly, we select the features that used frequently in both domains from the candidates of domain-independent features. To be more specific, given the reviews from both the source and target domains, we choose the features that used more than K times in both domains. In practice, the number of domain-independent features, n, is known. Therefore the value of K can be set to be the largest number such that we can obtain at least n domain-independent features.

4.3. Sentiment Related Index

A fundamental problem for a cross-domain sentiment classifier is that the features used in the target domain may do not occur in the source domain. Furthermore, the same word may indicate different sentiment polarities in different domains. As mentioned above, the gap between different domains can be reduced by using domain-independent features. With the help of domain-independent features, we can select some appropriate additional related features about the target domain from a set of candidates of additional related features, and append these additional features to feature vectors. To obtain the set of candidates of additional related features for a target domain, we first represent each review from the target domain by using unigram and bigram features, and remove those unigrams and bigrams which contain the stopping words or domain-independent words. Then the unigrams and bigrams still remaining form the set of candidates of additional related features for the target domain. In this section, we present two strategies for feature expansion.

The first strategy is to expand features based on the co-occurrence frequency between a candidate of additional related feature (*candidate* in short for the rest of this paper) and a domain-independent feature. For example, in the reviews about books, if the word “well-written” often co-occurs with the word “good”, which

is a positive domain-independent feature, then the word "well-written" has a high probability to be a positive sentiment indicator for book reviews. Therefore, the word "well-written" can be expanded to the feature vector, thereby reducing the feature mismatch between different domains. In information theory, pointwise mutual information (PMI) is commonly used to measure the association between two different elements based upon the co-occurrence frequency of elements. However, product reviews are short texts in general. Compared with long texts, they are much shorter, sparser, and noisier, which results in the sparsity of feature vectors for product reviews. Unfortunately, recent studies show that pointwise mutual information is biased toward infrequent features and words [26].

To overcome the challenge of data sparsity for product reviews, here we propose a novel strategy for expanding features based on Sentiment Related Index (SRI). Similar to pointwise mutual information, sentiment related index is used to measure the association between different lexical elements (unigrams and bigrams) in a specific domain.

Given a set of product reviews C, a domain-independent feature s and a candidate t, let C_s and C_t denote the reviews which contain s and t in C, respectively. The sentiment related index (SRI) for t and s is calculated as follows,

$$SRI(s,t) = \frac{1}{\sum_{w \in V} dist(w,s,t)}, \tag{1}$$

$$dist(w,s,t) = \begin{cases} P(w|C_t) \cdot \log(\dfrac{P(w|C_t)}{P(w|(C_s \bigcup C_t))}) & \text{if } w \in V_{s,t} \\ 0 & otherwise, \end{cases} \tag{2}$$

where V is the vocabulary and $V_{s,t}$ is the set of words that occur in $C_s \bigcup C_t$. $P(w|C_t)$ means the probability that a randomly selected review from C_t contains word w. Note that the stopping words are removed from the vocabulary V. According to the definition, the value of sentiment related index is surely positive.

Intuitively, if a candidate t and a domain-independent feature s indicate different sentiments in a specific domain, then the set C_t expresses a positive or negative sentiment polarity indicated by a few words with relatively high probabilities of occurrences against their probabilities in the set C_s. Therefore, the value of $\sum_{w \in V} dist(w,s,t)$ should be large relatively. On the other hand, if t and s express the same sentiment in a specific domain, then the reviews matching t are analogous to random samples from the set of reviews matching s or t, with very similar distributions of word occurrences. As a result, the value of $\sum_{w \in V} dist(w,s,t)$

should be small relatively. The larger $SRI(s,t)$ is, the more likely that s and t can be treated as the same sentiment indicator for a given domain.

Compared with pointwise mutual information, sentiment related index considers the distributions of word occurrences instead of the co-occurrence frequency between different words, thus surmounting the challenge caused by infrequent features and words.

More recently, a lot of word embedding based methods have been proposed to measure the association between different lexical elements [27, 28]. However, these methods are not effective enough for sentiment classification, because they model the syntactic context of words but ignore the sentiment information of texts. As a result, they usually map words with similar syntactic context but opposite sentiment polarity, such as *good* and *bad*, to neighboring word vectors [19]. For our proposed sentiment related index, it can learn sentiment-specific words by considering the sentiment information of texts when measuring the association between different lexical elements.

4.4. Feature Expansion

In the previous section, we have presented how to compute sentiment related index for a candidate and a domain-independent feature. As mentioned in the previous section, enlarging a feature vector with additional related features based on sentiment related index can reduce the gap between different domains. In this section, we describe how to expand features in detail.

First, we model a labeled review from the source domain as a bag of words. For some reviews, they are expressed in a sarcastic way, such as "The computer works as fast as a turtle", which results in that a review contains domain-independent features with different sentiment polarities. To solve this problem, when representing a review as a feature vector, we only consider these domain-independent features which have the same sentiment polarity with the review. More specifically, a review d from the source domain is represented by a vector $\mathbf{d} = \{w_1, w_2, \ldots, w_M\}$, where each element w_i is either unigram or bigram that occurs both in d and the list of domain-independent features, and w_i has the same sentiment polarity with review d. To select the appropriate additional related features to expand the vector $\mathbf{d}$, for each candidate u, we compute a ranking score

$RankingScore(u, \mathbf{d})$. The ranking score is computed as follows:

$$RankingScore(u, \mathbf{d}) = \frac{\sum_{i=1}^{M} r_i \cdot SRI(w_i, u)}{\sum_{i=1}^{M} r_i}, \tag{3}$$

where $r_i = tf(w_i) \cdot tf(u)$, and $tf(w_i)$ and $tf(u)$ denote the number of times that terms w_i and u occur in the unlabeled reviews on target domain, respectively. Note that the value of $SRI(w_i, u)$ is normalized to [0,1] according to Min-Max normalization method in this step. According to the definition of $RankingScore$, given a review d, a candidate u will have a high ranking score if there are many features w_i that have high sentiment related index $SRI(w_i, u)$. We not only consider all the unigrams and bigrams that occur in the review d from the source domain, but also emphasize the salient features extracted from the reviews on the target domain. The more frequency w_i and u occur in the reviews on target domain, the more important they are for the target domain. Therefore, for each feature w_i, we weight the ranking score by r_i.

Next, for a review d, we compute the ranking scores for all the candidates and rank these candidates according to their ranking scores. In this way, we can obtain the top-N ranked candidates. Let f_d^k be the k-th $(1 \leq k \leq N)$ ranked candidate. Then we extend $\mathbf{d} = \{w_1, w_2, \ldots, w_M\}$ by the top-N ranked candidates $f_d^1, f_d^2, \ldots, f_d^N$, and acquire a new $M + N$ dimensions vector $\mathbf{d}' = \{w_1, w_2, \ldots, w_M, f_d^1, f_d^2, \ldots, f_d^N\}$. Compared with the original features extracted from the labeled reviews on the source domain, the expanded features should be lower in feature values. Therefore, the expanded features can be weighted according to their ranking scores. In practice, when selecting appropriate candidates for a given review to expand its feature vector, we can apply parallel programming frameworks (e.g., Spark MLlib [29] and Petuum [30]) to compute the ranking scores for different candidates in parallel. By taking advantage of parallelism, our model can be trained in a shorter time.

Based on the extended feature vectors, a binary cross-domain sentiment classifier is trained to predict the sentiment polarity of product reviews on the target domain. Many popular classifiers can be used for this purpose, such as Support Vector Machine (SVM) classifier, Native Bayes classifier, Decision Tree classifier, and so on. Different classifiers may have different performances in practice. Here we just adopt SVM classifier for experiments.

Algorithm 1 ***SRI based Cross-Domain Sentiment Classification*** Algorithm

Input: labeled source domain data D^l_{src}, unlabeled source domain data D^u_{src}, unlabeled target domain data D^u_{tar}, K (the candidate features which used more than K times in both domains are domain-independent features), N (the number of extended features when the feature vector is expanded).
Output: cross-domain sentiment classifier f.

Apply the criteria mentioned in Section 4.2 on D^l_{src}, D^u_{src} and D^u_{tar} to select the set $Features_{independent}$ of domain-independent features.
Obtain the set $Features_{candidate}$ of candidate additional related features based on D^u_{tar}.
for all $f_c \in Features_{candidate}$ **do**
  for all $f_i \in Features_{independent}$ **do**
    Compute $SRI(f_c, f_i)$ based on D^u_{tar} as mentioned in Equation 1 and Equation 2.
  end for
end for
for all $d \in D^l_{src}$ **do**
  for all $f_c \in Features_{candidate}$ **do**
    Compute $RankingScore(f_c, \mathbf{d})$ based on D^u_{tar} as mentioned in Equation 3.
  end for
end for
for all $d \in D^l_{src}$ **do**
  Extend $\mathbf{d}$ by top-N ranked candidate additional related features and get a new feature vector $\mathbf{d}'$.
end for
Return a classifier f, trained on expanded feature vectors.

The whole process of our proposed algorithm for cross-domain sentiment classification is presented in Algorithm 1, and we term our proposed algorithm *SentiRelated*. When applying our proposed algorithm into practical application scenarios, we can utilize the computational architecture proposed in [31] to implement our proposed algorithm. Sentiment analytics can be divided into offline and online parts. Offline analytics usually deal with huge datasets and have large CPU consumption models. Therefore, we can train classification model based on a large amount of data on the offline part. On the other hand, online analytics have

shorter deadlines. Thus, we can predict the sentiment polarity of a new review on the online part.

5. Performance Evaluation

In this section, we conduct a series of experiments to validity the effectiveness of our proposed SentiRelated algorithm for cross-domain sentiment classification.

5.1. Experimental Setup

In our experiment, we adopt SVM classifier to determine the sentiment polarity of a product review. We use *sklearn.svm*[8] with a linear kernel and all options are set by default. 80% of our datasets are used as training set for classifiers and the rest are regarded as test set for evaluation on the accuracy.

5.2. Parameter Sensitivity

In our proposed SentiRelated algorithm, there are two parameters, K and N. As we describe in Section 4.2, if a candidate occurs more than K times in both the source and target domains, the candidate feature can be considered as a domain-independent feature. N denotes the number of extended features for each review from the source domain when the feature vector is extended. In this section, we aim to identify good (K, N) so that the classifier can accurately predict unknown data.

To achieve this goal, we use a "grid-search" on K and N using cross-validation. We change K from 10 to 100 with a step length 10, and increase the value of N from 5 to 45 with a step length of 5. Then various pairs of (K, N) values are tried and the one with the best cross-validation accuracy (here we consider the average accuracy of all cross-domain sentiment classification tasks) is picked (see Figure 1). From Figure 1, we can see that we can set K as 40 and N as 20 in order to achieve the highest accuracy on the *RewData* dataset.

We conduct similar experiments on the *DoubanData* dataset. The highest accuracy is achieved when we set $K = 30$ and $N = 30$.

[8] http://scikit-learn.org

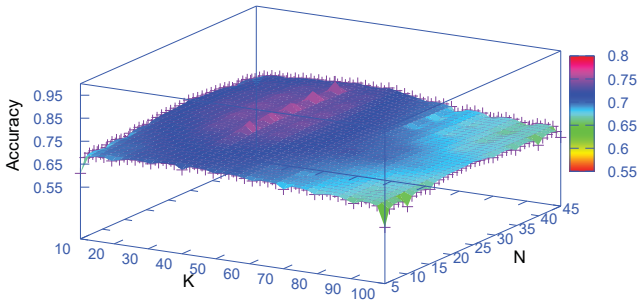


Fig. 1. Accuracy for different K and N on the *RewData* dataset.

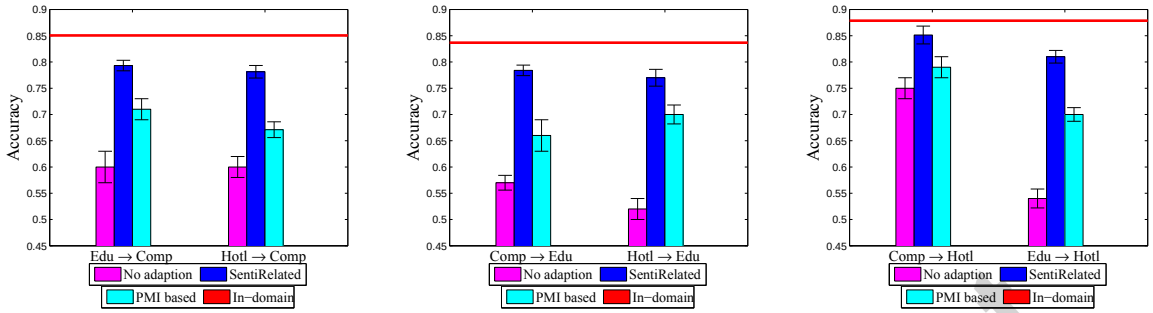
5.3. *Effectiveness of Sentiment Related Index*

As mentioned above, choosing appropriate features to expand feature vectors is a key challenge for cross-domain sentiment classification. In our proposed SentiRelated algorithm, Sentiment Related Index (SRI) is used to expand feature vectors when the cross-domain sentiment classifier is trained. To assess the advantage of applying sentiment related index to cross-domain sentiment classification, we compare the proposed SentiRelated algorithm against three baseline methods. The four baseline methods are listed as follows.

- *No adaption*. The sentiment classifier is trained on a specific domain and then it is directly used to predict the sentiment polarity of a review from a different domain. Here unigrams and bigrams occurring in each review are used as features to train the classifier. The experimental result of this method can be regarded as the lower bound for a cross-domain sentiment classifier.

- *SentiRelated*. This is our proposed algorithm in this paper. Additional features are appended to feature vectors according to Sentiment Related Index (SRI). The process of this method can be seen in Algorithm 1.

- *PMI based*. This baseline method is very similar to our proposed SentiRelated algorithm. The only difference is that sentiment related index is replaced by Pointwise Mutual Information (PMI) when enlarging feature vectors.



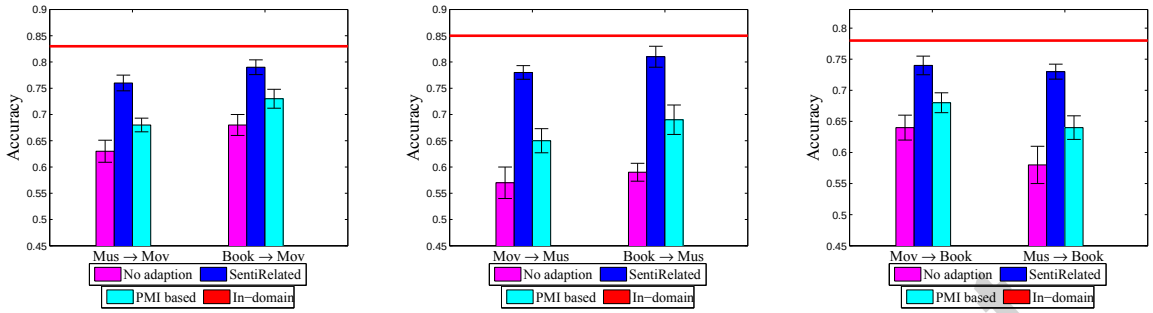
(a) *Comp domain* (b) *Edu domain* (c) *Hotl domain*

Fig. 2. The effectiveness of applying sentiment related index to cross-domain sentiment analysis on the *RewData* dataset.

- *In-domain*. In this method, the source domain and the target domain are the same domain. Similar to *No adaption*, unigrams and bigrams are used as features to train the classifier. The result of this method is the upper bound for cross-domain sentiment classifiers.

As we can see form Algorithm 1, in our proposed SentiRelated algorithm and the PMI based method, the values of K and N needed to be determined. According to the experimental results in the previous section, to achieve the highest accuracy, here we set $K = 40$ and $N = 20$ for the *RewData* dataset. At the same time, we set K as 30 and N as 30 for the *DoubanData* dataset. The classification accuracy of four methods mentioned above for the reviews from six different domains is shown in Figures 2 and 3.

From Figures 2 and 3, we can see that our SentiRelated algorithm achieves highest cross-domain sentiment classification accuracy for the datasets from six different domains. Note that the *in-domain* method is not the cross-domain setting, and its performance is treated as the upper bound for cross-domain sentiment classifiers. The PMI based method also obtains an obvious progressive performance compared with the *no adaption* method. The experimental results show that applying sentiment related index to expand feature vectors is very effective for cross-domain sentiment analysis, especially for the sentiment analysis of short texts. As aforementioned, sentiment related index, which considers the distributions of word occurrences instead of the co-occurrence frequency between different words, is unbiased toward infrequent features and words.



(a) *Mov domain* (b) *Mus domain* (c) *Book domain*

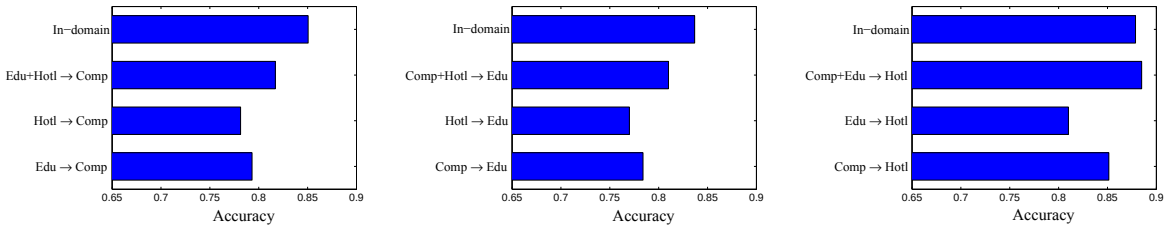
Fig. 3. The effectiveness of applying sentiment related index to cross-domain sentiment analysis on the *DoubanData* dataset.

5.4. Effectiveness of Using Multiple Sources

In real world, the labeled data is always from multiple domains when we analyze the cross-domain sentiment classification problem. Therefore, we can use reviews from multiple sources when the feature vectors are expanded. For a given target domain, it is a challenging issue to choose suitable source domains to adapt a sentiment classifier for the target domain. In this section, we study the effectiveness of using data from multiple source domains on our proposed algorithm.

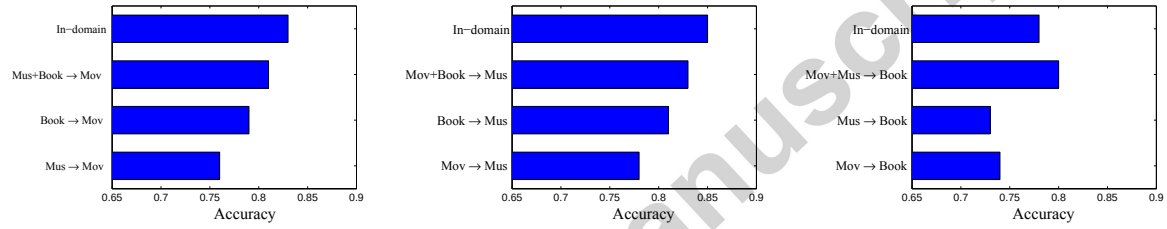
In this experiment, for the domain selected as the target domain, the sentiment classifier for it is trained using all possible combinations of the other domains. For example, for the *RewData* dataset, if we select domain *Comp* as the target domain, then will conduct three cross-domain sentiment classification experiments, i.e., *Edu* $\rightarrow$ *Comp*, *Hotl* $\rightarrow$ *Comp*, and *Edu* + *Hotl* $\rightarrow$ *Comp*. The effectiveness of using data from multiple source domains to build a sentiment classifier for a given target domain is presented in Figures 4 and 5.

As we can see from Figures 4 and 5, the performance of the sentiment classifier is improved in general after using data combining multiple source domains. Combination of multiple source domains means that the sentiment classifier can expand more features when the feature vectors are expanded compared with using those domains individually. Therefore, a higher accuracy is achieved when we use combination of multiple source domains. Moreover, we can observe a more interesting phenomenon. For the *Hotl* domain and *Book* domain, the accuracy when we use data from multiple source domains is higher than the accuracy of *in-domain* method, which is considered as the "upper bound" of cross-domain sentiment classifiers. This phenomenon can be explained by the fact that the *in-*



(a) *Comp domain* (b) *Edu domain* (c) *Hotl domain*

Fig. 4. The effectiveness of using multiple sources on the *RewData* dataset.



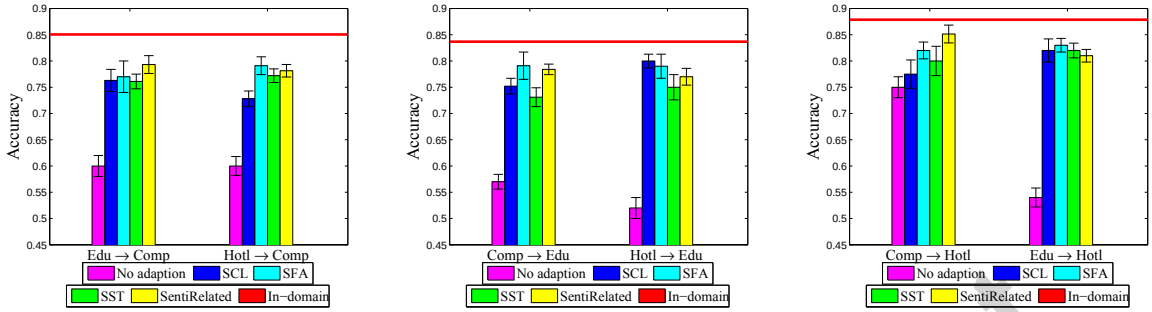
(a) *Mov domain* (b) *Mus domain* (c) *Book domain*

Fig. 5. The effectiveness of using multiple sources on the *DoubanData* dataset.

domain method only uses the labeled data from the target domain while our proposed algorithm can acquire some extra features from abundant unlabeled data on the target domain, which helps us to obtain a better accuracy.

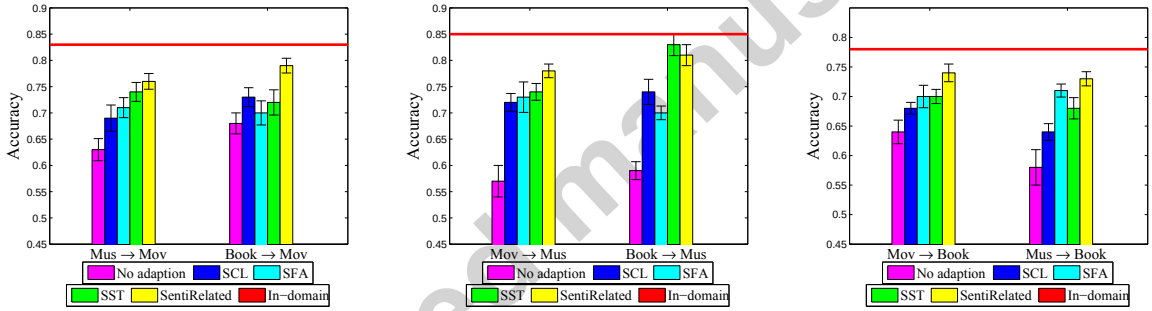
5.5. Comparison with Existing Work

In this section, we compare the performance of our proposed SentiRelated algorithm with three state-of-the-art cross-domain sentiment classification algorithms in 12 cross-domain sentiment classification tasks on two datasets. The three previously proposed algorithms include **SCL** [9], **SFA** [8] and **SST** [22]. Note that **SCL** and **SFA** apply data from a single source domain to build a sentiment classifier for a particular target domain. Therefore, firstly we conduct experiments where our SentiRelated algorithm and **SST** only use a single source domain to build a sentiment classifier. In this experiment, for the *RewData* dataset, the parameters K and N are set to 40 and 20 respectively. For the *DoubanData* dataset, we set $K = 30$ and $N = 30$. The comparison results of different algorithms are shown in Figures 6 and 7.



(a) *Comp domain* (b) *Edu domain* (c) *Hotl domain*

Fig. 6. Comparison against existing cross-domain sentiment classification algorithms on the *RewData* dataset.



(a) *Mov domain* (b) *Mus domain* (c) *Book domain*

Fig. 7. Comparison against existing cross-domain sentiment classification algorithms on the *DoubanData* dataset.

From Figures 6 and 7, we observe that all four algorithms achieve improved performances compared with *no adaption* method consistently. For the *RewData* dataset, among all 6 cross-domain sentiment classification tasks, **SFA** outperforms other algorithms in 3 tasks, our proposed SentiRelated algorithm outperforms in 2 tasks, and **SCL** achieves the highest accuracy in 1 task.

For the *DoubanData* dataset, the SentiRelated algorithm achieves the highest accuracy in 5 tasks and **SST** reports the best accuracy in 1 task. Note that the average length of the *DoubanData* dataset is much shorter than the *RewData* dataset. The experimental results demonstrate that SRI is unbiased toward infrequent features and words, and our SentiRelated algorithm can achieve obvious advantages

Table 4 Accuracy comparison against existing work on multiple source cross-domain sentiment classification

Method	$Comp$	Edu	$Hotl$	Mov	Mus	$Book$
No adaption	0.60	0.57	0.75	0.68	0.59	0.64
SCL	0.76	0.80	0.82	0.73	0.74	0.68
SFA	0.79	0.79	0.83	0.71	0.73	0.71
SST	**0.82**	0.78	0.85	0.77	**0.84**	0.72
SentiRelated	0.81	**0.80**	**0.89**	**0.81**	0.83	**0.80**
In-domain	0.85	0.84	0.88	0.83	0.85	0.78

Table 5 Recall comparison against existing work on multiple source cross-domain sentiment classification

Method	$Comp$	Edu	$Hotl$	Mov	Mus	$Book$
No adaption	0.58	0.63	0.69	0.70	0.66	0.62
SCL	0.80	**0.83**	0.79	0.79	0.87	0.75
SFA	0.83	0.72	0.83	0.80	0.81	0.79
SST	0.78	0.80	**0.88**	0.76	0.85	0.82
SentiRelated	**0.86**	0.79	0.85	**0.84**	**0.89**	**0.88**
In-domain	0.91	0.85	0.82	0.88	0.90	0.92

when analyzing the sentiment polarity of short texts.

As mentioned above, our proposed SentiRelated algorithm and **SST** can use data from multiple source domains. We apply SentiRelated algorithm and **SST** using multiple combined source domains. As illustrated in Tables 4 and 5, our proposed SentiRelated algorithm can achieve the highest accuracy and reall in 4 domains after combining multiple source domains in all cross-domain sentiment classification tasks, while **SST** achieves the highest accuracy in 2 tasks and the highest recall in 1 task. **SCL** can obtain the highest recall in domain *Edu*.

6. Conclusion

In this paper, we created a sentiment related index to measure the association between different lexical elements in a specific domain. Based on the sentiment related index, we proposed a novel algorithm, which is termed *SentiRelated*, where some additional features from the target domain are added to feature vectors extracted from the source domain. In this way, the gap between the source and target domains is narrowed. We validated our algorithm on two typical datasets and the experimental results showed our proposed SentiRelated algorithm is effective to analyze the sentiment polarity of short texts.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (61572060, 61772060), the foundation from State Key Laboratory of Software Development Environment (SKLSDE-2016ZX-23) and CERNET Innovation Project (NGII20151004, NGII20160316).

[1] K. Schouten, F. Frasincar, Survey on aspect-level sentiment analysis, IEEE Transactions on Knowledge and Data Engineering 28 (3) (2016) 813–830.

[2] C. Rohrdantz, M. C. Hao, U. Dayal, L.-E. Haug, D. A. Keim, Feature-based visual sentiment analysis of text document streams, ACM Transactions on Intelligent Systems and Technology (TIST) 3 (2) (2012) 26.

[3] J. Niu, L. Wang, Structural properties and generative model of non-giant connected components in social networks, Science China Information Sciences 59 (12) (2016) 123101.

[4] J. Niu, L. Wang, X. Liu, S. Yu, Fuir: Fusing user and item information to deal with data sparsity by using side information in recommendation systems, Journal of Network and Computer Applications 70 (2016) 41–50.

[5] N. Jindal, B. Liu, Opinion spam and analysis, in: Proceedings of the 2008 International Conference on Web Search and Data Mining, ACM, 2008, pp. 219–230.

[6] L. Wang, J. Niu, J. J. Rodrigues, Gma: An adult account identification algorithm on sina weibo using behavioral footprints, Future Generation Computer Systems.

[7] B. Pang, L. Lee, S. Vaithyanathan, Thumbs up?: sentiment classification using machine learning techniques, in: Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10, Association for Computational Linguistics, 2002, pp. 79–86.

[8] S. J. Pan, X. Ni, J.-T. Sun, Q. Yang, Z. Chen, Cross-domain sentiment classification via spectral feature alignment, in: Proceedings of the 19th international conference on World wide web, ACM, 2010, pp. 751–760.

[9] J. Blitzer, M. Dredze, F. Pereira, et al., Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification, in: ACL, Vol. 7, 2007, pp. 440–447.

[10] F. Zhuang, P. Luo, C. Du, Q. He, Z. Shi, H. Xiong, Triplex transfer learning: exploiting both shared and distinct concepts for text classification, Cybernetics, IEEE Transactions on 44 (7) (2014) 1191–1203.

[11] G. Xu, X. Meng, H. Wang, Build chinese emotion lexicons using a graph-based algorithm and multiple resources, in: Proceedings of the 23rd International Conference on Computational Linguistics, Association for Computational Linguistics, 2010, pp. 1209–1217.

[12] A. C.-R. Tsai, C.-E. Wu, R. T.-H. Tsai, J. Y.-j. Hsu, Building a concept-level sentiment dictionary based on commonsense knowledge, IEEE Intelligent Systems (2) (2013) 22–30.

[13] G. Paltoglou, M. Thelwall, Twitter, myspace, digg: Unsupervised sentiment analysis in social media, ACM Transactions on Intelligent Systems and Technology (TIST) 3 (4) (2012) 66.

[14] D.-T. Vo, Y. Zhang, Target-dependent twitter sentiment classification with rich automatic features, in: Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015), 2015, pp. 1347–1353.

[15] F. Li, S. Wang, S. Liu, M. Zhang, Suit: A supervised user-item based topic model for sentiment analysis, in: Twenty-Eighth AAAI Conference on Artificial Intelligence, 2014.

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

[17] R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng, C. Potts, Recursive deep models for semantic compositionality over a sentiment treebank, in: Proceedings of the conference on empirical methods in natural language processing (EMNLP), Vol. 1631, Citeseer, 2013, p. 1642.

[18] L. Dong, F. Wei, C. Tan, D. Tang, M. Zhou, K. Xu, Adaptive recursive neural network for target-dependent twitter sentiment classification, in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, 2014, pp. 49–54.

[19] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, B. Qin, Learning sentiment-specific word embedding for twitter sentiment classification, in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, Vol. 1, 2014, pp. 1555–1565.

[20] R. Xia, C. Zong, X. Hu, E. Cambria, Feature ensemble plus sample selection: domain adaptation for sentiment classification, Intelligent Systems, IEEE 28 (3) (2013) 10–18.

[21] D. Bollegala, T. Mu, J. Y. Goulermas, Cross-domain sentiment classification using sentiment sensitive embeddings, IEEE Transactions on Knowledge and Data Engineering 28 (2) (2016) 398–410.

[22] D. Bollegala, D. Weir, J. Carroll, Cross-domain sentiment classification using a sentiment sensitive thesaurus, IEEE Transactions on Knowledge and Data Engineering 25 (8) (2013) 1719–1731.

[23] S. Tan, G. Wu, H. Tang, X. Cheng, A novel scheme for domain-transfer problem in the context of sentiment analysis, in: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management, ACM, 2007, pp. 979–982.

[24] S. Tan, Y. Wang, X. Cheng, Combining learn-based and lexicon-based techniques for sentiment detection without using labeled examples, in: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '08, ACM, New York, NY, USA, 2008, pp. 743–744.

[25] M. Congosto, P. Basanta-Val, L. Sanchez-Fernandez, T-hoarder: A framework to process twitter data streams, Journal of Network and Computer Applications 83 (2017) 28–39.

[26] P. Pantel, D. Lin, Discovering word senses from text, in: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2002, pp. 613–619.

[27] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Advances in neural information processing systems, 2013, pp. 3111–3119.

[28] J. Pennington, R. Socher, C. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014, pp. 1532–1543.

[29] X. Meng, J. Bradley, B. Yavuz, E. Sparks, S. Venkataraman, D. Liu, J. Freeman, D. Tsai, M. Amde, S. Owen, et al., Mllib: Machine learning in apache spark, The Journal of Machine Learning Research 17 (1) (2016) 1235–1241.

[30] E. P. Xing, Q. Ho, W. Dai, J. K. Kim, J. Wei, S. Lee, X. Zheng, P. Xie, A. Kumar, Y. Yu, Petuum: A new platform for distributed machine learning on big data, IEEE Transactions on Big Data 1 (2) (2015) 49–67.

[31] P. Basanta-Val, N. C. Audsley, A. J. Wellings, I. Gray, N. Fernández-García, Architecting time-critical big-data systems, IEEE Transactions on Big Data 2 (4) (2016) 310–324.