Computers and Electrical Engineering 000 (2017) 1–13

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

# A novel sentiment aware dictionary for multi-domain sentiment classification[☆]

Vandana Jha[a,*], Savitha R[a], P Deepa Shenoy[a], Venugopal K R[a], Arun Kumar Sangaiah[b]

[a] *Department of Computer Science and Engineering, University Visvesvaraya College of Engineering, Bangalore University, Bangalore, India*
[b] *School of Computing Science and Engineering, VIT University, Vellore, Tamil Nadu, India*

ARTICLE INFO

*Article history:*
Received 19 July 2017
Revised 14 October 2017
Accepted 17 October 2017
Available online xxx

*Keywords:*
Sentiment analysis
Hindi language
Hindi Sentiwordnet
Multi-domain
Sentiment aware dictionary

ABSTRACT

Sentiment Analysis is a sub area of Natural Language Processing (NLP) which extracts user's opinion and classifies it according to its polarity. This task has many applications but it is domain dependent and a costly task to annotate the corpora in every possible domain of interest before training the classifier. We are making an attempt to solve this problem by creating a sentiment aware dictionary using multiple domain data. This dictionary is created using labeled data from the source domain and unlabeled data from both source and target domains. Next, this dictionary is used to classify the unlabeled reviews of the target domain. The work is carried out in Hindi, the official language of India. The web pages in Hindi language is booming after the introduction of UTF-8 encoding style. When compared with labeling done by Hindi Sentiwordnet (HSWN), a general lexicon for word polarity, the proposed method is able to label 23–24% more number of words of target domain. The labels assigned by our method and the labels given by HSWN, for the available words, are compared and found matching with 76% accuracy.

## 1. Introduction

Now a days, the opinions about movies, products or services are available in abundance on review sites, blogs and product sites. In products also, reviews are available for every type of products like kitchen appliances, books, DVDs, electronics etc.. Some of the always watched review sites are amazon.com, imdb.com, tripadvisor.com, caranddriver.com. These reviews are useful for both consumers and producers. The consumers can understand the performance of the product by reading other's views whereas the producers can get the information for improvement in the products or services. These advantages of reviews are the reason for the popularity of areas like opinion mining, opinion summarization, contextual advertising and market analysis. However, the words used to write reviews are different in different domains. For example, the words "energy saving" and "high quality" are used to write positive reviews about kitchen appliances, whereas "minimum in warranties" and "expensive" indicate negative reviews. In another way, the words "entertaining" and "enjoyable" are used to write positive reviews about DVDs, whereas "unfunny" and "boorish" indicate negative sentiments. It is expensive to train data in every new domain in which we want to test and classify the reviews. A supervised classifier trained in one domain

[☆] Reviews processed and recommended for publication to the Editor-in-Chief by Guest Editor Dr. Shuai Liu.
[*] Corresponding author.
*E-mail address:* vjvandanajha@gmail.com (V. Jha).

https://doi.org/10.1016/j.compeleceng.2017.10.015
0045-7906/© 2017 Elsevier Ltd. All rights reserved.

may not perform well in other domain test data because of the inability to learn unseen sentiment words. Hence, there is a need of sentiment aware dictionary from multiple domains to train the classifier for sentiment classification.

Sentiment classification is an important area of text classification whose goal is to classify a review based on the sentimental opinions conveyed by the reviewer in it. Sentiments can be classified into positive, negative, neutral or mixed category. A review with strong or more positive sentiment words in it, is treated as positive review whereas a review with strong or more negative sentiment words in it, is treated as negative review. A review with neither positive nor negative sentiment words is considered as neutral review whereas a review with both positive and negative sentiment words is considered as mixed review. Sentiment classification can be carried out at word level, sentence level or document level. Classifiers can be categorized, based on the domains in which they are trained and tested, into single-domain classifiers and multiple-domain classifiers. Single-domain classifiers are trained by the labeled data available in the domain and later tested on the same domain data whereas multiple-domain classifiers are trained by one or more domains, labeled or unlabeled data (source domains) and tested on another domain data (target domain). Our dictionary is useful for multiple-domain sentiment classification at document level. By creating a sentiment aware dictionary, the proposed method is able to label, unlabeled reviews from the target domain, into positive and negative classes with considerable accuracy. However, the proposed method can be easily extended to address multi-category sentiment classification problems.

### 1.1. Motivation

The multiple-domain sentiment classification is a challenging task and has recently received attention of the researchers. The main challenges involved are as follows:

1. It should be identified correctly that which features of the source domain are similar to which features of the target domain.
2. It should have a learning structure like a dictionary to accommodate the knowledge about the relatedness of the features from the source and target domains for the classification of target domain features.

Here, we are trying to overcome all these challenges by creating and using a multi-domain dictionary for labeling the target domain reviews into positive and negative classes. Our dictionary is in Hindi language. Hindi, the 4th largest language in the world, is the official language of India and has 310 million speakers across the world consisting of 4.46% of the world population.[1] Hindi content consumption on Internet is growing at whopping 94%.[2],[3],[4] But it is an uphill task for a resource scarce language like Hindi. Good Hindi language tagger and annotated corpus is not available. This problem is solved by using translation[5] of the reviews available in English language.

### 1.2. Contribution

In this paper, a fully automated Hindi Multi-Domain Sentiment Aware Dictionary, HMDSAD is proposed and used for the classification of unlabeled reviews of target domain. HMDSAD is created using labeled source domain data, unlabeled source domain data and unlabeled target domain data. It is based on the words those are co-occurring together in a review, also known as distributional context of the words. It keep, in multiple columns, different words from different domains, which express the same sentiment in the reviews.

A small part of our work describing HMDSAD creation is published in [1]. Here, we extend on that work in several ways:

1. HMDSAD dictionary is used to classify the unlabeled reviews from target domain into positive and negative classes. This can be done by labeling most of the words in the dictionary and using these labels for classification.
2. The dictionary is compared to HSWN, a general lexicon for word polarity, in which each sentiment associated word has a positive and a negative score. Our dictionary is able to label approximately 24% more number of words as compared to HSWN (shown in Table 10).
3. For the words available in HSWN, the label assigned by our methods is compared to the label given in HSWN. The achieved accuracy of available words with similar labeling is approximately 76% (shown in Table 11).

### 1.3. Resources

Hindi translation using translator[5] of the sentiment classification data set[6] for multiple-domain is used for our work. This is a benchmark data set, generated by Blitzer et al. [2]. It consists of product reviews from Amazon.com for four different product types: kitchen appliances, DVDs, electronics and books. The statistics of this data set is given in Table 1. The dataset contains user rating, review text and some other details. The reviews are labeled on the basis of user ratings. From now onwards, we refer this data set as review documents in this paper.

[1] http://en.wikipedia.org/wiki/List_of_languages_by_number_of_native_speakers.
[2] http://www.news18.com/news/tech/hindi-content-consumption-on-internet-growing-at-94-1-in-5-indian-users-prefer-hindi-google-1047247.html.
[3] http://www.internetworldstats.com/top20.htm.
[4] http://trak.in/tags/business/2015/08/19/hindi-content-content-consumption-growth-india-google/.
[5] https://translate.google.co.in/.
[6] https://www.cs.jhu.edu/~mdredze/datasets/sentiment/.

Please cite this article as: V. Jha et al., A novel sentiment aware dictionary for multi-domain sentiment classification, Computers and Electrical Engineering (2017), https://doi.org/10.1016/j.compeleceng.2017.10.015

**Table 1**
Statistics of Reviews in Review Documents.

| Domain | Positive | Negative | Unlabeled |
|---|---|---|---|
| kitchen appliances | 1000 | 1000 | 16,746 |
| DVDs | 1000 | 1000 | 34,377 |
| electronics | 1000 | 1000 | 13,116 |
| books | 1000 | 1000 | 5947 |

### *1.4. Organization*

The organization of the paper is as follows: A brief overview of the related work is given in Section 2. Section 3 describes dictionary creation using Pointwise Mutual Information (PMI) and relatedness weight calculation for multi-domain product reviews. Section 4 shows the employment of this dictionary in labeling and classification of target reviews. Simulation runs on product review data and the related results are discussed in Section 5. Comparison of our approach with HSWN is done in Section 6. Conclusions are given in Section 7.

## 2. Related Work

Sentiment Analysis problem can be divided into single-domain [3] and multiple-domain [4,5] based on the domains of data which are used to train the classifier and later to test the classifier.

### *2.1. Single-Domain Sentiment Analysis*

In single-domain sentiment analysis problem, a classifier receives training using labeled data from one domain and later tested with data from the same domain. Turney [3] used five patterns for calculating semantic orientation of reviews. The polarity of the words and phrases are measured by the words which are occurring together with a set of chosen positivity oriented words (e.g. excellent, good, nice etc.) and negativity oriented words (e.g. nasty, bad, poor etc.). This process used, a measure of association, Pointwise Mutual Information (PMI) to measure the sentimental orientation of a word. They achieved 84% accuracy on automobile review data and 66% on movie reviews. PMI method has been useful to weight features in many NLP tasks like word classification [6], word clustering [7] and similarity measurement [8]. The co-occurrence of the words, also known as its distributional context feature, is based on the assumption that words with comparable and similar distributions are semantically comparable and similar [9].

In Indian languages, works are comparatively lesser. Amitava Das and Bandopadhya [10] suggested a computational method for expanding SentiWordNet (Bengali) with the use of English-Bengali bilingual dictionary and English Sentiment Lexicons. They successfully got 35,805 Bengali words by applying lexical-transfer technique at word level to each word in English SentiWordNet using an English-Bengali dictionary to retrieve a Bengali SentiWordNet. Jha et al. [11,12] developed an opinion mining system in Hindi for Bollywood movie review data set. They achieved an overall accuracy of 87.1% for classifying positive and negative documents. Jha et al. [13,14] performed subjectivity analysis at the sentence level. They achieved 71.4% agreement with human annotators and approximately 80% accuracy in classifying a parallel data set in English and Hindi. Jha et al. [15] proposed a stopword removal algorithm for Hindi Language which is based on a Deterministic Finite Automata (DFA). They achieved 99% accurate results. Jha et al. [16] proposed a reputation system for evaluating trust among all good sellers of eBay website and able to rank the sellers efficiently.

### *2.2. Multiple-Domain Sentiment Analysis*

In multiple-domain sentiment analysis problem, a classifier is trained using labeled or unlabeled data from single or multiple domains and later tested with data from the different domains. Few interesting works are available in this area [4,17]. Blitzer et al. [17] proposed Structural Correspondence Learning (SCL) algorithm to train its multi-domain classifier. SCL method is built on the foundation of choosing a group of pivot features which gets repeated in both source and target domains. A linear predictor is trained to tell in advance the frequency of those pivot features and a binary sentiment classifier is trained using highlighted features of the lower dimensional matrix. Li et al. [4] proposed a multi-domain active learning framework. They identified that different domains have their own unique features, still they share some common latent features. A shared subspace is first learned to represent these common features of different domains. By considering duplicate information, they reduced the human labeling efforts in multi-domain learning.

We create a sentiment aware dictionary for multi-domain sentiment classification problem in Hindi language. However, to the best of our knowledge, multi-domain sentiment classification problem have not previously been dealt in Hindi language. One work is available in multilingual data [18] which uses Hindi language and Marathi language but it is also on single-domain data.

## 3. Dictionary Creation

As explained in the example of Section 1, the words used to express sentiments in different domains are different and when a classifier trained in one domain is applied to classify the reviews from another domain, words (features) mismatch problem occurs. This problem can be solved by creating a sentiment aware dictionary using multiple domain data. Algorithm 1 describes the overview of the dictionary creation task. It involves the extraction of review text from the review documents. Labeled and unlabeled reviews are considered from the source domain whereas target domain reviews are unlabeled. The labeled reviews have positive and negative labels assigned to them, based on user ratings, available with the reviews itself. Table 3 shows the various symbols used in this paper with its definitions.

The combination of source domain and target domain reviews are subjected to Part-Of-Speech (POS) tagging and lemmatization using hindi-pos-tagger.[7] This is a required step to filter sound clues of sentiments from the review documents. POS tagging is the process of assigning tags to each word in the sentence in review documents [19]. For example, suppose the sentence is, “यह पुस्तकअच्छा है” (*This book is good*), after POS tagging, the output is “यह/DEM पुस्तक/NN अच्छा/JJ है/VM” where DEM is Demonstrative, NN is Noun, JJ is Adjective and VM is Verb-finite.[8] Lemmatization is the process of removing inflectional endings and to return the base form of a word called lemma. Lemmatization is an effective method in text classification [20] as it reduces feature sparseness. Table 2 shows one positive and one negative review from two different domains: Kitchen Appliances and Electronics. A simple word filter is used to retain words those are nouns, verbs, adjectives and adverbs.[8] These are the sound clues of sentiments [21]. For each filtered list, unigrams list and bigrams list are generated. Next, for each labeled reviews of the source domain, sentiment awareness is created by appending label to each unigram in that review. For example, if a review is positive, all the unigrams are appended with “*P” and for negative reviews, “*N”, as shown in Table 4. Sentiment awareness are obtained only from labeled reviews of the source domain. We then compute the Pointwise Mutual Information (PMI), *f(k, z)* between a lexical or sentiment element *k* and feature *z* for each bigram as follows:

$$f(k,z) = log\left(\frac{\frac{c(k,z)}{N}}{\frac{\sum_{s=1}^{p} c(s,z)}{N} * \frac{\sum_{t=1}^{q} c(k,t)}{N}}\right)$$

Here, unigram/bigram is a lexical element, unigram appended with label is a sentiment element, feature is distributional context feature. The total number of reviews in which a lexical element *k* and a feature *z* co-occur is represented as *c(k, z)*, *p* and *q* are total number of *k* and *z* respectively and $N = \sum_{s=1}^{p} \sum_{t=1}^{q} c(s,t)$ (for detail, refer Eq. (1) from e-component supplementary material).

Second step is to calculate Relatedness Weight for the elements *k* and *z* as *r(z, k)*. It is required to calculate relatedness weight because even if PMI score of a bigram is high, the relatedness weight may increase or decrease depending on the overall review document file which consists of these bigrams and has specific co-occurrence factor. Relatedness weight explains the features of element *k* that it shares with element *z*. This weight is asymmetric as relatedness weight *r(z, k)* will not be equal to relatedness weight *r(k, z)* i.e. words that co-occur in one order need not co-occur in the reverse order. The formula for calculating relatedness weight is given in Algorithm 1 and further illustrated in the example using Eq. (2) in e-component supplementary material. Next step is dictionary creation. For each element *k*, we use the relatedness weight *r(z, k)* to list all the elements *z* that co-occur with element *k* (refer Algorithm 1). In this way, it aligns the words from different domains with similar sentiments.

Table 5 shows a sample of unigrams and bigrams with its frequency, PMI Score and relatedness weights. Frequency is calculated as the total number of occurrences of a feature in a review and is used to calculate PMI score and relatedness weight of each feature. Calculation of PMI score and relatedness weight is demonstrated by the example in e-component supplementary material. It is observed that PMI score is biased towards less occurring words. For less frequent bigrams, it is directly proportional to the relatedness weight, i.e., when PMI score is increasing, relatedness weight is also increasing and is shown in Fig. 1(a). For more frequent bigrams, it is inversely proportional to the relatedness weight, i.e., when PMI score is increasing, relatedness weight is decreasing and is shown in Fig. 1(b).

## 4. Labeling of Initial Dictionary and Classification of the Target Reviews using Labeled Dictionary

In this section, we describe application of the dictionary created in Section 3, for labeling and classification purposes.

### *4.1. Labeling*

After creating an initial HMDSAD dictionary, we have 1) words as base entries in the dictionary, named here as base_words, 2) RWC, count of the related words to the base_word and 3) related words list as Word1, Word2 and so on. These words are related to base_word based on its distributional context. A sample of this initial dictionary is shown in Table 6. In this dictionary, only about 50% words are labeled (from labeled source domain reviews), another 50% words are

[7] http://sivareddy.in/downloads#hindi_tools.
[8] http://ltrc.iiit.ac.in/nlptools2010/files/documents/POS-Tag-List.pdf.

**Algorithm 1:** Dictionary Creation

```
Data: Source Domain Review file, Target Domain Review file
Result: A file containing sentiment aware dictionary
begin
    Initialize:
    ReviewList[] = [Positive_Source, Negative_Source, Unlabeled_Source,
    Unlabeled_Target];
    Perform:
    for each j in ReviewList[] do
        Input = Input.append(ReviewList[j])
    end
    Tokenize and POS-tag Input
    Filter Pos-tagged Input with noun||adverb||adjective||verb tagging
    Create Unigram and Bigram (Unigram1 + Unigram2) list with its
    frequency
    for each Bigram do
        $PMI = log\frac{Bigram\_Frequency}{(Unigram1\_Frequency * Unigram2\_Frequency)}$
    end
    for each Bigram do
        $Relatedness\_weight(Unigram1, Unigram2) =$
        $\frac{\sum PMI(Unigram2, w1)}{\sum PMI(Unigram2, w2)}$
        where $w1$ is neighbour of Unigram1 satisfying
        $PMI(Unigram1, w1) > 0$
        where $w2$ is neighbour of Unigram2 satisfying
        $PMI(Unigram2, w2) > 0$
    end
    Apply sorting on the list of Unigrams, Bigrams, Relatedness_weight
    based on Unigrams, Relatedness_weight
    for each Bigram do
        Base_word = Unigram1
        RWC = Count of all Unigram2 with common Unigram1
        Related word = all Unigram2 with common Unigram1
    end
end
```

**Table 2**
Sample Reviews.

| | *Kitchen Appliances* | *Electronics* |
|---|---|---|
| **+ve** | मैं इन छुरियों से बहुत खुश हूँ, वे बहुत तेज हैं और देखने में भी सुन्दर हैं *(I'm very happy with these knives, they are very sharp and lookwise also beautiful)* | यह बच्चों के लिए बहुत ही सुरक्षित है, मेरी बेटी इस उत्पाद से खुश है *(It's very safe for children, my daughter is happy with this product)* |
| **-ve** | यह बेकार है, आसानी से टूट जाता है और अच्छा काम नहीं करता, पैसे बर्बाद नहीं करें *(It sucks, breaks easily and does not work well, do not waste money)* | मोनिटर मेरी सोच की तुलना से छोटी है, बहुत निराश *(The monitor is smaller than my thinking, very disappointed)* |

**Table 3**
Notations.

| *Symbol* | *Definition* |
|---|---|
| i | A base_word in HMDSAD dictionary |
| j | A review in the review file |
| k | A sentiment element in the review (unigram appended by label) |
| z | A feature in the review file |
| n | Number of words present in a review |
| N | Total number of words present in the review document file |
| x | All reviews in the review document file |
| y | Number of reviews in which base_word *i* is present |
| p | total number of sentiment elements *k* |
| q | total number of features *z* |
| r | relatedness weight |
| R | Pearson Correlation Coefficient |
| a and b | *AvgRankScore* by method 1 and *AvgRankScore* by method 2 respectively |

**Table 4**
Generating sentiment awareness (SA) from a negative review.

| | |
|---|---|
| Sentence | यह बेकार है, आसानी से टूट जाता है और अच्छा काम नहीं करता, पैसे बर्बाद नहीं करें *(It sucks, breaks easily and does not work well, do not waste money)* |
| Pos tags | यह/PRP बेकार/JJ है/VM ,/SYM आसानी/RB से/PSP टट/VM जाता/VAUX है/VAUX और/CC अच्छा/JJ काम/NN नहीं/NEG करता/VM ,/SYM पैसे/NN बर्बाद/JJ नहीं/NEG करें/VM |
| Unigrams | बेकार, आसानी, अच्छा, काम, पैसे, बर्बाद |
| Bigrams | बेकार+आसानी, आसानी+अच्छा, अच्छा+काम, काम+पैसे, पैसे+बर्बाद |
| SA | बेकार*N, आसानी*N, अच्छा*N, काम*N, पैसे*N, बर्बाद*N |

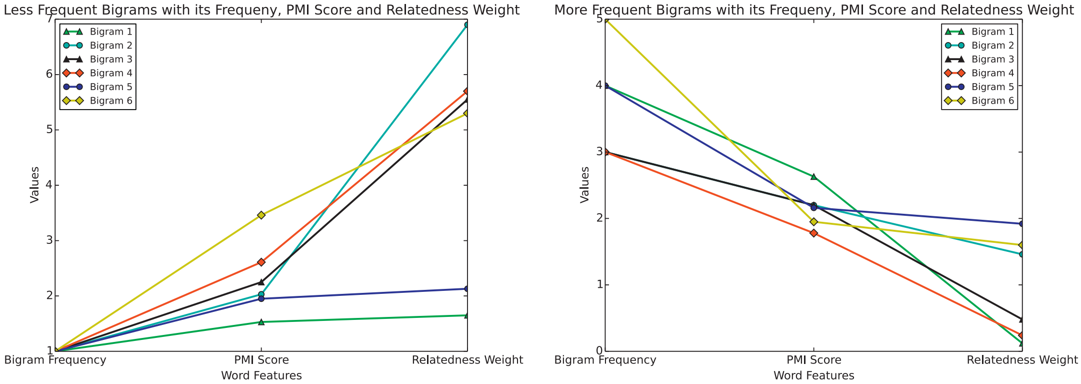
unlabeled (from unlabeled source and target domain reviews). In order to label as many words as possible, which will help later in accurate classification, average rank score, *AvgRankScore* is calculated for each base_word.

Algorithm 2 shows the steps for calculating *AvgRankScore*. For this calculation, we rank the base_words in dictionary by assigning a ranking score for each base_word. This ranking score is proportional to the frequency of the words present in the reviews. If a base_word is present in many reviews, it has high ranking score and if a base_word is not present in any review, its score is zero. $Rank[i, j]$, for base_word $i$ in review $j$ is computed as follows: Let us assume that there are $n$ unique words present in review $j$ with each has frequency [1,…,n] then $Rank[i, j]$ is calculated by finding the average of relatedness weights of base_word $i$ to all the words in review $j$, multiplied with the frequency of each word. If a base_word $i$ is not present in a review $j$ then its $Rank[i, j] = 0$ and it is neglected in average rank score calculation. If $Rank[i, j] > 0$ then it is considered for calculating average rank score for the base_word $i$. Let $x$ = all reviews in the review file and $y$ = number of reviews in which base_word $i$ is present then *AvgRankScore* for base_word $i = \frac{\sum_{j=1}^{x} Rank[i,j]}{y}$.

It is obvious from Algorithm 2 that *Rank* is dependent on relatedness weight but $r(i, k)$ and $r(k, i)$ (where $k$ is the word in the review $j$ of review file $x$) give different results so there are two methods for calculating *Rank* depending upon position of the base word in the bigram (formula given by Eqs. (1) and (2)) and thus have two different values for *AvgRankScore*.

**Table 5**
Sample of Tokens with its frequency (F), PMI Score (PScore) and Relatedness Weight (RWeight).

| Sl.No. | Unigram | F | Bigram | F | PScore | RWeight |
|---|---|---|---|---|---|---|
| 1 | उत्पाद<br>*(product)* | 60 | उत्पाद + खुश<br>*(product + happy)* | 4 | 1.14 | 0.0791094268 |
| 2 | खुश<br>*(happy)* | 30 | खुश + आराम<br>*(happy + ease)* | 2 | 1.78 | 0.0343878955 |
| 3 | सुन्दर<br>*(beautiful)* | 17 | सुन्दर + लायक<br>*(beautiful + able)* | 1 | 1.49 | 0.0827718961 |



(a) Bigram 1 = “अच्छा+आकार”
Bigram 2 =“असली+काम”
Bigram 3 = “अद्भुत+उपहार”
Bigram 4 =“अद्भुत+जानकारी”
Bigram 5 = “अच्छा*P+आकार*P”
Bigram 6 = “आकार*P+उज्जवल*P”

(b) Bigram 1 = “आराम+दायक”
Bigram 2 = “पूरी+तरह”
Bigram 3 = “लंबा+समय”
Bigram 4 =“समय+पसंदीदा”
Bigram 5 = “उत्कृष्ट+दस्तावेजी”
Bigram 6 = “अच्छी*P+तरह*P”

**Fig. 1.** Result of a sample of Bigrams with its frequency, PMI Score and Relatedness Weight.

**Table 6**
Sample of initial HMDSAD Dictionary.

| Base_word | RWC | Word1 | Word2 | Word3 | Word4 | Word5 | Word6 |
|---|---|---|---|---|---|---|---|
| आनंद | 2 | अलौकिक | उत्पाद | | | | |
| बढिया*P | 1 | संग्रह*P | | | | | |
| सुन्दर*P | 6 | लायक*P | सही*P | उत्पाद*P | शानदार*P | खुशबू*P | नया*P |
| स्वच्छ*P | 2 | सरल*P | ताजा*P | | | | |

### *4.1.1. Method 1*

$$Rank[i,j] = \frac{\sum_{k=1}^{n} relatedness(i,k) * frequency\_of\_k^{th}\_word}{n} \tag{1}$$

*AvgRankScore* for base_word, $i = \frac{\sum_{j=1}^{x} Rank[i,j]}{y}$

### *4.1.2. Method 2*

$$Rank[i,j] = \frac{\sum_{k=1}^{n} relatedness(k,i) * frequency\_of\_k^{th}\_word}{n} \tag{2}$$

*AvgRankScore* for base_word, $i = \frac{\sum_{j=1}^{x} Rank[i,j]}{y}$

Once the *AvgRankScore* for each base_word of initial HMDSAD dictionary is calculated (supporting examples are given in e-component supplementary material), the unlabeled base_words can be labeled by following steps:

**Algorithm 2:** Ranking Base_words of the dictionary

```
Data: Initial HMDSAD dictionary
Result: Dictionary with $AvgRankScore$
begin
    Perform:
    for each base_word i in dictionary do
        for each review j do
            $Rank[i, j] = 0$
        end
    end
    for each base_word i in dictionary do
        sum_of_scores=0, y=0
        for each review j in Review file do
            n=0
            for each word k in review j do
                n+=frequency[k]
                score[i,j]+=frequency[k] * relatedness weight between i and k
            end
            $Rank[i, j] = \frac{score[i, j]}{n}$
            if $Rank[i, j] > 0$ then
                y+=1
                sum_of_scores+=Rank[i,j]
            end
        end
        $AvgRankScore[i] = \frac{sum\_of\_scores}{y}$
    end
end
```

- If in the dictionary, same word is present in unlabeled and in one labeled form (either *P or *N), the label is appended to the unlabeled base_word.
- If in the dictionary, same word is present in unlabeled and in two labeled forms (both *P and *N), for both the labels *AvgRankScore* are considered and the label with higher *AvgRankScore* is appended to the unlabeled base_word, together with its *AvgRankScore* value. Considering higher *AvgRankScore* value is the best way to label words because *AvgRankScore* formula depends on relatedness weight, frequency of the base_word and the number of reviews in which this base_word is present so it is not neglecting any factor and thus not biased at all.

As base_words are less in number as compared to the number of related words, labeling only base_words are not sufficient and we also try to label, unlabeled related words of the initial dictionary as much as possible. This is done recursively. For each unlabeled base_word and unlabeled related words, the same words in labeled form are searched in first iteration and the labels are appended. Once the related words are labeled, in the next iteration, if same unlabeled base_word found, those are also labeled. The process terminates if no more labeling is possible for the base_words and the related words of the dictionary. The result of all this recursive labeling is final HMDSAD dictionary.

### 4.2. Classification

After final HMDSAD dictionary is created, next task is classification of unlabeled target reviews into either positive or negative class. This can be done by tokenizing, pos-tagging and filtering out nouns, verbs, adjectives and adverbs tokens from the reviews. Now these tokens are matched for each review and accordingly classified using three methods:

#### 4.2.1. Classification based on User Ratings

The basic method of classifying target reviews into positive and negative class is based on the user ratings. If the user rating is equal to or greater than 3, the review is classified as positive and if the user rating is less than 3, the review is classified as negative. Out of 100 target reviews, 59 reviews are classified as positive and 41 reviews as negative.

#### 4.2.2. Classification based on Sentiment Aware Dictionary

The filtered tokens of each review from target domain are matched with the unigrams available in final HMDSAD dictionary. If the match occurs, its label and *AvgRankScore* is considered. At the end of each review, we perform $\sum AvgRankScore(for\_positive\_labels) - \sum AvgRankScore(for\_negative\_labels)$. If the result of this calculation is greater than 0, the review is classified as positive; if less than 0, the review is classified as negative and if equal to 0, the review is classified as neutral. Out of 100 reviews from the target domain, by method 1, 42 reviews are classified as positive and 58 reviews as negative; by method 2, 31 reviews are classified as positive, 68 reviews as negative and 1 review as neutral.

#### 4.2.3. Classification based on HSWN

Hindi Sentiwordnet (HSWN) [22] is a general lexicon for word polarity in Hindi language. It contains following fields: POS tag, Synset ID (Hindi WordNet ID), Positive score, Negative score and Related terms (separated by comma). The filtered tokens of target reviews are matched with the words available in HSWN and for the matched words, available positive and negative scores are used to classify each target domain reviews into positive and negative classes in the same way as *AvgRankScore* values are used in the above step *4.2.2*. Out of 100 target reviews, 59 reviews are classified as positive, 33 reviews as negative and 8 as neutral.

## 5. Performance Evaluation

### 5.1. Datasets

Source domain reviews are a combination of reviews from all four domains, i.e., kitchen appliances, DVDs, electronics and books. We have randomly selected 200 labeled reviews and 200 unlabeled reviews from the source domain. Out of 200 labeled reviews, 100 positive reviews (25 reviews from each domain) and 100 negative reviews (25 reviews from each domain) are selected. Out of 200 unlabeled reviews, 50 reviews from each domain are selected. We have also randomly selected 100 unlabeled reviews from the target domain i.e. kitchen appliances. Thus the total number of reviews used for the creation of dictionary are 500 reviews. Table 2 displays one positive and one negative review from two different domains as samples. The review documents contain reviewer's name, product name, user rating, review text and other details. User ratings range between 0–5 stars. A rating greater than or equal to 3 is considered positive and less than 3 is considered negative. The reviews are classified on the basis of ratings and extracted only the review text sentences from each review documents.

### 5.2. Evaluation Results

Table 6 shows a sample of initial HMDSAD Dictionary. This dictionary is obtained, after sorting the results of PMI score and relatedness weights. First sorting is applied on the basis of unigrams so that all unigrams, with different combination of bigrams occur together in the results. Second sorting is applied on the basis of relatedness weight. After this, for each unigram, we have all the related words. In Table 6, base_word is shown with its RWC, i.e., the count of the words which are related to the base_word according to its distributional context. The words which are related are also shown in the Table 6 as word1, word2 and so on according to its count value.

Table 7 displays the values for *AvgRankScore* of base_words by using two methods as explained in Section 4 for labeling of base_words. It is obvious from Table 7 that *AvgRankScore* values calculated by two methods are different and the scatterplot of Fig. 2. further supports this difference. This scatterplot is plotted for *AvgRankScore* values by method 1 against *AvgRankScore* values by method 2 for 327 base_words. Fig. 2 shows that *AvgRankScore* by method 1 is weakly positive correlated with *AvgRankScore* by method 2. We have calculated the Pearson Correlation Coefficient, *R*, between method 1 and method 2 using the formula:

**Table 7**
*AvgRankScore* of Base_words using Method 1 and Method 2.

| Base_word | *AvgRankScore* by Method 1 | *AvgRankScore* by Method 2 |
|---|---|---|
| आदेश | 0.0087991291 | 0.0343900825 |
| आनंद | 0.0008594994 | 0.0160950161 |
| आराम | 0.0013540946 | 0.0077667751 |
| आश्चर्यजनक | 0.0006809935 | 0.0647015097 |
| इस्तेमाल | 0.0461034061 | 0.056402198 |

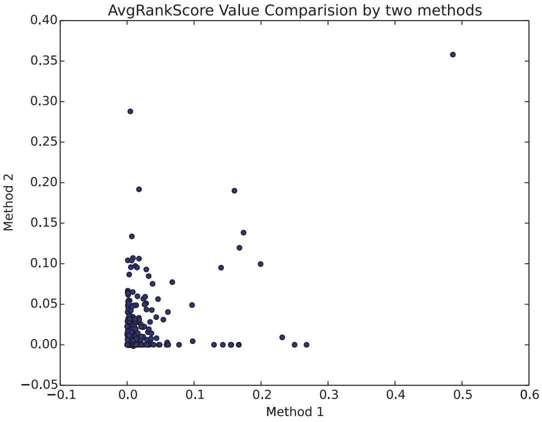


**Fig. 2.** Result of Comparison between *AvgRankScore* value calculated by two methods.

**Table 8**
Final HMDSAD Dictionary created by Method 1.

| Base_word | *AvgRankScore* | RWC | Word1 | Word2 | Word3 | Word4 | Word5 |
|---|---|---|---|---|---|---|---|
| कीमत *P | 0.002340198 | 5 | आसान *P | तेजी *P | महान *P | लायक *P | हाथ *P |
| सुन्दर *P | 0.011687297 | 4 | उत्पाद *P | लायक *P | शानदार *P | सही *P | |
| वापसी *N | 0.028792913 | 2 | तस्वीर *N | पता *N | | | |
| खिलौना *P | 0.043771044 | 4 | अच्छा *P | परीक्षण *P | समय *P | सही *P | |
| बुरा *N | 0.160165417 | 2 | खराब *N | गणवत्ता *N | | | |

**Table 9**
Final HMDSAD Dictionary created by Method 2.

| Base_word | *AvgRankScore* | RWC | Word1 | Word2 | Word3 | Word4 | Word5 |
|---|---|---|---|---|---|---|---|
| कीमत *P | 0 | 1 | महान *P | | | | |
| सन्दर *P | 0.0315613 | 2 | उत्पाद *P | लायक *P | | | |
| वापसी *N | 0.0435101 | 2 | तस्वीर *N | पता *N | | | |
| खिलौना *P | 0.0081099 | 4 | अच्छा *P | परीक्षण *P | समय *P | सही *P | |
| बुरा *N | 0.1901068 | 2 | खराब *N | गुणवत्ता *N | | | |

$R = \frac{\sum ab - \frac{(\sum a)(\sum b)}{327}}{\sqrt{(\sum a^2 - \frac{(\sum a)^2}{327})(\sum b^2 - \frac{(\sum b)^2}{327})}}$, where $a$ and $b$ represents *AvgRankScore* values by method 1 and 2 respectively.

The value of $R$ is 0.2251834, justifying weak positive correlation between method 1 and method 2. Based on the values of *AvgRankScore*, the unlabeled words are labeled. After the completion of the labeling task, final HMDSAD dictionary using method 1 and method 2 are displayed by Tables 8 and 9 respectively.

Next the target reviews are classified according to the labels of the base_words and their *AvgRankScore*. These results are compared with Hindi Sentiwordnet (HSWN) and shown in Table 12.

Please cite this article as: V. Jha et al., A novel sentiment aware dictionary for multi-domain sentiment classification, Computers and Electrical Engineering (2017), https://doi.org/10.1016/j.compeleceng.2017.10.015

**Table 10**
Labeling Percentage Comparision.

| Approach | No. of words in target review | No. of labeled words | No. of unlabeled words | Labeled% |
|---|---|---|---|---|
| Method 1 | 3291 | 1850 | 1441 | 56.213 |
| Method 2 | 3291 | 1814 | 1477 | 55.120 |
| HSWN | 3291 | 1052 | 2239 | 31.965 |

**Table 11**
Matching of Labels between Our Approach and HSWN.

| Approach | Total Target Review Words found in HSWN | Target Review Words Match with HSWN | Target Review Labels Match with HSWN | Label Matching% Accuracy% |
|---|---|---|---|---|
| Method 1 | 470 | 379 | 256 | 67.5 |
| Method 2 | 470 | 311 | 237 | 76.2 |

**Table 12**
Classification of Target Domain Reviews.

| Approach | Total Reviews | Positive Reviews | Negative Reviews | Neutral Reviews |
|---|---|---|---|---|
| User Rating | 100 | 59 | 41 | 0 |
| Method 1 | 100 | 42 | 58 | 0 |
| Method 2 | 100 | 31 | 68 | 1 |
| HSWN | 100 | 59 | 33 | 8 |

## 6. Comparision with Hindi Sentiwordnet (HSWN)

The comparison between our approach and HSWN is performed at three levels as described below.

### *6.1. Number of Labeled Words from Target Domain*

HSWN is a general lexicon and does not contain all the words from the target domain review. General lexicons are non-adaptable to domain-specific lexicon [17,23]. For the words available in HSWN, positive and negative scores are used for labeling the words. The number of words which can be labeled by HSWN and by our approach are shown in Table 10 and our approach, both by method 1 and method 2, are able to label more number of words as compared to HSWN.

### *6.2. Matching of Labels*

At this level, we are comparing the labels assigned by our approach and labels assigned by HSWN. In our approach, label is assigned based on *AvgRankScore* value and in HSWN, label is assigned on the basis of positive and negative score. The labels which are matched and their accuracy of labeling, considering label assigned by HSWN is correct, are given in Table 11.

### *6.3. Classification of Target Domain Reviews*

Target domain reviews can be classified by using four approaches. First approach is by ratings given by user; second, by using labels assigned by method 1; third, by using labels assigned by method 2; and fourth, by using positive and negative score assigned by HSWN. The results are given in Table 12 for the comparative analysis of these four approaches. User rating only, is not a reliable source for review classification [24]. Even in HSWN, some words like छोटा (*small*) is assigned negative score but in case of some products, "**small and handy**" is used in positive sentiment. Among the two methods proposed by us, method 1 outperforms method 2 in similar classification of target reviews.

## 7. Conclusion

We introduce an innovative way to find the relatedness of large words dataset taken from multiple source domains and a target domain which is used to build a multiple-domain sentiment aware dictionary for classifying unknown target domain reviews as positive or negative. This is required because getting unlabeled data in any domain is cheaper than getting annotated data in that domain. Most of the supervised learning algorithms for classification are using labeled data which are already existing in that domain for training but this may not be always possible. Our dictionary is useful in those situations. The algorithm used in our method is robust and assigns weight to each base_word of the dictionary. Basically the

Please cite this article as: V. Jha et al., A novel sentiment aware dictionary for multi-domain sentiment classification, Computers and Electrical Engineering (2017), https://doi.org/10.1016/j.compeleceng.2017.10.015

dictionary consists of Hindi nouns, adjectives, verbs and adverbs organized into a set of words representing the frequently occurring words with high weights. We have used this dictionary to classify target domain product reviews and compared the results at different levels with HSWN.

Our work can be extended in future by the processing of some same words used in different domains but mean different. For example, the word “bank” used as financial institution as well as used as river bank. The word is same but the domains are different and accordingly the meaning is different.

## Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.compeleceng.2017.10.015.

## References

[1] Jha V, Savitha R, Hebbar SS, Shenoy PD, Venugopal K. Hmdsad: Hindi multi-domain sentiment aware dictionary. In: Computing and network communications (CoCoNet), 2015 international conference on. IEEE; 2015. p. 241–7.
[2] Blitzer J, Dredze M, Pereira F. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In: Proceedings of the 45th annual meeting of the association of computational linguistics; 2007. p. 440–7.
[3] Turney PD. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In: Proceedings of the 40th annual meeting on association for computational linguistics. Association for computational linguistics; 2002. p. 417–24.
[4] Li L, Jin X, Pan SJ, Sun J-T. Multi-domain active learning for text classification. In: Proceedings of the 18th ACM SIGKDD international conference on knowledge discovery and data mining. ACM; 2012. p. 1086–94.
[5] Xia R, Zong C, Hu X, Cambria E. Feature ensemble plus sample selection: domain adaptation for sentiment classification. Intell Syst IEEE 2013;28(3):10–18.
[6] Bollegala D, Weir D, Carroll J. Cross-domain sentiment classification using a sentiment sensitive thesaurus. Knowl Data Eng IEEE Trans 2013;25(8):1719–31.
[7] Lin D. Automatic retrieval and clustering of similar words. In: Proceedings of the 36th annual meeting of the association for computational linguistics and 17th international conference on computational linguistics-Volume 2. Association for Computational Linguistics; 1998. p. 768–74.
[8] Saleh MR, Martín-Valdivia MT, Montejo-Ráez A, Ureña-López L. Experiments with svm to classify opinions in different domains. Expert Syst Appl 2011;38(12):14799–804.
[9] Harris ZS. Distributional structure. Word 1954;10(2–3):146–62.
[10] Das A, Bandyopadhyay S. Sentiwordnet for bangla. Knowl. Sharing Event-4 2010;2:1–8.
[11] Jha V, Manjunath N, Shenoy PD, Venugopal K, Patnaik LM. Homs: Hindi opinion mining system. In: Recent trends in information systems (ReTIS), 2015 IEEE 2nd international conference on. IEEE; 2015. p. 366–71.
[12] Jha V, Manjunath N, Shenoy PD, Venugopal K. Sentiment analysis in a resource scarce language: hindi. Int J Sci Eng Res 2016;7(9):968–80.
[13] Jha V, Manjunath N, Shenoy PD, Venugopal K. Hsas: Hindi subjectivity analysis system. In: India conference (INDICON), 2015 annual IEEE. IEEE; 2015. p. 1–6.
[14] Jha V, Shreedevi G, Shenoy PD, Venugopal K. Generating multilingual subjectivity resources using english language. Int J Comput Appl 2016;152(9):41–7.
[15] Jha V, Manjunath N, Shenoy PD, Venugopal K. Hsra: Hindi stopword removal algorithm. In: Microelectronics, computing and communications (MicroCom), 2016 international conference on. IEEE; 2016. p. 1–5.
[16] Jha V, Savitha R, Shenoy PD, Venugopal K. Reputation system: Evaluating reputation among all good sellers. In: Proceedings of NAACL-HLT; 2016. p. 115–21.
[17] Blitzer J, McDonald R, Pereira F. Domain adaptation with structural correspondence learning. In: Proceedings of the 2006 conference on empirical methods in natural language processing. Association for Computational Linguistics; 2006. p. 120–8.
[18] Balamurali A. Cross-lingual sentiment analysis for indian languages using linked wordnets. In: Proceedings of COLING 2012: Posters; 2012. p. 73–82.
[19] Manning CD. Part-of-speech tagging from 97% to 100%: is it time for some linguistics?. In: Computational linguistics and intelligent text processing. Springer; 2011. p. 171–89.
[20] Joachims T. Text categorization with support vector machines: Learning with many relevant features. In: European conference on machine learning. Springer; 1998. p. 137–42.
[21] Wiebe J. Learning subjective adjectives from corpora. In: AAAI/IAAI; 2000. p. 735–40.
[22] Das A, Bandyopadhyay S. Sentiwordnet for indian languages. In: Asian federation for natural language processing, China; 2010. p. 56–63.
[23] Denecke K. Are sentiwordnet scores suited for multi-domain sentiment classification?. In: Digital information management, 2009. ICDIM 2009. Fourth international conference on. IEEE; 2009. p. 1–6.
[24] O’Donovan J, Smyth B, Evrim V, McLeod D. Extracting and visualizing trust relationships from online auction feedback comments. In: IJCAI; 2007. p. 2826–31.

**Vandana Jha** obtained her Bachelor of Engineering in Computer Science and Engineering in 2003 and her Masters of Technology in 2009. Currently she is working as Research Scholar in the Department of Computer Science and Engineering, University Visvesvaraya College of Engineering, Bangalore University, Bangalore, India. Her research interests include Information Retrieval, Data Mining, Opinion Mining and Web Mining.

**Savitha R** obtained her Bachelor of Engineering in Information Science and Engineering from Visvesvaraya Technological University. She is Head of the department of Information Science and Engineering, Government polytechnic, Kalaburagi, Karnataka.

**P Deepa Shenoy** is currently a Professor in Computer Science and Engineering, University Visvesvaraya College of Engineering, Bangalore University, Bangalore, India. She did her doctorate in the area of Data Mining from Bangalore University. Her areas of research include Data Mining, Soft Computing and Social Media Analysis. She has published more than 150 papers in refereed International Conferences and Journals.

**K R Venugopal** is currently the Principal, University Visvesvaraya College of Engineering, Bangalore University, Bangalore. He was awarded Ph.D. in Computer Science from Indian Institute of Technology, Madras. He has authored and edited 70 books and has 500 research papers to his credit. His research interests include Computer Networks, Wireless Sensor Networks, Digital Signal Processing and Data Mining.

**Arun Kumar Sangaiah** has received his Ph.D. in Computer Science and Engineering from VIT University, Vellore, India. He is currently an Associate Professor in VIT University. He has authored more than 100 publications in different journals and conference of national and international repute. His current research work includes global software development, wireless ad hoc and sensor networks, machine learning etc.

Please cite this article as: V. Jha et al., A novel sentiment aware dictionary for multi-domain sentiment classification, Computers and Electrical Engineering (2017), https://doi.org/10.1016/j.compeleceng.2017.10.015