HMDSAD: Hindi Multi-Domain Sentiment Aware Dictionary

Vandana Jha*, Savitha R.*, Sudhashri S Hebbar*, P Deepa Shenoy* and Venugopal K R*
*Department of Computer Science and Engineering
University Visvesvaraya College of Engineering, Bangalore University, Bangalore, India
Email: vjvandanajha@gmail.com

***Abstract*—Sentiment Analysis is a fast growing sub area of Natural Language Processing which extracts user's opinion and classify it according to its polarity into positive, negative or neutral classes. This task of classification is required for many purposes like opinion mining, opinion summarization, contextual advertising and market analysis but it is domain dependent. The words used to convey sentiments in one domain is different from the words used to express sentiments in other domain and it is a costly task to annotate the corpora in every possible domain of interest before training the classifier for the classification. We are making an attempt to solve this problem by creating a sentiment aware dictionary using multiple domain data. The source domain data is labeled into positive and negative classes at the document level and the target domain data is unlabeled. The dictionary is created using both source and target domain data. The words used to express positive or negative sentiments in labeled data has relatedness weights assigned to it which signifies its co-occurrence frequency with the words expressing the similar sentiments in target domain. This work is carried out in Hindi, the official language of India. The web pages in Hindi language is booming very quickly after the introduction of UTF-8 encoding style. The dictionary can be used to classify the unlabeled data in the target domain by training a classifier.**

***Index Terms*—Domain Adaptation, Hindi Language, Hindi SentiwordNet, Multi-Domain, Natural Language Processing**

I. 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 [1], opinion summarization [2], contextual advertising [3] and market analysis [4]. 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 review about kitchen appliances, whereas "minimum in warranties" and "expensive" indicates negative review. In another way, the words "entertaining" and "enjoyable" are used to write positive review about movies, whereas "unfunny" and "boorish" indicates negative sentiment. 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 may not perform well on other domain test data because of inability to learn unseen sentiment words. Hence, there is a need for 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 [5], [6], sentence level [7] or document level [8], [9]. Classifiers can be categorized, based on the domains in which they are trained and tested, into single-domain classifiers [8], [9] and multiple-domain classifiers [10], [11]. 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.

A. 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 dictionary to accommodate the knowledge about the relatedness of the features from the source and target domains.

In this paper, we are trying to overcome both these challenges by creating a multi-domain dictionary. Our dictionary is in Hindi language. Hindi, the official language of India, is the 4th largest spoken language and has 490 million speakers across the world, which is 4.7% of the world population [12].

978-1-4673-7309-8/15/$31.00 ©2015 IEEE

Only 28.6 percent of Internet users understand English [13] so it is important to focus on other languages. 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[1] and then manual correction of the reviews available in English language.

B. Contribution

In this paper, a fully automated Sentiment Aware dictionary, HMDSAD is proposed. It is created using labeled source domain data, unlabeled source domain data and unlabeled target domain data. It is based on the words which are co-occurring together in a review, also known as distributional context of the words. It keeps parallely different words from different domains which express the same sentiment in the reviews. For this task, Pointwise Mutual Information (PMI) is calculated among the words and its relatedness to other words are measured. The related words are grouped together to generate a sentiment aware dictionary. The dictionary grabs the relatedness between words of source and target domains based on their distributional context and sentiment labels are extended to the words, wherever possible (available from labeled source domain data), which provides the sentiment awareness to the distributional dictionary.

C. Resources

Hindi translation using translator[1] of the sentiment classification data set[2] for multiple-domain is used for our work. This is generated by Blitzer et al. [10]. It is a benchmark data set and has been employed in many works on multiple-domain sentiment classification. 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 I. From now onwards, we refers this data set as review documents in this paper.

TABLE I: Statistics of Reviews in Review Documents

Domain	Positive	Negative	Unlabeled
kitchen appliances	1000	1000	16746
DVDs	1000	1000	34377
electronics	1000	1000	13116
books	1000	1000	5947

D. Organization

The organization of the paper is as follows: A brief overview of the related work is provided in section II. Section III describes dictionary creation using PMI calculation and relatedness weights calculation for Hindi language product reviews. Simulation runs on product review data and all the related results are discussed in section IV. Conclusions of the paper are given in Section V.

[1]https://translate.google.co.in/

[2]http://www.cs.jhu.edu/∼mdredze/data sets/sentiment/

II. Related Work

Sentiment analysis problem can be divided into single-domain [8], [9] and multiple-domain [10], [11] problems based on the domains of data which are used to train the classifier and later test the classifier. It can further be categorized into word-level, sentence level and document level sentiment analysis.

Single-domain Sentiment Analysis

In single-domain sentiment analysis problem, a classifier receives training using labeled data from the domain and later tested with data from the same domain. Turney [9] used five patterns for calculating semantic orientation on reviews. The polarity of the words and phrases are measured by the word which are occurring together with a set of choosen positively oriented words (e.g. excellent, good, nice etc.) and negatively oriented words (e.g. nasty, bad, poor etc.). This process used, a measure of association, pointwise mutual information to measure the sentimental orientation of a word. They achieved 84% accuracy on automobile review data and 66% on movie reviews. Pointwise mutual information method has been useful to weight features in many natural language processing tasks like word classification [14], word clustering [15] and similarity measurement [16]. 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 [17]. Association rule mining using Genetic Algorithm is used in the papers [18], [19], [20].

In Indian languages, works are comparatively less.

In Narayan et.al. [21], Hindi Subjective Lexicon and hindi WordNet has been used to identify the semantic orientation of adjectives and adverbs. In 2009, Dray et.al. [22] performed blog sentiment analysis to extract domain specific adjectives. First they automatically extracted from the Internet, a learning data set for a specific domain. Second they extracted from this learning set, the set of positive and negative adjectives relevant for the domain. Rao and Ravichandran [23], performed the classification of bi-polar nature. Amitava Das and Bandopadhya [24], suggested a computational method for evolving Senti-WordNet(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 obtain a Bengali SentiWordNet.

Das and Bandopadhya [25], made known four ways to judge the polarity of a word. The first method used an interactive game which identifies the polarity of the words. The second method developed a bi-lingual dictionary for English and Indian Languages. The third method expanded word net using antonym and synonym relations. The fourth method used a pre-annotated corpus for learning. Das and Bandopadhya [26], developed the method for tagging using the Bengali words. Classification of words is done into six emotion classes (happy, sad, surprise, fear, disgust, anger) according to three levels of intensities (low, general and

high). Joshi et al. [27] used two lexical resources: English-Hindi WordNet Linking [28] and English SentiWordNet and created H-SWN(Hindi-SentiWordNet). They substituted words in English SentiWordNet with synonymous Hindi words to get H-SWN using WordNet linking. They used a SVM classifier for identifying the polarity of the opinion. In this method they managed to create the H-SWN of 16253 synsets which consists of Adjective, Adverb, Noun and Verb. By using a graph based method Bakliwal et al. [29] created subjectivity lexicon. The lexicon was built using a seed list of 45 adjectives and 75 adverbs.

Namita Mittal et al. [30] developed an effective method based on negation handling and discourse relation to identify the sentiments from Hindi data. They generated an annotated corpus in Hindi language and improved the existing Hindi SentiWordNet (HSWN) by including more opinion words into it. The algorithm proposed by them was approximately 80% accurate in classifying reviews. Paper [31] 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. Paper [32] performed subjectivity analysis at the sentence level. They achieved 71.4% agreement with human annotators and 80% accuracy in classification on a parallel data set in English and Hindi. Paper [33] proposed a stopword removal algorithm for Hindi Language which is based on a Deterministic Finite Automata (DFA). They achieved 99% accurate results.

Multiple-domain Sentiment Analysis

In multiple-domain sentiment analysis problem, a classifier is trained using labeled data from single or multiple domains and later tested with data from the different domain. Blitzer et al. [10] proposed Structural Correspondence Learning (SCL) algorithm to train its multi-domain classifier. SCL method was built on the foundation of choosing a set of pivot features which gets repeated in both source and target domains when we have labeled data from a source domain and unlabeled data from both source and target domains. A linear predictor was trained to tell in advance the frequency of those pivot features. The learned weight vectors were lined up as rows in a matrix and Singular Value Decomposition (SVD) was executed to reduce the dimensionality of this matrix. Finally, this lower dimensional matrix was used to highlight features to train a binary sentiment classifier. It is worth noting that this method does not require any manually labeled feature vectors for understanding and learning the pivot feature predictors.

In 2010, Pan et al. [11] proposed Structural Feature Alignment (SFA) method to find a collaboration between domain specific and domain independent features. In this, Features were classified into domain-specific or domain-independent using the mutual information between a feature and a domain label. Both unigrams and bigrams were considered as features to set forth a review. They constructed a bipartite graph between domain-specific and domain-independent features. Between a domain-specific and a domain independent feature in the graph, an edge was formed if those two features co-occur in some feature vector. Post that, spectral clustering was performed to spot feature clusters. In the end, a binary classifier was trained using the feature clusters for categorization of positive and negative sentiment.

We create a sentiment aware dictionary for the multi-domain sentiment classification problem in Hindi language. However, to the best of our knowledge, multi-domain sentiment classification problem have not previously dealt in Hindi language. One work is available in multilingual data [34] which uses Hindi lanuage and Marathi language but it is also on single-domain data. Here, we use PMI Score and relatedness weights to decrease the dissimilarity of features between the two domains. Relatedness weights is already used in methods for query expansion [35], in information retrieval [36] and document classification [37] and it improves the results. However, it has not been used for multi-domain sentiment classification.

III. Proposed Work

Main Algorithm describes the overview of the proposed work.

First part involves the extraction of individual reviews from the review documents. The review documents contain reviewer's name, product name, rating, review text and other details. Review ratings range from 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 based on the ratings and extracted only the review text sentences from each review documents. We have considered 100 positive reviews and 100 negative reviews for the source domain and these are labeled reviews. Also 200 unprocessed reviews are considered and these are unlabeled reviews. Source domain reviews are a combination of all four domains from the review documents and 400 reviews in total. We considered 100 unprocessed, unlabeled reviews for the target domain i.e. kitchen appliances.

Main Algorithm: 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 i in ReviewList[] do
    |  Input = Input.append(ReviewList[i])
    end
    call Function 1: PMI Calculation with Input as
    argument
    PMIScore = PMI(Input)
    call Function 2: Relatedness Weight Calculation
    with PMIScore as argument
    Dictionary = relatedness(PMIScore)
end
```

TABLE II: Sample Reviews

	Source Domain	Target Domain
+ve	मैं इस उत्पाद से बहुत खुश हूँ, यह आरामदायक और सुन्दर है यह खरीदनेलायक है *(I'm very happy with this product, This is comfortable and beautiful, It is worth buying)*	यह सबसे अच्छा है, मेरी रसोई घर में मेरा सबसे प्रिय वस्तु है ! *(It's the best, the most beloved object in my kitchen !)*
-ve	यह उपयोग करने के लिए बहुत मुश्किल, बहुत छोटा और भारी है, मुझे पसन्द नहीं आया है *(I did not like this product, It is very difficult to use and too small and heavy)*	यह इस्तेमाल में कठिन है, वास्तव में, फर्श की सफाई का काम आसान नहीं कर सकता *(It is difficult to use, actually, can not easy the work of cleaning the floor)*

Table II displays one positive and one negative review from both domains as a sample.

Function 1: PMI Calculation

Data: Input from Main Algorithm
Result: PMI Score of each Bigram

begin
  Perform:
  Tokenize the Input data
  for *each Token* **do**
    Part-of-Speech tagging (Pos)
    if *Pos = noun||adverb||adjective||verb* **then**
      Write Token to FilteredList
    end
  end
  for *each i in FilteredList[]* **do**
    Unigram[i] = FilteredList[i]
    Bigram[i] = FilteredList[i] + FilteredList[i+1]
  end
  for *each i in unigram and bigram* **do**
    UFreq[i] =1
    BiFreq[i] =1
    for *each j in unigram and bigram* **do**
      if *unigram[i] = unigram[j]* **then**
        UFreq[i]+=1
      end
      if *Bigram[i] = Bigram[j]* **then**
        BiFreq[i]+=1
      end
    end
  end
  for *each i in Bigram* **do**
    $PMI[Bigram[i]] = log \frac{BiFreq[i]}{(UFreq[i]*UFreq[i+1])}$
  end
end

Next, for each labeled reviews of the source domain, sentiment awareness is created by appending label to each token in that review. For example, if a review is positive, then all the tokens are appended with "*P" and for negative reviews, "*N" is appended. Sentiment awareness are obtained only from labeled reviews in the source domain. The combination of labeled and unlabeled reviews are then subjected to Part-Of-Speech (POS) tagging and lemmatization using hindi-pos-tagger[3]. Lemmatization is the process of removing inflectional endings properly with the use of a vocabulary and morphological analysis of words and to return the base form or dictionary form of a word, which is called lemma. Lemmatization is an effective method in text classification [38] as it reduces feature sparseness. POS tagging is used to know the part of speech of each tokens in review sentences. A simple word filter is used to retain words that are nouns, verbs, adjectives and adverbs. These are the sound clues of sentiments [39], [40]. For each filtered list, unigrams list and bigrams list are generated. As explained in Function 1, we then compute the Pointwise Mutual Information (PMI) as *f(x,z)* between a lexical or sentiment element *x* and feature *z* for each unigram and bigram is as follows:

$$f(x,z) = log\left(\frac{\frac{c(x,z)}{N}}{\frac{\sum_{i=1}^{n} c(i,z)}{N} * \frac{\sum_{j=1}^{m} c(x,j)}{N}}\right)$$

Here, the total number of reviews in which a lexical element *x* and a feature *z* co-occur is represented as *c(x,z)*, *n* and *m* are total number of *x* and *z* respectively and $N = \sum_{i=1}^{n}\sum_{j=1}^{m} c(i,j)$. We have considered only positive PMI values to overcome the bias of PMI towards infrequent words and features i.e. words and features that occur only once might have negative PMI values. Next step is to calculate Relatedness Weight for the elements *x* and *y* as *r(y,x)* and is detailed in Function 2.

Relatedness weight explains the features of element *x* that it shares with element *y*. This weight is asymmetric as relatedness weight *r(y,x)* will not always be equal to relatedness weight *r(x,y)* i.e. words that co-occur in one order need not co-occur in the reverse order. We have only considered positive relatedness weights. Next step is dictionary creation. For each element *x*, we use the relatedness weight *r(y,x)* to list all the elements *y* that co-occur with element *x*. An example of this is, for the word उत्कृष्ट *(Excellent)*, the words listed in the dictionary are अद्भुत *(Amazing)* and स्वादिष्ट *(Delicious)*.

IV. Simulation Results

Table III shows a sample of unigrams and bigrams with its frequency, PMI Score and relatedness weights. Frequency is calculated as the total number of occurrence of a feature in a review and is used to calculate PMI score and relatedness

[3]http://sivareddy.in/downloads#hindi_tools

Function 2: Relatedness Weight Calculation

Data: PMI Score from Function 1
Result: A file containing Relatedness Weight

```
begin
    Initialize:
    Num=0, Den=0, y=1, x=2
    Perform:
    for each y, x in PMI do
        for all z which is neighbour to y do
            if PMI[x,z] exists and > 0 then
                Num += PMI[x,z]
            end
        end
        for all z which is neighbour to x do
            if PMI[x,z] exists and > 0 then
                Den += PMI[x,z]
            end
        end
        relatedness[y,x] = Num / Den
        if relatedness[i] > 0 then
            Dict[i]=relatedness[i]
        end
    end
end
```

TABLE III: Sample of Tokens with its frequency (F), PMI Score (PScore) and Relatedness Weight (RWeight)

Unigram	F	Bigram	F	PScore	RWeight
महान	14	महान + सेवा	1	1.63	14.67
अलग	2	अलग + डीवीडी	1	1.88	4.06
डीवीडी	4	डीवीडी + प्लेयर	1	2.18	4.06
पत्रिका*N	1	पत्रिका*N + आनंदमय*N	1	2.78	2.96
अद्भुत	1	अद्भुत + खिलौना	1	2.78	2.66
महान	14	महान + उत्पाद	1	0.86	2.49
महान	14	महान + महान	1	0.49	2.23
शानदार	2	शानदार + पैंट	1	2.48	1.84
शिप्पिंग*P	2	शिप्पिंग*P + समय*P	2	2.18	1.35

weights of each feature. Next PMI is calculated and it is demonstrated by the following example.

Example: For the given data in table IV, suppose, TotalBigrams = 1298 and TotalUnigrams = 1299 then equation (1), (2) and (3) shows PMI computation for three different bigrams:

TABLE IV: Example demonstrating computation of PMI

Sl.No.	Unigram	F	Bigram	F	PMI
1	उत्पाद	20	उत्पाद + खुश	1	1.03
2	खुश	6	खुश + आराम	2	2.34
3	आराम	2	आराम + दायक	2	2.81
4	दायक	2			

$$
\begin{aligned}
&\text{PMI}(\text{उत्पाद} + \text{खुश}) \\
&= \text{Round}\left(\text{Log}\ \frac{\frac{F(\text{उत्पाद} + \text{खुश})}{\text{TotalBigrams}}}{\frac{F(\text{उत्पाद})}{\text{TotalUnigrams}} * \frac{F(\text{खुश})}{\text{TotalUnigrams}}}\right) \\
&= \text{Round}\left(\text{Log}\ \frac{\frac{1}{1298}}{\frac{20}{1299} * \frac{6}{1299}}, 2\right) \\
&= \text{Round}\left(\text{Log}\ \frac{0.000770416}{0.0153964588 * 0.0046189376}, 2\right) \\
&= \text{Round}(\text{Log}(10.8333397535), 2) \\
&= \text{Round}(1.0347623636, 2) \\
&= 1.03
\end{aligned} \tag{1}
$$

$$
\begin{aligned}
&\text{PMI}(\text{खुश} + \text{आराम}) \\
&= \text{Round}\left(\text{Log}\ \frac{\frac{F(\text{खुश} + \text{आराम})}{\text{TotalBigrams}}}{\frac{F(\text{खुश})}{\text{TotalUnigrams}} * \frac{F(\text{आराम})}{\text{TotalUnigrams}}}\right) \\
&= \text{Round}\left(\text{Log}\ \frac{\frac{2}{1298}}{\frac{6}{1299} * \frac{2}{1299}}, 2\right) \\
&= \text{Round}\left(\text{Log}\ \frac{0.0015408320493}{0.0046189376 * 0.00153964588}, 2\right) \\
&= \text{Round}(\text{Log}(219.666790216), 2) \\
&= \text{Round}(2.3417644041388, 2) \\
&= 2.34
\end{aligned} \tag{2}
$$

$$
\begin{aligned}
&\text{PMI}(\text{आराम} + \text{दायक}) \\
&= \text{Round}\left(\text{Log}\ \frac{\frac{F(\text{आराम} + \text{दायक})}{\text{TotalBigrams}}}{\frac{F(\text{आराम})}{\text{TotalUnigrams}} * \frac{F(\text{दायक})}{\text{TotalUnigrams}}}\right) \\
&= \text{Round}\left(\text{Log}\ \frac{\frac{2}{1298}}{\frac{2}{1299} * \frac{2}{1299}}, 2\right) \\
&= \text{Round}\left(\text{Log}\ \frac{0.0015408320493}{0.00153964588 * 0.00153964588}, 2\right) \\
&= \text{Round}(\text{Log}(650.0029529635), 2) \\
&= \text{Round}(2.8129153296, 2) \\
&= 2.81
\end{aligned} \tag{3}
$$

We are calculating relatedness weights because even if PMI score of a bigram is high, the relatedness weight may increase or decrease depending on the whole review document which consists of these bigrams and having specific co-occurrence factor. PMI score is biased towards less occurring words. For the less frequent bigrams, it is directly proportional to the relatedness weights, i.e., when PMI score is increasing,

relatedness weights are also increasing and is shown in Fig. 1. For the more frequent bigrams, it is inversely proportional

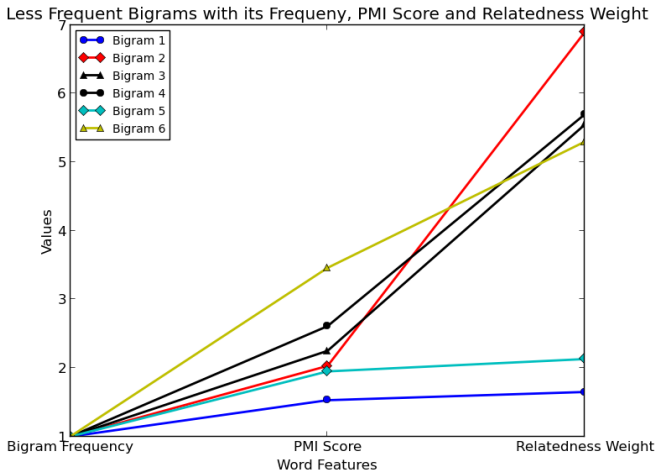

Fig. 1: Result of a sample of Bigrams with its frequency, PMI Score and Relatedness Weight where Bigram 1 = "अच्छा+आकार", Bigram 2 ="असली+काम", Bigram 3 = "अद्भुत+उपहार", Bigram 4 ="अद्भुत+जानकारी", Bigram 5 = "अच्छा*P+आकार*P", Bigram 6 = "आकार*P+उज्जवल*P"

to the relatedness weights, i.e., when PMI score is increasing, relatedness weights are decreasing and is shown in Fig. 2.

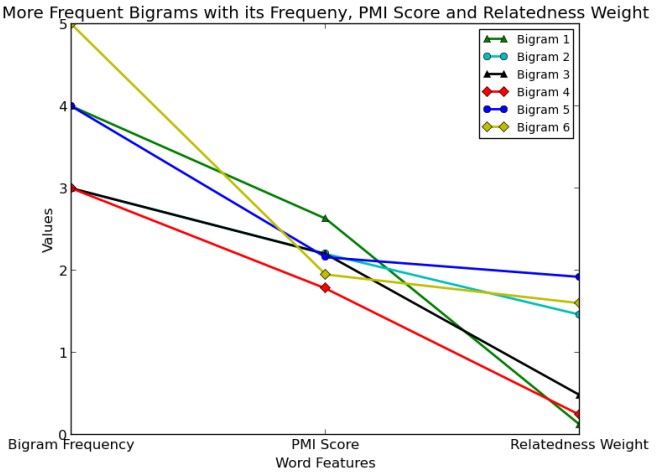

Fig. 2: Result of a sample of Bigrams with its frequency, PMI Score and Relatedness Weight where Bigram 1 = "आराम+दायक", Bigram 2 = "पूरी+तरह", Bigram 3 = "लंबा+समय", Bigram 4 ="समय+पसंदीदा", Bigram 5 = "उत्कृष्ट+दस्तावेजी", Bigram 6 = "अच्छी*P+तरह*P"

Table V shows a sample of Sentiment Aware Dictionary. This dictionary is obtained, after sorting the results of PMI score and relatedness weights, a sample of this is shown in table III. 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 weights. After this, for each unigram, we have all the related words. In table V, Base_word is given with its related word count, 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 as word1, word2 and so on according to its count value.

V. Conclusions

We introduce an innovative way to find the relatedness of large word data set 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 dictionary consists of Hindi nouns, adjectives, verbs and adverbs organized into a set of words representing the frequently occurring words with high weights. In future, we are planning to use this sentiment aware dictionary to train a classifier which can classify the product reviews from multiple domains and can work on unlabeled data as effectively as on labeled data.

References

[1] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Foundations and trends in information retrieval*, vol. 2, no. 1-2, pp. 1–135, 2008.

[2] Y. Lu, C. Zhai, and N. Sundaresan, "Rated aspect summarization of short comments," in *Proceedings of the 18th international conference on World wide web*. ACM, 2009, pp. 131–140.

[3] T.-K. Fan and C.-H. Chang, "Sentiment-oriented contextual advertising," *Knowledge and Information Systems*, vol. 23, no. 3, pp. 321–344, 2010.

[4] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2004, pp. 168–177.

[5] H. Takamura, T. Inui, and M. Okumura, "Extracting semantic orientations of phrases from dictionary." in *HLT-NAACL*, vol. 2007, 2007, pp. 292–299.

[6] E. Breck, Y. Choi, and C. Cardie, "Identifying expressions of opinion in context." in *IJCAI*, vol. 7, 2007, pp. 2683–2688.

[7] H. Yu and V. Hatzivassiloglou, "Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences," in *Proceedings of the 2003 conference on Empirical methods in natural language processing*. Association for Computational Linguistics, 2003, pp. 129–136.

[8] B. Pang, L. Lee, and 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.

[9] P. D. Turney, "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, pp. 417–424.

[10] J. Blitzer, M. Dredze, and F. Pereira, "Jun. 2007. biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification," in *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pp. 440–447.

TABLE V: Sample of Sentiment Aware Dictionary

Base_word	Related Word Count	Word1	Word2	Word3	Word4	Word5	Word6
आनंद	2	अलौकिक	उत्पाद				
आरामदायक*P	1	सुन्दर*P					
बढिया	2	कीमत	संग्रह				
बढिया*P	1	संग्रह*P					
मुश्किल*N	3	कठोर*N	छोटा*N	हाथ*N			
संस्करण*P	2	अच्छा*P	सुपर*P				
समय*N	4	पूरी*N	विफल*N	बर्बाद*N	बर्बादी*N		
सुन्दर*P	6	लायक*P	सही*P	उत्पाद*P	शानदार*P	खुशबू*P	नया*P
स्वच्छ*P	2	सरल*P	ताजा*P				

[11] S. J. Pan, X. Ni, J.-T. Sun, Q. Yang, and 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.

[12] [Online]. Available: http://en.wikipedia.org/wiki/List_of_languages_by_number_of_native_speakers

[13] (2013, December 31). [Online]. Available: http://www.internetworldstats.com/stats7.htm

[14] P. Pantel and D. Ravichandran, "Automatically labeling semantic classes." in *HLT-NAACL*, vol. 4, 2004, pp. 321–328.

[15] D. Lin, "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, pp. 768–774.

[16] P. D. Turney, "Similarity of semantic relations," *Computational Linguistics*, vol. 32, no. 3, pp. 379–416, 2006.

[17] Z. S. Harris, "Distributional structure." *Word*, 1954.

[18] P. D. Shenoy, K. Srinivasa, K. Venugopal, and L. M. Patnaik, "Evolutionary approach for mining association rules on dynamic databases," in *Advances in knowledge discovery and data mining*. Springer, 2003, pp. 325–336.

[19] P. D. Shenoy, K. Srinivasa, K. Venugopal, and L. M. Patnaik, "Dynamic association rule mining using genetic algorithms," *Intelligent Data Analysis*, vol. 9, no. 5, pp. 439–453, 2005.

[20] V. H. Bhat, P. G. Rao, R. Abhilash, P. D. Shenoy, K. Venugopal, and L. Patnaik, "A data mining approach for data generation and analysis for digital forensic application," *IACSIT International Journal of Engineering and Technology*, vol. 2, no. 3, pp. 314–319, 2010.

[21] D. Narayan, D. Chakrabarti, P. Pande, and P. Bhattacharyya, "An experience in building the indo wordnet-a wordnet for hindi," in *First International Conference on Global WordNet, Mysore, India*, 2002.

[22] G. Dray, M. Plantié, A. Harb, P. Poncelet, M. Roche, and F. Trousset, "Opinion mining from blogs," *International Journal of Computer Information Systems and Industrial Management Applications (IJCISIM)*, vol. 1, pp. 205–213, 2009.

[23] D. Rao and D. Ravichandran, "Semi-supervised polarity lexicon induction," in *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2009, pp. 675–682.

[24] A. Das and S. Bandyopadhyay, "Sentiwordnet for bangla," *Knowledge Sharing Event-4: Task*, vol. 2, 2010.

[25] A. Das and S. Bandyopadhyay, "Sentiwordnet for indian languages," *Asian Federation for Natural Language Processing, China*, pp. 56–63, 2010.

[26] D. Das and S. Bandyopadhyay, "Labeling emotion in bengali blog corpus–a fine grained tagging at sentence level," in *Proceedings of the 8th Workshop on Asian Language Resources*, 2010, p. 47.

[27] A. Joshi, A. Balamurali, and P. Bhattacharyya, "A fall-back strategy for sentiment analysis in hindi: a case study," *Proceedings of the 8th ICON*, 2010.

[28] A. K. Karra, P. Pande, R. Railkar, A. Sharma, and P. Bhattacharyya, "Hindi english wordnet linkage," 2009.

[29] A. Bakliwal, P. Arora, and V. Varma, "Hindi subjective lexicon: A lexical resource for hindi polarity classification," in *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC)*, 2012.

[30] N. Mittal, B. Agarwal, G. Chouhan, N. Bania, and P. Pareek, "Sentiment analysis of hindi review based on negation and discourse relation," in *Sixth International Joint Conference on Natural Language Processing*, 2013, p. 45.

[31] V. Jha, N. Manjunath, P. D. Shenoy, K. Venugopal, and L. Patnaik, "Homs: Hindi opinion mining system," in *Recent Trends in Information Systems (ReTIS), 2015 IEEE 2nd International Conference on*. IEEE, 2015, pp. 366–371.

[32] V. Jha, N. Manjunath, P. D. Shenoy, and K. Venugopal, "HSAS: Hindi Subjectivity Analysis System," in *2015 Annual IEEE India Conference (IEEE INDICON 2015)*, Jamia Millia Islamia, New Delhi, India, 2015.

[33] V. Jha, N. Manjunath, P. D. Shenoy, and K. Venugopal, "HSRA: Hindi Stopword Removal Algorithm," in *2015 IEEE International WIE Conference on Electrical and Computer Engineering (Wiecon-ECE 2015)*, Bangladesh University of Engineering and Technology (BUET), Dhaka, Bangladesh, 2015.

[34] A. Balamurali, "Cross-lingual sentiment analysis for indian languages using linked wordnets," 2012.

[35] H. Fang, "A re-examination of query expansion using lexical resources." in *ACL*, vol. 2008. Citeseer, 2008, pp. 139–147.

[36] M. Dillon, "Introduction to modern information retrieval: G. salton and m. mcgill. mcgraw-hill, new york (1983). xv+ 448 pp." 1983.

[37] D. Shen, J. Wu, B. Cao, J.-T. Sun, Q. Yang, Z. Chen, and Y. Li, "Exploiting term relationship to boost text classification," in *Proceedings of the 18th ACM conference on Information and knowledge management*. ACM, 2009, pp. 1637–1640.

[38] T. Joachims, *Text categorization with support vector machines: Learning with many relevant features*. Springer, 1998.

[39] V. Hatzivassiloglou and K. R. McKeown, "Predicting the semantic orientation of adjectives," in *Proceedings of the 35th annual meeting of the association for computational linguistics and eighth conference of the european chapter of the association for computational linguistics*. Association for Computational Linguistics, 1997, pp. 174–181.

[40] J. Wiebe, "Learning subjective adjectives from corpora," in *AAAI/IAAI*, 2000, pp. 735–740.