International Journal of Current Advanced Research
ISSN: O: 2319-6475, ISSN: P: 2319-6505, Impact Factor: 6.614
Available Online at www.journalijcar.org
Volume 7; Issue 4(A); April 2018; Page No. 11335-11343
DOI: http://dx.doi.org/10.24327/ijcar.2018.11343.1959

Research Article

OPINION MINING FROM ONLINE REVIEWS USING OPINION AND SEMANTIC RELATIONS

Savitha Mathapati*., Shreelekha B S., Tanuja R., S H Manjula and Venugopal K R

Department of Computer Science and Engineering, University Visvesvaraya College of Engineering, Bangalore University, Bengaluru, India

A R T I C L E I N F O

Article History:

Received 24th January, 2018
Received in revised form 13th February, 2018 Accepted 8th March, 2018
Published online 28th April, 2018

Key words:

Opinion Mining; Opinion Target Extraction; Opinion Word Extraction; Semantic Relations; Word Alignment.

A B S T R A C T

Mining opinions from online reviews is a fundamental step in obtaining the overall sentiment of a product. Detection of opinion relations among the words play an important role in the opinion target *(OT)* and opinion word *(OW)* extraction. In this paper, Partially Supervised Word Alignment Model is used to find opinion relations among words. Graph based co-ranking algorithm is used in estimating the confidence of each *OT* and *OW*. Candidates having confidence value higher than the threshold are extracted as final *OT* and *OW*. We propose a hybrid method that considers semantic relations along with opinion relations that result in fine grained opinion target *(OT)* and opinion word *(OW)* extraction. This semantic relations and opinion relations affect the confidence calculation of the *OT* and *OW* and improve the precision of extraction.

INTRODUCTION

In the task of analyzing reviews on a product, obtaining the overall sentiment is not sufficient. The consumers expect to get the fine grained opinion on the desired product. For example, what aspect of the product is the main highlighting feature of it, what aspect is making it to not keep up with the other market products *etc*.. To illustrate this, let us consider a review for example:

Review 1: "*This TV has a vivid and large screen, but its clarity is disappointing.*"

In the Review 1, the reader can get to know that the reviewer is expressing positive opinion on the size and color of the *"screen"*. On the other hand, he is disappointed with the clarity of the *"screen"* and hence he expresses negative feeling. Buyer expects all these details to be presented as summary rather than reviewer's overall sentiment (Positive or Negative) on the product. Users express his/her opinion on an object that is called Opinion Target (*OT*). Sentiment in the reviews is expressed on these opinion targets that are noun/noun phrases in the sentence. In the above example the *OT* are *"screen"* and *"clarity"*, as the review is commenting on these features of the product. The words that qualify the *OT* are called as Opinion Words (*OW*). In the above example the words *"vivid"*, *"large"* and *"disappointing"* are the words that are likely to be *OW*.

Corresponding author:* **Savitha Mathapati
Department of Computer Science and Engineering,
University Visvesvaraya College of Engineering,
Bangalore University, Bengaluru, India

The lexicon is used to identify the opinion words in the reviews.

Nearest neighbour and syntactic patterns are the most used and implemented methods for extracting the Opinion Targets (*OT*) and Opinion Words (*OW*) in the given set of reviews. These standard methods have their own limitations while extracting *OT* and *OW*. In nearest neighbour method, for a particular noun/noun phrase, its nearest adjective in a restricted window size is selected as its modifier. For example:

Review 2: *"This phone has amazing sound quality, but the maximum loudness reachable is not satisfactory"*

The Review 2 is on the aspect "*sound*" of the product "*phone*". In the first part of the review, the reviewer expresses positive opinion on the *"sound quality"*. Hence according to the nearest neighbor method, the modifier for "*sound*" is captured as "*amazing*". But as we can see in the second part of the review, reviewer is expressing the negative opinion on "*sound*" aspect of the "*phone*". He says that the "*maximum volume of the sound is not satisfactory*". As this part is very far from the noun "*sound*", it is not captured. The main drawback of the Nearest Neighbor method is that it cannot capture the long window size modifiers in case of extended span modifiers. Syntactic patterns method can handle this problem of nearest neighbor method. Using this method, we can parse the reviews grammatically and construct the parsing trees. These trees give the complete information of the grammatical dependencies among the words in the given sentence. This is a well-trained tool that has been trained with the Syntactic or grammatical rules and regulations. There are many important syntactic patterns designed [1], [2], [3]. But, the reviews that are collected for the implementation generally have informal style of writing. Those reviews may be written with the grammatical mistakes, punctuation mistakes and typological mistakes. Using these trained tools on reviews, we end up in inaccurate results. This is because the tool is trained using the standard text formats like news report. Hence it is a challenging task to extract opinion target and opinion word.

To overcome the challenges present in nearest neighbour and syntactic patterns methods to extract *(OT)* and *(OW)*, we use alignment-based method along with graph based co-ranking to extract *OT* and *OW*. To accurately mine opinion relations, we use word alignment model (WAM) [4]. Opinion modifier can find its corresponding opinion target using word alignment. Let us consider an example in Fig. 1 that shows the opinion relations between the words using word alignment model. Here target word *"screen"* is aligned with opinion words *"colorful"* and *"big"*. As there is no restriction on the window size, WAM captures long-span modified relations. Word alignment model is trained in an unsupervised method. This results in unsatisfactory quality of alignment. This is overcome by supervised training that certainly improves the alignment quality. However, it is impractical to manually label every alignment in the sentences and is also time consuming process. Thus, further partially supervised word alignment model *(PSWAM)* easily get portions of the links of the full alignment. To obtain partial alignments, a constrained *Expectation-Maximization* algorithm based Hill-climbing algorithm is applied to extract all possible alignments in sentences. This method removes the wrong alignments that may occur due to completely unsupervised *WAM*. Unsupervised alignments, syntactic patterns alignment and partially supervised word alignments are represented with example in the Fig. 2(a),(b) and (c) respectively. From the Fig. 2(a), we observe that under unsupervised alignments, *"food"* is modified by *"tasty"*, *"heartly"* and *"excellent"* words that is actually not true. *"heartly"* and *"excellent"* are the actual modifiers of the noun *"services"*. This is one of the example where the unsupervised training results in irrelevant links. Using syntactic patterns alignment as shown in 2 (b), it removes the disadvantage of nearest neighbour method and still *"heartly"* is not identified as modifier of *"services"*. In 2 (c) representing partially supervised word alignments, the opinion target *"food"* is correctly aligned with the opinion word *"tasty"*. Also the *OT* *"services"* is correctly aligned with *OW* *"heartly"* and *"excellent"* in the second part of the review. Extraction of opinion target/word is called as a co-ranking process. A Random walk based co-ranking algorithm is used to estimate the *OT* and *OW* candidate's confidence. Finally, candidates with higher confidence than a threshold are extracted as final opinion targets and opinion words.

Motivation: Liu *et al.,* [5] proposed Partially Supervised Word Alignment Model (*PSWAM*) to extract the opinion relations from the reviews. Compared to unsupervised Word Alignment model, *PSWAM* gives better precision value due to partial supervision and it captures Opinion Relations more precisely. Only using opinion relations gives lower precision value compared to using other relations such as, Conjunction relations, Semantic relations *etc.,* along with it.

Contribution: In this paper, we propose Partially Supervised Word Alignment Model (PSWAM) to extract opinion targets and opinion words using both opinion and Semantic relations.

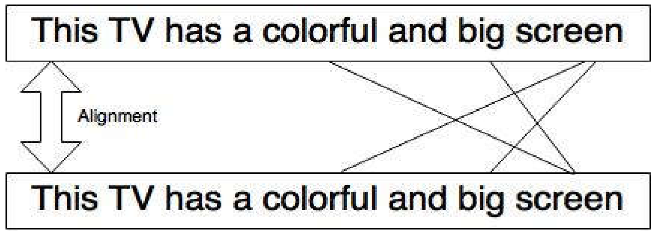
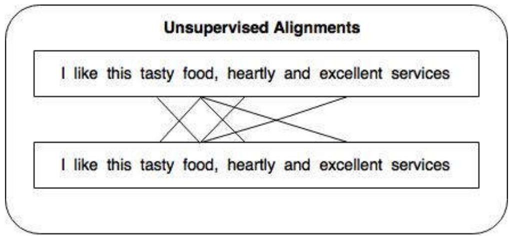
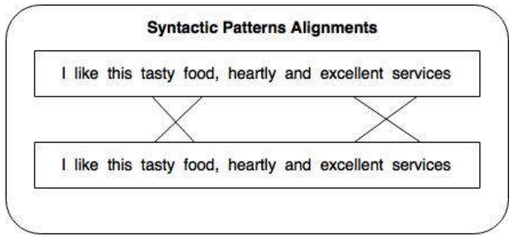


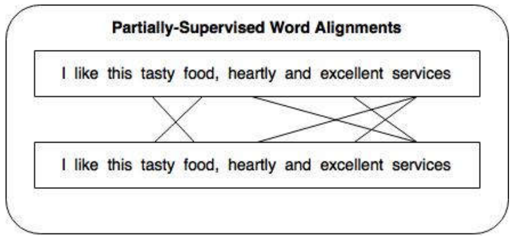
Fig 1 Extraction of Opinion Relations between Words using Word Alignment Model



a Unsupervised Alignments



b Syntactic Patterns Alignments



c Partially-Supervised Word Alignments

Fig 2 Comparison of Different Alignment Models

Semantic relation exists among the homogeneous words. As an example consider the words *"LCD"* and *"LED"*, here both of them denote the same attribute screen in the domain of *"Television"*. These words are topically related to each other. Heterogeneous graph model is constructed to model both the relationships. Addition of semantic relations improves estimation of confidence of the candidates.

The rest of the paper is organized as follows. Section 2 analyzes previous related works. Problem definition is stated in section 3. An overview of the Word Alignment Model and Partially Supervised Word Alignment Model is described in section 4 and 5 respectively. Constrained Hill-climbing Algorithm is explained in section 6. Section 7 describes the need of semantic relations in *OT* and *OW* extraction. Section 8 gives a description of the implementation part of the algorithm. Performance analysis with results is analyzed in section 9. Lastly section 10 concludes the paper.

Related Work

Opinion Target *(OT)* and Opinion Word *(OW)* extraction are the major goals in the field of opinion mining. There are several efforts made to extract *OT* and *OW* using different approaches. The extraction process is categorized into sentence level extraction and corpus level extraction. In the method of sentence level extraction, the opinion target or opinion word extraction is the task of identifying the opinion target/word in each sentence. Hence this task is considered as sequence labeling problems [6], [7], [8], [9]. Common example for sequence labeling is part-of-speech tagging. Related words are extracted as features to represent the *OT* or *OW* in sentences. Also the extractors such as Conditional Random Fields *(CRF)* and Hidden Morkov Model *(HMM)* are built by using the standard sequence labeling model. The model of lexicalized *HMM* was proposed by Huang *et al.,* [10] for the purpose of opinion mining. *CRFs* were used in the implementation to extract *OT* from reviews. Methods *CRF* and *HMM* requires the labeled data for training the model. The extraction performance of these methods become unsatisfied if the labeled data for training the model is insufficient or if the labeled data appear from dissimilar domain. Li *et al.,* [11] proposed a transfer learning technique for cross domain selection of *OT* and *OW*.

The task of identifying the opinion relations among the words is the important component. Wang *et al.,* [12] incor-porated the method of co-occurrence frequency of *OT* and *OW* for calculating their opinion association. Liu and Hu [13] identified the opinion relation among words using the nearest neighbor method. Further bootstrapping method was used for extracting the frequent and explicit features. These methods provide the satisfactory performances. But only using the co-occurrence frequency information or the rules of nearest neighbor method cannot give the precise result in identifying opinion relations among words. Hence the method proposed in [2] used the syntax information in *OT* extraction and few syntactic patterns are designed to identify the opinion relations among words. Experimental results in [2] proved that they have obtained better results compared to method in [13]. Qiu*et al.,* [1] proposed Double propagation that expands the *OT* and sentiment words iteratively by exploiting the syntactic relations among words. The syntactic relations have the limitation of not covering all opinion relations by using the dependency parsing tree. Hence, Zhang *et al.,* [14] defined a method where, along with the patterns used in [1], added specific patterns that increased the recall. Hyperlink-induced Topic Search (*HITS*) [15] algorithm is used to calculate the confidence of opinion target candidates that increases the precision value. Liu *et al.,* [16] extracted the opinion target (*OT*) based on Word Alignment Model (*WAM*). It is proved that *WAM* was very effective in extraction of *OT*. Thus, no evidences demonstrate that *WAM* is efficient for opinion word extraction. Completely unsupervised WAM was used to identify the opinion relations in given sentences. Further the extraction of *OT* was carried out using the standard framework called random walk. Mei *et al.,* [17] and Titov and McDonald [18] explained topic modeling for identifying the sentiment words and implicit topics. These methods aimed at extracting *OT* list or *OW* lexicon from the reviews obtained and also to cluster all words to their respective aspects in reviews. These methods for identifying the proper *OT* or *OW* incorporated the coarser techniques like phrase detection and statistics on frequency that focus on clustering the words [19], [20].

Problem Definition

Extraction of opinion targets and opinion words from online tweets is efficient when both opinion and semantic relations are considered. The objective of the proposed work is to extract the opinion targets and the opinion words using both relations. This is achieved using Partially Supervised Word Alignment Model using Constrained Hill Climbing algorithm for detecting opinion relations and Random walk algorithm for calculating the confidence of opinion target/word. To get Hybrid model of semantic relations and opinion relations, two Random walk algorithms are coupled. While calculating the confidence of opinion target/word, both the relations are considered. This results in an extraction of a more precise opinion target/word. Our model is efficient for tweets as it is designed to work on informal text.

Word Alignment Model

The proposed system uses the Word Alignment Model *(WAM)*. The *WAM* is based on monolingual alignments where monolingual sentence is replicated twice and relations are drawn among the words between two replicates. Any opinion target can find its modifier using the *WAM*. Fig. 1 represents monolingual *WAM*. In Fig. 1, the opinion words *"big"* and "colorful" are aligned with the target *"screen"*. In comparison with the methods like nearest neighbor and syntactic patterns, the *WAM* finds the alignments more precisely. By comparing to the nearest neighbor method, *WAM* is not limited in identifying the modifier within the restricted window size. As the *WAM* do not parse any text, it does not find the modifiers based on the syntactic patterns. Hence, it need not parse the text. This method integrates many factors like word position, co-occurrence frequency *etc.,* for finding opinion relations among words.

The word alignment model is a natural language processing task of identifying relationships among the words those results in bipartite graph. The bipartite graph is represented by *G(V,E, W)* and it is called as opinion relation graph. In the graph G, vertices set is $V = V_t \cup V_o$ has two sets of vertices $v_t \in V_t$ is the set of vertices that denote the opinion target (*OT*) candidates. Similarly, $v_o \in V_o$ is the set of vertices that denotes the opinion word (*OW*) candidates. In the graph *G, E* represents the set of edges where, $e_{ij} \in E$ represents that the opinion relation exists between two vertices at position *i* and *j*. The main condition in bipartite graph is that the edge e_{ij} should exist only between the sets V_t and V_o.

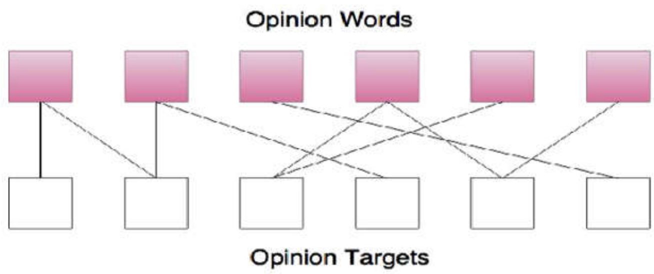


Fig 3 Opinion Relation Graph

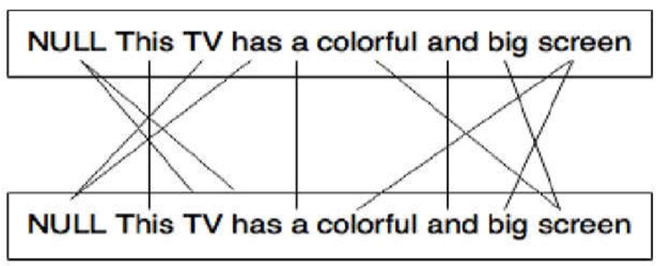


Fig 4 Word Alignment Model after Applying Constraints

Term W in graph *G* is the weight, where $w_{ij} \epsilon W$ represents the weight of edge eij.

The Fig. 3 represents the opinion relation graph. One set of vertices represents opinion words and the other set represents opinion targets. The edges exist only between vertices of *OT* set and *OW* set. It is called as a bipartite graph as there cannot exist any edges among same set of vertices. Similar relation exists between the words of a sentence.

Given a sentence S with *n* words where, $S = \{W_1, W_{2,} \dots, W_{n,}\}$, the word alignment $A = \{(i, a_i | i \in [1, n], a_i \in [1, n])\}$ can be obtained using Equation (1).

$$A^* = argmaxP(A|S) \quad (1)$$

where (i, a_i) indicates that a noun or noun phrase at position *i* is aligned with its modifier at position a_i. We are using *IBM-3* [21] word alignment model that perform better than earlier models. Thus, we have Equation (2).

$$P_{ibm3}(A|S) \propto \prod_{i=1}^{n} n(\varphi_i|w_i) \prod_{j=1}^{n} t\left(w_j \middle| w_{a_j}\right), d(j\,|a_j, n) \quad (2)$$

The three main factors in Equation (2) are

1. $n(\varphi_i|w_i)$ Indicates the ability of a word for *"one-to-many"* relation, *i.e.,* Word can modify or can get modified by several words. Number of words that are aligned with w_i is denoted by $\varphi_{i.}$ For example,

Review 3: "*This TV has a colorful and big screen*"

In review 3, *"colorful"* and *"big"* are the two words used to modify word *"screen"*. Hence, *i* equal to 2 for word *"screen"*.

2. $t(w_j|\, w_{a_j})$ indicates the co-occurrence informationof two words in corpora. If a word frequently modifies a noun/noun phrase, then these words have higher value of $t(w_j|\, w_{a_j})$. For example, in the reviews of *TV*, *"big"* often co-occurs with the *"TV"* size. and hence *"big"* has high association with *"TV"* size.
3. $d(j\,|a_j, n)$ indicates the probability that a word inposition a_j is aligned with a word in position *j*.

In Words Alignment Model, some constraints are intro-duced. This is because by directly using the alignment stan-dards, the opinion target may g*et al*igned with some irrelevant words than with the required opinion word. It may contain the irrelevant alignments like, *OT* with the preposition or *OT* with conjunction. The introduced constraints are as follows:

- Nouns/noun phrases (adjectives/verbs) must be aligned with the adjectives/verbs (Nouns/noun phrases) or a NULL word. A word is aligned to a NULL word indicates that the word has no modifier or modifies nothing.
- The irrelevant words like adverbs, conjunction and prepositions should align with themselves.

The Fig. (4) shows the alignments after applying the con-straints. Nouns or the noun phrases are supposed to be aligned with adjectives. Here the word *"screen"* is noun, *"colourful"* and *"big"* are adjectives. The noun in the sentence aligns only with the adjectives. Irrelevant words like *"a"*, *"This"* aligns with themselves. Words *"TV"*, *"has"* are aligned with the *NULL*. Even though the word *"TV"* is a noun still it is aligned with *NULL* as it has no modifiers to modify it.

Partially Supervised Word Alignment Model (PSWAM)

The standard word alignment models are usually not trained under supervision that leads to unsatisfactory performance. The result is enhanced by improvement in the alignment quality by training under supervision. On the other hand, it is time consuming process and manually labeling all the alignments in the sentence is practically impossible.

To improve the performance of alignment, we perform partially-supervised word alignment model. Here, partial alignment links are used as the constraints for the trained alignment algorithm. Given the partial alignment links $A^{**} = \{(i,\ a_i)|i \in [1, n], a_i \in [1, n]\}$ optimal alignment A* in Equation (1) is rewritten in Equation (3).

$$A^* = argmaxP(A|S, A^{**}) \quad (3)$$

Alignments are obtained using the *PSWAM* as shown in Fig. 2 (c). It is observed that the target "food" is correctly aligned with the word *"tasty"* and the target "services" is correctly aligned with words "*heartly"* and *"excellent"*.

Consistent alignments are generated using *PSWAM* as la-beled partial alignment are used. GIZA++ [22] provide a Hill-climbing algorithm to extract all potential alignments. GIZA++ first trains the simple models (IBM-1, IBM-2) as initial alignments for the IBM-3 model. To find the optimal alignments, a greedy search algorithm is used iteratively. The search space for the optimal alignment is constrained on the "neighbour alignments" of the current alignment, where "neighbour alignments" denote the alignments that is generated from the current alignment by one of the following operators:

1. MOVE operator $m_{i,j}$, that changes $a_j = i$.
2. SWAP operator s_{j_1,j_2} that exchange a_{j_1} and a_{j_2}.

Two matrices, *MOVE* matrix *M* and *SWAP* matrix *S* are created to record all possible MOVE and SWAP costs respectively between two alignments. To make the trained alignments consistent with the pre-provided partial alignments, illegal operation costs in *M* and *S* are set to -1. By this method, inconsistent alignments are never picked up.

Constrained Hill Climbing Algorithm

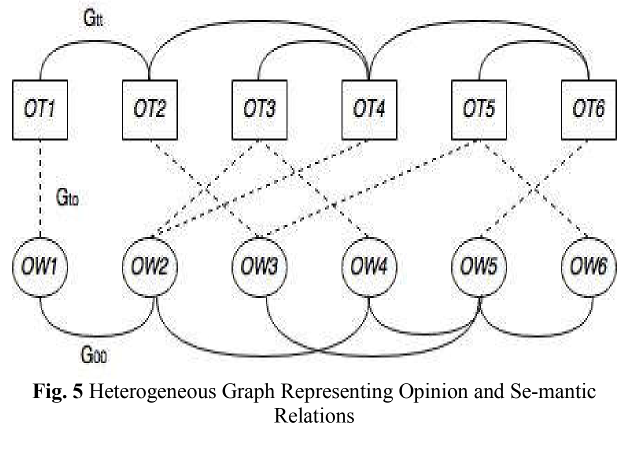


Fig. 5 Heterogeneous Graph Representing Opinion and Se-mantic Relations

Algorithm: Constrained Hill-climbing Algorithm

Input : Review sentences $S=\{w_1, w_2, \ldots, w_n\}$
Output: The calculated alignment a^* for sentences
begin
  Initialization: Calculate the seed alignment a_0 orderly using simple model (IBM-1, IBM-2)
  Step 1: Optimize toward the constraints
  while $\{N_{ill}(a^*) > 0\}$ **do**
    if $\{a\colon N_{ill}(a) < N_{ill}(a^*)\} = \phi$ **then**
      break
    else
      $a^* = argmax_{a \in nb(a^*)} Pro(f|e, a)$
  Step 2:Toward the optimal alignment under the constraint
  for $\{i < N \text{ and } j < N\}$ **do**
    If (i,j)$\notin A^{**}$ Then, $M_{i,j}$ = -1
  while $\{M_{i_1,j_1} > 1 or S_{j_1,j_2} > 1\}$ **do**
    if $\{(j_1, a_{j_2}) \notin A^{**} or (j_2, a_{j_1}) \notin A^{**}\}$ **then**
      S_{j_1,j_2} = -1
    $M_{i_1,j_1} = argmax M_{i,j}$
    $S_{j_1,j_2} = argmax S_{i,j}$
    if $M_{i_1,j_1} > S_{j_1,j_2}$ **then**
      Update $M_{i_1,*}, M_{j_1,*}, M_{*,i_1}, M_{*,j_1}$
      Update $S_{i_1,*}, S_{j_1,*}, S_{*,i_1}, S_{*,j_1}$
      set $a^* := M_{i_1,j_1}(a)$
    else
      Update $M_{i_1,*}, M_{j_2,*}, M_{*,i_1}, M_{*,j_2}$
      Update $S_{j_2,*}, S_{j_1,*}, S_{*,j_2}, S_{*,j_1}$
      set $a^* := S_{j_1,j_2}(a)$
    End
  return a^*

The initial state of the reviews is the initial alignment a_0 and is calculated using the simple models like *HMM, IBM-1 andIBM-2*. In the algorithm there are mainly two following steps:

1. Optimization of initial alignment towards the constraints.
2. Obtaining the Optimal alignment under constraints

The complete working of the algorithm is explained in Con-strained Hill Climbing Algorithm.

The two steps followed in the algorithm is explained below.

Step 1: Optimization of initial alignment towards the constraints: The main objective of this step is to obtain the alignment that is near to the constraints for the alignment model. In the algorithm nb() is used to represent alignments among neighboring elements and in the present alignment entire number of possible inconsistent alignments is represented by Nill(). First, the input to the algorithm a_0 that is the initial alignments obtained by sequentially training with simple models (*IBM-1, IBM-2,IBM-3 etc.,*). Second, inconsistent alignments are elimi-nated from the initials alignments obtained using MOVE operator $M_{i,j}$ and *SWAP* operator s_{j_1,j_2}. Third, the align-ment is updated iteratively until no further inconsistent alignments can be eliminated.

Step 2: Obtaining the Optimal alignment under con-straints: In this step we aim to obtain the optimized alignments under constraints $a*$ using the provided list of partial alignment links $A**$. This step uses the initial alignments for beginning the optimization. Invalid *MOVE* and *SWAP* operations are assigned with -1. By this the probability of final alignments being consistent is high compared to the existing partial alignments.

Semantic Relations

Relations that exist among the homogeneous words are called as semantic relations. For example, consider the words *LCD* and *LED*. Both of them denote the same attribute *"screen"* in the domain of *"Television"*. These words are semantically related to each other. Hence besides opinion relations, semantic relations provide additional clues for indicating opinion target/words. Let us consider an example of three reviews as given below:

1. "For me the cost of this phone is expensive."
2. "Cost of it is good."
3. "Nokia XX has good price value."

Here in all three reviews, the attribute *"price"* or *"cost"* is modified by the word *"good"* maximum number of times than the word "expensive". We know that the word "expensive" is more related to word *"price"* or *"cost"* than the word "good". As the word "good" has more co-occurrence frequency and hence strong opinion relation exists between *"good"* and "*price*" than "*expensive*" and "*price*". This problem can be solved by using semantic relations.

Semantic relations is represented as a graph *G = (V, E,W)* with the set of vertices $V = V_0 \cup V_t$, where $v_t \in V_t$ is the set of opinion target candidates. Similarly the vertices set V_o is represented by $v_o \in V_o$ is the set of opinion word candidates. E is the set of edges where $e_{ij} \in E$ is the edge that connects the two vertices of same set. It means that the edge e_{ij} exists only between the vertices of either between opinion targets or between opinion words. Fig. 5 represents both opinion and semantic relations. The dashed lines show the opinion relations represented by G_{to} and the semantic relations are represented by solid lines and it is represented by G_{tt} and G_{oo}. Here, G_{tt} represents the relation between opinion targets and G_{oo} represents the relations among the opinion words. Random walk algorithm [23] is performed on G_{tt}, G_{oo} and G_{to} separately to estimate all candidate's confidence. Candidates with higher confidence than a threshold are correspondingly extracted as opinion target/words. The result may reflect what type of relation is more useful for the extraction.

Implementation

In this section, we discuss in detail the implementation procedure for extraction of opinion targets/words based on:

1. Only using opinion relations
2. Only using semantic relations
3. Using both opinion and semantic relations

The process of extracting the opinion targets and opinion words is represented in the flow chart in Fig. 6. Description of the steps in the flow chart are as follows:

Step 1: The process begins with the collection of reviews on desired products. *Customer Review* dataset is used as the review set for the implementation. Word Tokenizer is used to form the list of words. The word that occurs more frequently has a high term frequency and it is more likely to be an opinion target (*OT*).

Step 2: There are some words that are not domain specific, still these words occurs more frequently. Few examples of such words are *'mood'*, *'time'*, *'feelings'*, *etc.,* but these words are not likely to be *OT*. These words are categorized as general nouns. A list of general nouns are collected. The nouns and adjectives list that were extracted from the reviews in the Step 1 are compared with the general noun list. The words that match the comparison are eliminated from the list of words [24].

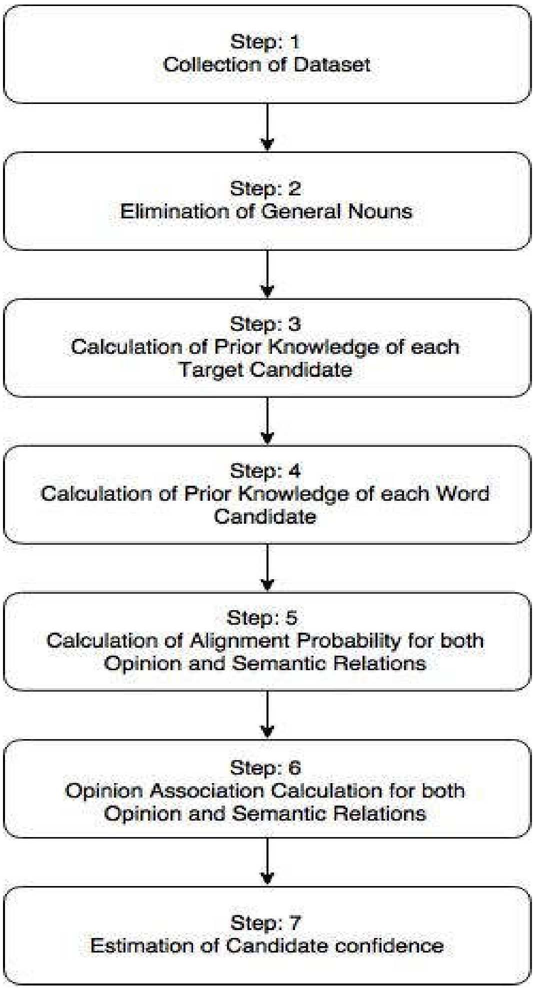


Fig 6 Process of Extraction of *OT* and *OW*

Step 3: In this step, the prior knowledge of each opinion target $prior_t$ is calculated. It requires the term frequency (*TF*) and inverse document frequency (*IDF*) of each target candidate. Term frequency is the number of times a particular word x has been appeared in a document or a review. Let d represent a document or a review. Using *TF* value, *IDF* value and *TF-IDF* score of word x is calculated using Equations (4) and (5) respectively.

$$IDF = log\frac{(No.of\ d)}{(No.\ of\ d\ containing\ word\ x)} \tag{4}$$

$$TF - IDF_{score} = TF.IDF \tag{5}$$

This obtained *TF IDF* score is converted to prior knowledge by normalizing the scores. Using the similar technique, prior knowledge of all targets are calculated.

Step 4: Prior knowledge of each opinion words $prior_o$ is calculated. This is carried out by using the *SentiWordNet* [25]. The words in *Senti WordNet* are collected along with the positive score and negative score of it. By adding these scores we get the subjective scores. The subjective scores are finally used as the prior knowledge for the opinion word candidates.

Step 5: Calculation of alignment probability is different for opinion relations and semantic relations and it is described below.

Opinion Relations: Alignment probability between opinion targets (x_t) and opinion word x_o is $P(x_t|x_o)$and it is calculated using the Equation (6). Similarly by changing the alignment direction, alignment probability between opinion word x_o and opinion target x_t is $P(x_o|x_t)$ and it is calculated using the Equation

$$P(x_t|x_o) = \frac{Cou\ n(x_t,x_o)}{Cou\ n(x_o)} \tag{6}$$

$$P(x_o|x_t) = \frac{Cou\ n(x_t,x_o)}{Cou\ n(x_t)} \tag{7}$$

$P(x_t|x_o)$and$P(x_o|x_t)$are values that arevery key factors in the calculation of the opinion association among the *OT* and *OW* candidates. Here, $Cou\ n(x_t|x_o)$ is the total count of the *OT* and *OW* pair frequency.

* Semantic Relations: Alignment probability between opinion targets x_{t1} and $x_{t2}$is $P(x_{t1}|x_{t2})$and it is calculated using the Equation (8). Similarly, alignment probability between opinion word x_{o1} and x_{o2} is $P(x_{o1}|x_{o2})$ and it is calculated using the Equation (9).

$$P(x_{t1}|x_{t2}) = \frac{Cou\ n(x_{t1},x_{t2})}{Cou\ n(x_{t2})} \tag{8}$$

$$P(x_{o1}|x_{o2}) = \frac{Cou\ n(x_{o1},x_{o2})}{Cou\ n(x_{o2})} \tag{9}$$

$P(x_{t1}|x_{t2})$ and $P(x_{o1}|x_{o2})$ are values

that are used for the calculation of the opinion association within the *OT* and *OW* candidates. Here, $Count(x_{t1}|x_{t2})$ and $Count(x_{o1}|x_{o2})$ is the total count of the *OT* pair and *OW* pair frequency. Step 6: In this step, Opinion Association (*OA*) is calculated for opinion and semantic relations.

* Opinion Relations: The opinion association is calculated between each *OT* and *OW* pairs. It is calculated using Equation (10).

$$OA = (\propto * P(x_t|x_o) + (1-\propto)P(x_o|x_t))^{-1} \tag{10}$$

Here, the harmonic factor '∝' is the factor used to combine the probability values obtained in Equations (6) and (7). In our work the harmonic factor value is set to 0.5.

* Semantic Relations: Opinion Association (OA_{tt}) of opinion target pairs and (OA_{oo}) of opinion word pairs are calculated using the Equations (11) and (12) respectively.

$$OA_{tt} = (\propto * P(x_{t1}|x_{t2}) + (1-\propto)P(x_{t1}|x_{t2}))^{-1} \tag{11}$$

$$OA_{oo} = (\propto * P(x_{o1}|x_{o2}) + (1-\propto)P(x_{o1}|x_{o2}))^{-1} \tag{12}$$

Step 7: In this step, we are going to estimate the candidate confidence by using Random Walk algorithm. Here, confidence values are assigned to the obtained *OT* and *OW*. The candidates with higher values of confidence than the threshold are likely to be *OT* or *OW*. Estimation process of candidate confidence value is split into three different ways. They are:

1. Only using opinion relations
2. Only using semantic relations
3. Using both opinion and semantic relations

Here we describe the implementation of all the above three methods.

Only using opinion relations: The confidence values of opinion target and opinion word are calculated by using the Equations (13) and (14) respectively.

$$Confidence_t^{k+1} = (1-\mu) \times OA_{to} \times Confidence_o^k + \mu \times prior_t \quad (13)$$

$$Confidence_0^{k+1} = (1-\mu) \times OA_{to}^T \times Confidence_t^k + \mu \times prior_o \quad (14)$$

$Confidence_t^{k+1}$ and $Confidence_0^{k+1}$ are the confidences of target and word respectively in $(k + 1)^{th}$ iteration. Similarly, $Confidence_t^k$ and $Confidence_o^k$ are confidences of target and word candidates in $(k)^{th}$ iteration. OA_{to} is the opinion association of a target and word pair. $prior_t$ and $prior_o$ denote prior knowledge of candidates being opinion targets and opinion words respectively. $\mu \in [0,1]$ represents the impact of prior knowledge in the final results.

It is noticed that the confidence values are calculated in two separate parts. In the first part of the equation,

$OA_{to} \times Confidence_o^k$ and $OA_{to}^T \times Confidence_t^k$ shows the influence of confidence values of neighboring candidates. Hence it is clear that the confidence of a candidate is the aggregate of confidences of all neighboring *OT* and *OW* respectively. The second part of the Equations(13) and (14), shows the importance of prior knowledge in the process of confidence calculation. They are $\mu \times prior_t$ and $\mu \times prior_o$. Here, $prior_t$ and $prior_o$ are the prior knowledge of target and word respectively. The value of μ decides the final result. If the value of μ is 1 then the final result or confidence value completely depends on the prior knowledge of the candidates as the first part becomes 0. If μ takes the value of 0, then the result or confidence value completely depends on the prior knowledge of the candidates as the first part becomes 0. If μ takes the value of 0 then the confidence is determined by the confidences of neighboring candidates and also the opinion association of the *OT* and *OW* pair.

Only using Semantic relations: The confidence values of opinion target pair and opinion word pair are calculated by using the Equations (15) and (16) respectively.

$$\mathrm{Confidence}_t^{k+1} = (1-\mu) \times \mathrm{OA}_{tt} \times \mathrm{Confidence}_o^k + \mu \times \mathrm{prior}_t \quad (15)$$

$$\mathrm{Confidence}_0^{k+1} = (1-\mu) \times \mathrm{OA}_{oo}^T \times \mathrm{Confidence}_t^k + \mu \times \mathrm{prior}_o \quad (16)$$

$\mathrm{Confidence}_t^{k+1}$ and $\mathrm{Confidence}_0^{k+1}$ are the confidencesof target and word respectively in $(k+1)^{th}$ iteration. As in semantic relations relation found between opinion targets or opinion words OA_{tt} and OA_{oo} are used in the calculation of the $\mathrm{Confidence}_t^{k+1}$ and $\mathrm{Confidence}_0^{k+1}$ respectively.

Using both opinion and semantic relations: The confidence values of the obtained OT and OW are calculated by considering both the Opinion relations and the semantic relations using Equations (17) and (18)

$$\mathrm{Confidence}_t^{k+1} = (1 - \lambda - \mu) \times \mathrm{OA}_{tt} \times \mathrm{Confidence}_o^k + \lambda \times \mathrm{OA}_{tt} \times \mathrm{Confidence}_t^k + \mu \times \mathrm{prior}_t \quad (17)$$

$$\mathrm{Confidence}_t^{k+1} = (1 - \lambda - \mu) \times \mathrm{OA}_{oo} \times \mathrm{Confidence}_o^k + \lambda \times \mathrm{OA}_{oo} \times \mathrm{Confidence}_o^k + \mu \times \mathrm{prior}_o \quad (18)$$

In the Equation (17) and (18), the value of λ decides whether the confidence of the candidate is determined by opinion or semantic relations. If the value of λ is 0, then the candidate confidence is calculated by considering only the opinion relations. If the value of λ is 1, then theconfidence value is completely determined by considering only the semantic relations.

Experimental results and performance Analysis

We have selected Customer Review Datasets (CRD) [13] to evaluate our approach. CRD includes reviews in English. Reviews are given on five products namely, *apexdvd, canon camera, nikon camera, nokia mobile and creative audio player*. These Products are named as *D1, D2, D3, D4* and *D5* respectively. Annotators are required to judge whether every noun/noun phrase (adjectives/verbs) is an opinion target (opinion word) or not. Table I contains the information about the dataset CRD where *OW* and *OT* stand for the number of opinion word and opinion target respectively.

Table I Details of Customer Review Dataset (CRD)

Domain	#Sentences	# OW	# OT
D1: apex DVD	597	175	109
D2: canon Camera	346	182	98
D3:nikon Camera	546	261	177
D4: nokia Mobile	1714	138	73
D5: creative audio player	740	164	103

List of sentences from each review can be formed by using sentence tokenizer. Similarly, list of words can be formed by using word tokenizer with part of speech tagging (by using Stanford NLP tool [26]). Precision(P), Recall(R) and F-measure(F) are selected as the evaluation metrics. We assign μ = 0.5 that indicate the impact of the prior knowledge while calculating the confidence value.

Table II, Table III, Table IV represents the comparison of F-measure, Precision and Recall values respectively of opinion target extraction using *WAM*, $PSWAM_{OR}$ and $PSWAM_{ORandSR}$ methods. Here *WAM* is a unsupervised alignment model, $PSWAM_{OR}$ indicates partially supervised *WAM* that considers only opinion relation and $PSWAM_{ORandSR}$ is a partially supervised *WAM* that considers both opinion and semantic relation. Similarly Table V, Table VI and Table VII represent the comparison of F-measure, Precision and Recall values respectively of OW extraction using *WAM,* $PSWAM_{OR}$ and $PSWAM_{ORandSR}$ methods. We can observe that there is a significant improvement in the values of evaluation metrics in $PSWAM_{ORandSR}$ method compared to the *WAM* and $PSWAM_{OR}$ method.

Table II F-measure Values of Opinion Target Extracion

Methods	D1	D2	D3	D4	D5
WAM	0.85	0.86	0.89	0.83	0.88
$PSWAM_{OR}$	0.85	0.86	0.90	0.82	0.90
$PSWAM_{SR}$	0.78	0.80	0.81	0.75	0.81
$PSWAM_{ORandSR}$	0.88	0.87	0.91	0.83	0.91

Table III Precision Values of Opinion Target Extraction

Methods	D1	D2	D3	D4	D5
WAM	0.86	0.88	0.89	0.81	0.89
$PSWAM_{OR}$	0.87	0.89	0.90	0.82	0.92
$PSWAM_{SR}$	0.72	0.81	0.81	0.76	0.85
$PSWAM_{ORandSR}$	0.88	0.91	0.91	0.85	0.93

Table IV Recall Values of Opinion Target Extraction

Methods	D1	D2	D3	D4	D5
WAM	0.85	0.85	0.89	0.83	0.87
$PSWAM_{OR}$	0.84	0.84	0.90	0.83	0.88
$PSWAM_{SR}$	0.76	0.79	0.83	0.75	0.81
$PSWAM_{ORandSR}$	0.86	0.86	0.92	0.84	0.90

Table V F-measure Values of Opinion Word Extraction

Methods	D1	D2	D3	D4	D5
WAM	0.68	0.66	0.69	0.66	0.70
$PSWAM_{OR}$	0.70	0.68	0.71	0.67	0.71
$PSWAM_{SR}$	0.61	0.61	0.66	0.60	0.63
$PSWAM_{ORandSR}$	0.72	0.70	0.72	0.69	0.73

Table VI Precision Values of Opinion Word Extraction

Methods	D1	D2	D3	D4	D5
WAM	0.62	0.57	0.63	0.62	0.70
$PSWAM_{OR}$	0.65	0.59	0.66	0.64	0.72
$PSWAM_{SR}$	0.55	0.50	0.59	0.60	0.66
$PSWAM_{ORandSR}$	0.68	0.63	0.70	0.66	0.74

Table VII Recall Values of Opinion Word Extraction

Methods	D1	D2	D3	D4	D5
WAM	0.76	o.79	0.77	0.71	0.71
$PSWAM_{OR}$	0.76	0.80	0.78	0.70	0.71
$PSWAM_{SR}$	0.67	0.70	0.61	0.63	0.62
$PSWAM_{ORandSR}$	0.78	0.81	0.79	0.72	0.72

Few experimental observations made are as follows Result set in Table II contains F-measure values of opinion target extraction. *$PSWAM_{ORandSR}$* and *$PSWAM_{OR}$* outperforms *WAM* methods in all five different domains. This is due to the fact that*$PSWAM_{OR}$* and *$PSWAM_{ORandSR}$* extract the opinion relations by using the word alignment model under partial supervision, whereas *WAM* is completely unsupervised.

$PSWAM_{ORandSR}$ method results are better compared to *$PSWAM_{OR}$* method as in *$PSWAM_{ORandSR}$* method, Semantic Relations are also considered along with opinion relations. This results in extraction of more number of opinion targets and opinion words. It is observed that the *F-measure* value of *$PSWAM_{ORandSR}$* method increases by 3% for domain *D1*, 1% for *D2, D3, D4* and *D5* compared to *$PSWAM_{OR}$* method. Similar performance is seen in opinion word extraction as shown in Table V. In the extraction of opinion word, *F-measure* value of *$PSWAM_{ORandSR}$* method increases by 2% for domain *D1, D2, D4, D5* and 1% for *D3* compared to *PSWAM* method. *Precision* and *Recall* values are tabulated in Table III and Table IV respectively for opinion target extraction. Even the *Precision* and *Recall* values are listed in Table VI and Table VII respectively for opinion word extraction. These values also indicate that the *$PSWAM_{ORandSR}$* extract more precise relations in all domains compared to *WAM*, *$PSWAM_{OR}$* and *$PSWAM_{SR}$*

Method *$PSWAM_{ORandSR}$* outperforms *$PSWAM_{OR}$* as semantic relations are considered along with the opinion relations that exists among words. By considering semantic relations, there is increased number of relations captured among words resulting in more accurate *OT* and *OW* extraction. Fig. 7 gives the comparison of *F-measure* values of *$PSWAM_{OR}$*, *$PSWAM_{SR}$* and *$PSWAM_{ORandSR}$* methods for the extraction of opinion target candidate. It indicates that considering only opinion relations perform better compared to considering only semantic relations. Similar performance is observed for all domains (*D1, D2, D3, D4* and *D5*). Identical results are observed in Fig. 8 and Fig. 9 that represents precision and recall values respectively for the extraction of opinion target. Fig. 10, Fig. 11 and Fig. 12 also represents the significant performance of *$PSWAM_{ORandSR}$* method compared to *$PSWAM_{OR}$* and *$PSWAM_{SR}$* method. Here Fig. 10, Fig. 11 and Fig. 12 represent the F-measure, precision and recall values respectively of opinion word extraction.

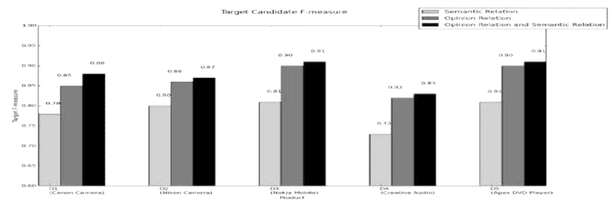


Fig 7 F-measure Values for the Extraction of Opinion Target

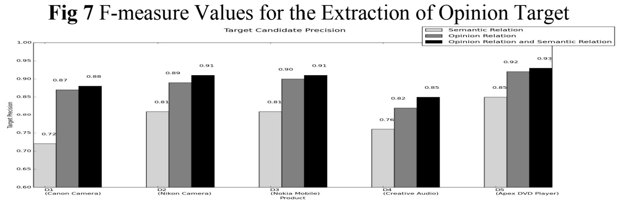


Fig 8 Precision Values for the Extraction of Opinion Target

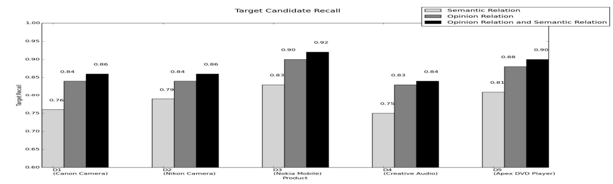


Fig 9 Recall Values for the Extraction of Opinion Target

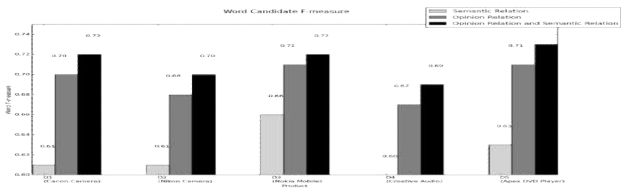


Fig 10 F-measure Values for the Extraction of Opinion Word

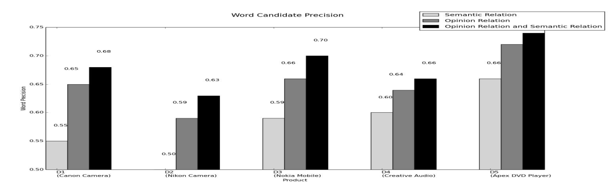


Fig. 11 Precision Values for the Extraction of Opinion Word

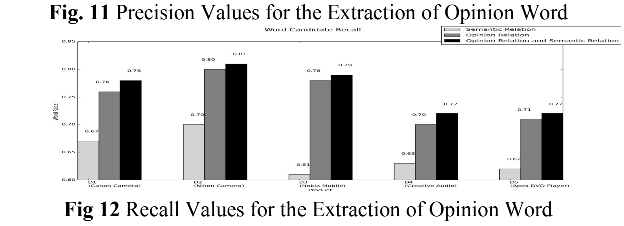


Fig 12 Recall Values for the Extraction of Opinion Word

CONCLUSIONS

Extraction of opinion targets and opinion words play an important role in finding the sentiment of the product review. Partially Supervised Word Alignment Model (*PSWAM*) is used to extract opinion relations among the *OT* and *OW*. Graph based co-ranking algorithm is used to extract the *OT* and *OW* using both relations. Using opinion and semantic relations together, the effectiveness of the opinion target/word mining enhances. This results in considerable improvement in the values of *precision*, *recall* and *F-measure*. Results depicts that opinion target/word extraction is more effective when opinion and semantic relations are considered together rather than considering each relation separately.

References

1. G. Qiu, B. Liu, J. Bu, and C. Chen, "Expanding Domain Sentiment Lexicon through Double Propagation," in *IJCAI*, vol. 9, pp. 1199-1204, 2009.
2. A. Popescu and O. Etzioni, "Extracting Product Features and Opinions from Reviews," *in Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing. Association for Computational Linguistics*, pp. 339-346, 2005.
3. G. Qiu, B. Liu, J. Bu, and C. Chen, "Opinion Word Expansion and Tar-get Extraction through Double Propagation," *Computational Linguistics*, vol. 37, no. 1, pp. 9–27, 2011.
4. S. Mathapati, S. H. Manjula, and K. R. Venugopal, "Sentiment Analysis and Opinion Mining from Social Media: A Review," *Global Journal of Computer Science and Technology*, vol. 16, no. 5, pp. 77-90, 2017
5. K. Liu, L. Xu, and J. Zhao, "Co-extracting Opinion Targets and Opinion Words from Online Reviews based on the Word Alignment Model," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 3, pp. 636–650, 2015.
6. F. Li, C. Han, M. Huang, X. Zhu, Y.-J. Xia, S. Zhang, and H. Yu, "Structure-aware Review Mining and Summarization," *in Proceedings of the 23rd international conference on computational linguistics*, pp. 653-661, 2010.
7. Y. Wu, Q. Zhang, X. Huang, and L. Wu, "Phrase Dependency Parsing for Opinion Mining," *in Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing,* vol. 3. Association for Computational Linguistics, pp. 1533-1541, 2009.
8. T. Ma and X. Wan, "Opinion Target Extraction in Chinese News Comments," *in Proceedings of the 23rd International Conference on Computational Linguistics: Posters. Association for Computational Linguistics*, pp. 782–790, 2010.
9. Q. Zhang, Y. Wu, T. Li, M. Ogihara, J. Johnson, and X. Huang, "Mining Product Reviews based on Shallow Dependency Parsing," *in Proceedings of the 32rd international ACM SIGIR conference on Research and development in information retrieval*, ACM, pp. 726-727, 2009.
10. W. Jin, H. H. Ho, and R. K. Srihari, "A Novel Lexicalized HMM-based Learning Framework for Web Opinion Mining," *in Proceedings of the 26th annual international conference on machine learning,* pp. 465-472, 2009.
11. F. Li, S. J. Pan, O. Jin, Q. Yang, and X. Zhu, "Cross-domain Co-extraction of Sentiment and Topic Lexicons," *in Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-* Vol. 1, , pp. 410-419, 2012.
12. B. Wang and H. Wang, "Bootstrapping Both Product Features and Opinion Words from Chinese Customer Reviews with Cross-inducing." in *IJCNLP*, vol. 8, pp. 289-295, 2008
13. 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*, pp. 168-177, 2004.
14. L. Zhang, B. Liu, S. H. Lim, and E. O'Brien-Strain, "Extracting and Ranking Product Features in Opinion Documents," *in Proceedings of the 23rd International Conference on Computational Linguistics: Posters. Association for Computational Linguistics*, pp. 1462-1470, 2010.
15. J. M. Kleinberg, "Authoritative Sources in a Hyperlinked Environment," *in Proceedings of the ACM-SIAM Symposium on Discrete Algorithms. Citeseer*, 1998.
16. K. Liu, L. Xu, and J. Zhao, "Opinion Target Extraction using Word-based Translation Model," *in Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. Association for Computational Linguistics,* pp. 1346-1356, 2012.
17. Q. Mei, X. Ling, M. Wondra, H. Su, and C. Zhai, "Topic Sentiment Mixture: Modeling Facets and Opinions in Weblogs," *in Proceedings of the 16th international conference on World Wide Web. ACM*, pp. 171-180, 2007.
18. I. Titov and R. T. McDonald, "A Joint Model of Text and Aspect Ratings for Sentiment Summarization." *in ACL*, vol. 8, pp. 308-316, 2008.
19. K. R. Venugopal, K. G. Srinivasa, and L. M. Patnaik, *Soft Computing for Data Mining Applications. Springer*, 2009.
20. S. Joshi, P. D. Shenoy, K. R. Venugopal, and L. M. Patnaik, "Classification of Neurodegenerative Disorders based on Major Risk Factors Employing Machine Learning Techniques," *International Journal of Engineering and Technology,* vol. 2, no. 4, pp. 350-356, 2010.
21. P. F. Brown, V. J. D. Pietra, S. A. D. Pietra, and R. L. Mercer, "The Mathematics of Statistical Machine Translation: Parameter Estimation," *Computational Linguistics,* vol. 19, no. 2, pp. 263-311, 1993.
22. "http://fabioticconi.wordpress.com/2017."
23. K. Liu, L. Xu, J. Zhao *et al*., "Extracting Opinion Targets and Opinion Words from Online Reviews with Graph Co-ranking." *in ACL* , 2014, pp. 314-324.
24. V. Jha, N. Manjunath, P. D. Shenoy, and K. R. Venugopal, "Hsra: Hindi Stopword Removal Algorithm," *in 2016 IEEE International Conference on Microelectronics, Computing and Communications (MicroCom)*, pp. 1-5, 2016.
25. "http://sentiwordnet.isti.cnr.it/."
26. C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky, "The Stanford Corenlp Natural Language Processing Toolkit." *in ACL (System Demonstrations)*, pp. 55-60, 2014