

Probabilistic Sense Sentiment Similarity through Hidden Emotions

Mitra Mohtarami¹, Man Lan², and Chew Lim Tan¹

¹Department of Computer Science, National University of Singapore;

²Department of Computer Science, East China Normal University

{mitra, tancl}@comp.nus.edu.sg;mlan@cs.ecnu.edu.cn

Abstract

Sentiment Similarity of word pairs reflects the distance between the words regarding their underlying sentiments. This paper aims to infer the sentiment similarity between word pairs with respect to their senses. To achieve this aim, we propose a probabilistic emotion-based approach that is built on a hidden emotional model. The model aims to predict a vector of basic human emotions for each sense of the words. The resultant emotional vectors are then employed to infer the sentiment similarity of word pairs. We apply the proposed approach to address two main NLP tasks, namely, *Indirect yes/no Question Answer Pairs* inference and *Sentiment Orientation* prediction. Extensive experiments demonstrate the effectiveness of the proposed approach.

1 Introduction

Sentiment similarity reflects the distance between words based on their underlying sentiments. Semantic similarity measures such as Latent Semantic Analysis (LSA) (Landauer et al., 1998) can effectively capture the similarity between semantically related words like "car" and "automobile", but they are less effective in relating words with similar sentiment orientation like "excellent" and "superior". For example, the following relations show the semantic similarity between some sentiment words computed by LSA:

$$\begin{aligned} \mathbf{E1}: \text{LSA}(\text{excellent}, \text{superior}) &= 0.40 \\ &< \text{LSA}(\text{excellent}, \text{good}) = 0.46 \\ &< \text{LSA}(\text{good}, \text{bad}) = 0.65 \end{aligned}$$

Clearly, the sentiment similarity between the above words should be in the reversed order. In fact, the sentiment intensity in "excellent" is closer to "superior" than "good". Furthermore,

sentiment similarity between "good" and "bad" should be 0.

In this paper, we propose a probabilistic approach to detect the sentiment similarity of words regarding their senses and underlying sentiments. For this purpose, we propose to model the hidden emotions of word senses. We show that our approach effectively outperforms the semantic similarity measures in two NLP tasks: *Indirect yes/no Question Answer Pairs (IQAPs) Inference* and *Sentiment Orientation (SO) prediction* that are described as follows:

In IQAPs, answers do not explicitly contain the *yes* or *no* keywords, but rather provide context information to infer the *yes* or *no* answer (e.g. **Q:** *Was she the **best** one on that old show?* **A:** *She was simply **funny**.*). Clearly, the sentiment words in IQAPs are the pivots to infer the *yes* or *no* answers. We show that sentiment similarity between such words (e.g., here the adjectives *best* and *Funny*) can be used effectively to infer the answers.

The second application (SO prediction) aims to determine the sentiment orientation of individual words. Previous research utilized the semantic relations between words obtained from WordNet (Hassan and Radev, 2010) and semantic similarity measures (e.g. Turney and Littman, 2003) for this purpose. In this paper, we show that sentiment similarity between word pairs can be effectively utilized to compute SO of words.

The contributions of this paper are follows:

- We propose an effective approach to predict the sentiment similarity between word pairs through hidden emotions at the sense level,
- We show the utility of sentiment similarity prediction in IQAP inference and SO prediction tasks, and
- Our hidden emotional model can infer the type and number of hidden emotions in a corpus.

2 Sentiment Similarity through Hidden Emotions

As we discussed above, semantic similarity measures are less effective to infer sentiment similarity between word pairs. In addition, different senses of sentiment words carry different human emotions. In fact, a sentiment word can be represented as a vector of emotions with *intensity* values from "very weak" to "very strong". For example, Table 1 shows several sentiment words and their corresponding emotion vectors based the following set of emotions: $e = [\textit{anger}, \textit{disgust}, \textit{sadness}, \textit{fear}, \textit{guilt}, \textit{interest}, \textit{joy}, \textit{shame}, \textit{surprise}]$. For example, "deceive" has 0.4 and 0.5 intensity values with respect to the emotions "disgust" and "sadness" with an overall -0.9 (i.e. $-0.4-0.5$) value for sentiment orientation (Neviarouskaya et al., 2007; Neviarouskaya et al., 2009).

Word	Emotional Vector	SO
$e = [\textit{anger}, \textit{disgust}, \textit{sadness}, \textit{fear}, \textit{guilt}, \textit{interest}, \textit{joy}, \textit{shame}, \textit{surprise}]$		
<i>Rude</i>	[0.2, 0.4, 0, 0, 0, 0, 0, 0]	-0.6
<i>doleful</i>	[0, 0, 0.4, 0, 0, 0, 0, 0]	-0.4
<i>smashed</i>	[0, 0, 0.8, 0.6, 0, 0, 0, 0]	-1.4
<i>shamefully</i>	[0, 0, 0, 0, 0, 0, 0.7, 0]	-0.7
<i>deceive</i>	[0, 0.4, 0.5, 0, 0, 0, 0, 0]	-0.9

Table 1. Sample of emotional vectors

The difficulty of the sentiment similarity prediction task is evident when terms carry different types of emotions. For instance, all the words in Table 1 have negative sentiment orientation, but, they carry different emotions with different emotion vectors. For example, "rude" reflects the emotions "anger" and "disgust", while the word "doleful" only reflects the emotion "sadness". As such, the word "doleful" is closer to the words "smashed" and "deceive" involving the emotion "sadness" than others. We show that emotion vectors of the words can be effectively utilized to predict the sentiment similarity between them.

Previous research shows little agreement about the number and types of the basic emotions (Ortony and Turner 1990; Izard 1971). Thus, we assume that the number and types of basic emotions are hidden and not pre-defined and propose a Probabilistic Sense Sentiment Similarity (PSSS) approach to extract the hidden emotions of word senses to infer their sentiment similarity.

3 Hidden Emotional Model

Online review portals provide rating mechanisms (in terms of stars, e.g. 5- or 10-star rating) to al-

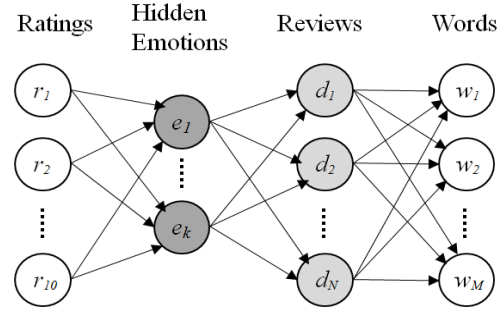


Figure 1. The structure of PSSS model

low users to attach ratings to their reviews. A rating indicates the summarized opinion of a user who ranks a product or service based on his feelings. There are various feelings and emotions behind such ratings with respect to the content of the reviews.

Figure 1 shows the intermediate layer of hidden emotions behind the ratings (sentiments) assigned to the documents (reviews) containing the words. This Figure indicates the general structure of our PSSS model. It shows that hidden emotions (e_i) link the rating (r_i) and the documents (d_k). In this Section, we aim to employ ratings and the relations among ratings, documents, and words to extract the hidden emotions.

Figure 2 illustrates a simple graphical model using plate representation of Figure 1. As Figure 2 shows, the rating r from a set of ratings $R = \{r_1, \dots, r_p\}$ is assigned to a hidden emotion set $E = \{e_1, \dots, e_k\}$. A document d from a set of documents $D = \{d_1, \dots, d_N\}$ with vocabulary set $W = \{w_1, \dots, w_M\}$ is associated with the hidden emotion set.

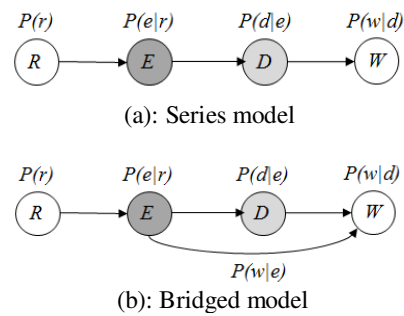


Figure 2. Hidden emotional model

The model presented in Figure 2(a) has been explored in (Mohtarami et al., 2013) and is called *Series Hidden Emotional Model* (SHEM). This representation assumes that *the word w is dependent to d and independent to e* (we refer to this Assumption as *A1*). However, in reality, a word w can inherit properties (e.g., emotions)

from the document d that contains w . Thus, we can assume that w is implicitly dependant on e . To account for this, we present *Bridged Hidden Emotional Model* (BHEM) shown in Figure 2(b). Our assumption, A2, in the BHEM model is as follows: w is dependent to both d and e .

Considering Figure 1, we represent the entire text collection as a set of (w,d,r) in which each observation (w,d,r) is associated with a set of unobserved emotions. If we assume that the observed tuples are independently generated, the whole data set is generated based on the joint probability of the observation tuples (w,d,r) as the follows (Mohtarami et al., 2013):

$$D = \prod_r \prod_d \prod_w P(w,d,r)^{n(w,d,r)} = \prod_r \prod_d \prod_w P(w,d,r)^{n(w,d)n(d,r)} \quad (1)$$

where, $P(w,d,r)$ is the joint probability of the tuple (w,d,r) , and $n(w,d,r)$ is the frequency of w in document d of rating r (note that $n(w,d)$ is the term frequency of w in d and $n(d,r)$ is one if r is assigned to d , and 0 otherwise). The joint probability for the BHEM is defined as follows considering hidden emotion e :

- regarding class probability of the hidden emotion e to be assigned to the observation (w,d,r) :

$$\begin{aligned} P(w,d,r) &= \sum_e P(w,d,r|e)P(e) = \\ &= \sum_e P(w,d|e)P(r|e)P(e) \\ - \text{regarding assumption A2 and Bayes' Rule:} \\ &= \sum_e P(w|d,e)P(d,e)P(r|e) \\ - \text{using Bayes' Rule:} \\ &= \sum_e P(d,e|w)P(w)P(r|e) \\ - \text{regarding A2 and conditional independency:} \\ &= \sum_e P(d|w)P(e|w)P(w)P(r|e) \\ &= P(d|w) \sum_e P(w|e)P(e)P(r|e) \end{aligned} \quad (2)$$

In the bridged model, the joint probability does not depend on the probability $P(d|e)$ and the probabilities $P(w|e)$, $P(e)$ and $P(r|e)$ are unknown, while in the SHEM model explained in (Mohtarami et al., 2013), the joint probability does not depend on $P(w|e)$, and probabilities $P(d|e)$, $P(e)$, and $P(r|e)$ are unknown.

We employ Maximum Likelihood approach to learn the probabilities and infer the possible hidden emotions. The log-likelihood of the whole

data set D in Equation (1) can be defined as follows:

$$L = \sum_r \sum_d \sum_w n(w,d)n(d,r) \log P(w,d,r) \quad (3)$$

Replacing $P(w,d,r)$ by the values computed using the bridged model in Equation (2) results in:

$$L = \sum_r \sum_d \sum_w n(w,d)n(d,r) \log [P(d|w) \sum_e P(w|e)P(e)P(r|e)] \quad (4)$$

The above optimization problems are hard to compute due to the log of sum. Thus, *Expectation-maximization* (EM) is usually employed. EM consists of two following steps:

1. **E-step**: Calculates posterior probabilities for hidden emotions given the words, documents and ratings, and
2. **M-step**: Updates unknown probabilities (such as $P(w|e)$ etc) using the posterior probabilities in the E-step.

The steps of EM can be computed for BHEM model. EM of the model employs assumptions A2 and Bayes Rule and is defined as follows:

E-step:

$$P(e|w,d,r) = \frac{P(r|e)P(e)P(w|e)}{\sum_e P(r|e)P(e)P(w|e)} \quad (5)$$

M-step:

$$\begin{aligned} P(r|e) &= \frac{\sum_d \sum_w n(w,d)n(d,r) P(e|w,d,r)}{\sum_r \sum_d \sum_w n(w,d)n(d,r) P(e|w,d,r)} \\ &= \frac{\sum_w n(w,r) P(e|w,d,r)}{\sum_r \sum_w n(w,r) P(e|w,d,r)} \end{aligned} \quad (6)$$

$$\begin{aligned} P(w|e) &= \frac{\sum_r \sum_d n(w,d)n(d,r) P(e|w,d,r)}{\sum_w \sum_r \sum_d n(w,d)n(d,r) P(e|w,d,r)} \\ &= \frac{\sum_r n(w,r) P(e|w,d,r)}{\sum_w \sum_r n(w,r) P(e|w,d,r)} \end{aligned} \quad (7)$$

$$\begin{aligned} P(e) &= \frac{\sum_r \sum_d \sum_w n(w,d)n(d,r) P(e|w,d,r)}{\sum_e \sum_d \sum_r \sum_w n(w,d)n(d,r) P(e|w,d,r)} \\ &= \frac{\sum_r \sum_w n(w,r) P(e|w,d,r)}{\sum_e \sum_r \sum_w n(w,r) P(e|w,d,r)} \end{aligned} \quad (8)$$

Note that in Equation (5), the probability $P(e|w,d,r)$ does not depend on the document d . Also, in Equations (6)-(8) we remove the dependency on document d using the following Equation:

$$\sum_d n(w,d)n(d,r) = n(w,r) \quad (9)$$

where $n(w,r)$ is the occurrence of w in all the documents in the rating r .

The EM steps computed by the bridged model do not depend on the variable document d , and discard d from the model. The reason is that w bypasses d to directly associate with the hidden emotion e in Figure 2(b).

Similar to BHEM, the EM steps for SHEM can be computed by considering assumptions *AI* and Bayes Rule as follows (Mohtarami et al., 2013):

E-step:

$$P(e|w, d, r) = \frac{P(r|e)P(e)P(d|e)}{\sum_e P(r|e)P(e)P(d|e)} \quad (10)$$

M-step:

$$P(r|e) = \frac{\sum_d \sum_w n(w, d)n(d, r)P(e|w, d, r)}{\sum_r \sum_d \sum_w n(w, d)n(d, r)P(e|w, d, r)} \quad (11)$$

$$P(d|e) = \frac{\sum_r \sum_w n(w, d)n(d, r)P(e|w, d, r)}{\sum_d \sum_r \sum_w n(w, d)n(d, r)P(e|w, d, r)} \quad (12)$$

$$P(e) = \frac{\sum_r \sum_d \sum_w n(w, d)n(d, r)P(e|w, d, r)}{\sum_e \sum_d \sum_r \sum_w n(w, d)n(d, r)P(e|w, d, r)} \quad (13)$$

Finally, we construct the emotional vectors using the algorithm presented in Table 2. The algorithm employs document-rating, term-document and term-rating matrices to infer the unknown probabilities. This algorithm can be used with both bridged or series models. Our goal is to infer the emotional vector for each word w that can be obtained by the probability $P(w|e)$. Note that, this probability can be simply computed for the SHEM model using $P(d|e)$ as follows:

$$P(w|e) = \sum_d P(w|d)P(d|e) \quad (14)$$

3.1 Enriching Hidden Emotional Models

We enrich our emotional model by employing the requirement that the emotional vectors of two synonym words w_1 and w_2 should be similar. For this purpose, we utilize the semantic similarity between each two words and create an enriched matrix. Equation (15) shows how we compute this matrix. To compute the semantic similarity between word senses, we utilize their synsets as follows:

$$\begin{aligned} w_i w_j &= P(\text{syn}(w_i) | \text{syn}(w_j)) \\ &= \frac{1}{|\text{syn}(w_i)|} \sum_{\text{syn}(w_i)} \frac{1}{|\text{syn}(w_j)|} \sum_{\text{syn}(w_j)} P(w_i | w_j) \end{aligned} \quad (15)$$

where, $\text{syn}(w)$ is the synset of w . Let $\text{count}(w_i, w_j)$ be the co-occurrence of the w_i and w_j , and let $\text{count}(w_j)$ be the total word count. The probability of w_i given w_j will then be $P(w_i | w_j) = \text{count}(w_i, w_j) / \text{count}(w_j)$. In addition, note that employing the synset of the words help to obtain different emotional vectors for each sense of a word.

The resultant enriched matrix $W \times W$ is multiplied to the inputs of our hidden model (matrices $W \times D$ or $W \times R$). Note that this takes into account

Input:

Series Model: Document-Rate $D \times R$, Term-Document $W \times D$
Bridged Model: Term-Rate $W \times R$

Output: Emotional vectors $\{e_1, e_2, \dots, e_k\}$ for w

Algorithm:

1. Enriching hidden emotional model:

Series Model: Update Term-Document $W \times D$
Bridged Model: Update Term-Rate $W \times R$

2. Initialize unknown probabilities:

Series Model: Initialize $P(d|e)$, $P(r|e)$, and $P(e)$, randomly
Bridged Model: Initialize $P(w|e)$, $P(r|e)$, and $P(e)$

3. **while** L has not converged to a pre-specified value **do**

4. E-step;

Series Model: estimate the value of $P(e|w, d, r)$ in Equation 10
Bridged Model: estimate the value of $P(e|w, d, r)$ in Equation 5

5. M-step;

Series Model: estimate the values of $P(r|e)$, $P(d|e)$, and $P(e)$ in Equations 11-13, respectively
Bridged Model: estimate the values of $P(r|e)$, $P(w|e)$, and $P(e)$ in Equations 6-8, respectively

6. **end while**

7. **If** series hidden emotional model is used then

8. Infer word emotional vector: estimate $P(w|e)$ in Equation 14.

9. **End if**

Table 2. Constructing emotional vectors via $P(w|e)$

the senses of the words as well. The learning step of EM is done using the updated inputs. In this case, the correlated words can inherit the properties of each other. For example, if w_i does not occur in a document or rating involving another word (i.e., w_j), the word w_i can still be indirectly associated with the document or rating through the word w_j . However, the distribution of the opinion words in documents and ratings is not uniform. This may decrease the effectiveness of the enriched matrix.

The nonuniform distribution of opinion words has been also reported by Amiri et al. (2012) who showed that the positive words are frequently used in negative reviews. We also observed the same pattern in the development dataset. Figure 3 shows the overall occurrence of some positive and negative seeds in various ratings. As shown, in spite of the negative words, the positive words may frequently occur in both positive and negative documents. Such distribution of

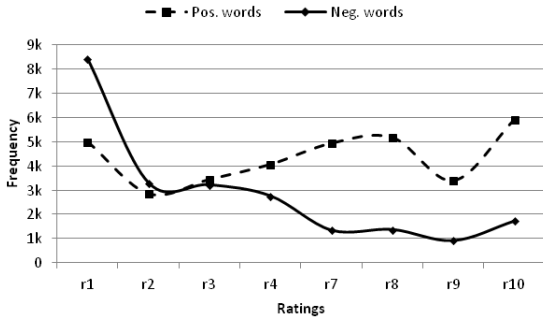


Figure 3. Nonuniform distribution of opinion words

positive words can mislead the enriched model.

To address this issue, we measure the confidence of an opinion word in the enriched matrix as follows.

$$Confidence_w = \frac{ABS[(TF_w^- \times DF_w^-) - (TF_w^+ \times DF_w^+)]}{(TF_w^- \times DF_w^-) + (TF_w^+ \times DF_w^+)} \quad (16)$$

where, TF_w^- (TF_w^+) is the frequency of w in the ratings 1 to 4 (7 to 10), and DF_w^- (DF_w^+) is the total number of documents with rating 1 to 4 (7 to 10) that contain w . The confidence value of w varies from 0 to 1, and it increases if:

- There is a large difference between the occurrences of w in positive and negative ratings.
- There is a large number of reviews involving w in the relative ratings.

To improve the efficiency of enriched matrix, the columns corresponding to each word in the matrix are multiplied by its confidence value.

4 Predicting Sentiment Similarity

We utilize the approach proposed in (Mohtarami et al., 2013) to compute the sentiment similarity between two words. This approach compares the emotional vector of the given words. Let X and Y be the emotional vectors of two words. Equation (17) computes their correlation:

$$corr(X, Y) = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{(n-1)S_X S_Y} \quad (17)$$

where, n is number of emotional categories, \bar{X} , \bar{Y} and S_X , S_Y are the mean and standard deviation values of X and Y respectively. $corr(X, Y) = -1$ indicates that the two vectors are completely dissimilar, and $corr(X, Y) = 1$ indicates that the vectors have perfect similarity.

The approach makes use of a thresholding mechanism to estimate the proper correlation value to find sentimentally similar words. For this, as in Mohtarami et al. (2013) we utilized the antonyms of the words. We consider two words,

Input:

SAQ : The adjective in the question of given IQAP.
 SAA : The adjective in the answer of given IQAP.

Output: answer $\in \{yes, no, uncertain\}$

Algorithm:

1. if SAQ or SAA are missing from our corpus then
2. answer=*Uncertain*;
3. else if $SS(SAQ, SAA) < 0$ then
4. answer=*No*;
5. else if $SS(SAQ, SAA) > 0$ then
6. answer=*yes*;

Figure 4. Sentiment similarity for IQAP inference

w_i and w_j as similar in sentiment iff they satisfy both of the following conditions:

1. $corr(w_i, w_j) > corr(w_i, \sim w_j)$, and
2. $corr(w_i, w_j) > corr(\sim w_i, w_j)$

where, $\sim w_i$ is antonym of w_i , and $corr(w_i, w_j)$ is obtained from Equation (17). Finally, we compute the sentiment similarity (SS) as follows:

$$SS(w_i, w_j) = corr(w_i, w_j) - Max\{corr(w_i, \sim w_j), corr(\sim w_i, w_j)\} \quad (18)$$

Equation (18) enforces two sentimentally similar words to have weak correlation to the antonym of each others. A positive value of $SS(...)$ indicates the words are sentimentally similar and a negative value shows that they are dissimilar.

5 Applications

We explain our approach in utilizing sentiment similarity between words to perform IQAP inference and SO prediction tasks respectively.

In IQAPs, we employ the sentiment similarity between the adjectives in questions and answers to interpret the indirect answers. Figure 4 shows the algorithm for this purpose. $SS(...)$ indicates sentiment similarity computed by Equation (18). A positive SS means the words are sentimentally similar and thus the answer is *yes*. However, negative SS leads to a *no* response.

In SO-prediction task, we aim to compute more accurate SO using our sentiment similarity method. Turney and Littman (2003) proposed a method in which the SO of a word is calculated based on its semantic similarity with seven positive words minus its similarity with seven negative words as shown in Figure 5. As the similarity function, $A(...)$, they employed point-wise mutual information (PMI) to compute the similarity between the words. Here, we utilize the same approach, but instead of PMI we use our $SS(...)$ measure as the similarity function.

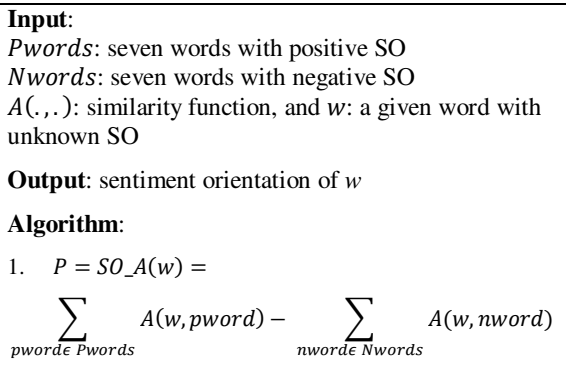


Figure 5. SO based on the similarity function *A*(.,.)

6 Evaluation and Results

6.1 Data and Settings

We used the review dataset employed by Maas et al. (2011) as the development dataset that contains movie reviews with star rating from one star (most negative) to 10 stars (most positive). We exclude the ratings 5 and 6 that are more neutral. We used this dataset to compute all the input matrices in Table 2 as well as the enriched matrix. The development dataset contains 50k movie reviews and 90k vocabulary.

We also used two datasets for the evaluation purpose: the MPQA (Wilson et al., 2005) and IQAPs (Marneffe et al., 2010) datasets. The MPQA dataset is used for SO prediction experiments, while the IQAP dataset is used for the IQAP experiments. We ignored the neutral words in MPQA dataset and used the remaining 4k opinion words. Also, the IQAPs dataset (Marneffe et al., 2010) contains 125 IQAPs and their corresponding *yes* or *no* labels as the ground truth.

6.2 Experimental Results

To evaluate our PSSS model, we perform experiments on the SO prediction and IQAPs inference tasks. Here, we consider six emotions for both bridged and series models. We study the effect of emotion numbers in Section 7.1. Also, we set a threshold of 0.3 for the confidence value in Equation (16), i.e. we set the confidence values smaller than the threshold to 0. We explain the effect of this parameter in Section 7.3.

Evaluation of SO Prediction

We evaluate the performance of our PSSS models in the SO prediction task using the algorithm explained in Figure 5 by setting our PSSS as similarity function (*A*). The results on SO prediction are presented in Table 3. The first and se-

Method	Precision	Recall	F1
PMI	56.20	56.36	55.01
ER	65.68	65.68	63.27
PSSS-SHEM	68.51	69.19	67.96
PSSS-BHEM	69.39	70.07	68.68

Table 3. Performance on SO prediction task

cond rows present the results of our baselines, PMI (Turney and Littman, 2003) and Expected Rating (ER) (Potts, 2011) of words respectively.

PMI extracts the semantic similarity between words using their co-occurrences. As Table 3 shows, it leads to poor performance. This is mainly due to the relatively small size of the development dataset which affects the quality of the co-occurrence information used by the PMI.

ER computes the expected rating of a word based on the distribution of the word across rating categories. The value of ER indicates the SO of the word. As shown in the two last rows of the table, the results of PSSS approach are higher than PMI and ER. The reason is that PSSS is based on the combination between sentiment space (through using ratings, and matrices $W \times R$ in BHEM, $D \times R$ in SHEM) and semantic space (through the input $W \times D$ in SHEM and enriched matrix $W \times W$ in both hidden models). However, the PMI employs only the semantic space (i.e., the co-occurrence of the words) and ER uses occurrence of the words in rating categories.

Furthermore, the PSSS model achieves higher performance with BHEM rather than SHEM. This is because the emotional vectors of the words are directly computed from the EM steps of BHEM. However, the emotional vectors of SHEM are computed after finishing the EM steps using Equation (14). This causes the SHEM model to estimate the number and type of the hidden emotions with a lower performance as compared to BHEM, although the performances of SHEM and BHEM are comparable as explained in Section 7.1.

Evaluation of IQAPs Inference

To apply our PSSS on IQAPs inference task, we use it as the sentiment similarity measure in the algorithm explained in Figure 4. The results are presented in Table 4. The first and second rows are baselines. The first row is the result obtained by Marneffe et al. (2010) approach. This approach is based on the similarity between the SO of the adjectives in question and answer. The second row of Table 4 show the results of using a popular semantic similarity measure, PMI, as the sentiment similarity (SS) measure in Figure 4.

Method	Prec.	Rec.	F1
Marneffe et al. (2010)	60.00	60.00	60.00
PMI	60.61	58.70	59.64
PSSS-SHEM	62.55	61.75	61.71
PSSS-BHEM (w/o WSD)	65.90	66.11	63.74
SS-BHEM (with WSD)	66.95	67.15	65.66

Table 4. Performance on IQAP inference task

The result shows that PMI is less effective to capture the sentiment similarity.

Our PSSS approach directly infers *yes* or *no* responses using SS between the adjectives and does not require computing SO of the adjectives. In Table 4, *PSSS-SHEM* and *PSSS-BHEM* indicate the results when we use our PSSS with SHEM and BHEM respectively. Table 4 shows the effectiveness of our sentiment similarity measure. Both models improve the performance over the baselines, while the bridged model leads to higher performance than the series model.

Furthermore, we employ Word Sense Disambiguation (WSD) to disambiguate the adjectives in the question and its corresponding answer. For example, **Q:** ... *Is that true?* **A:** *This is extraordinary and preposterous.* In the answer, the correct sense of the *extraordinary* is *unusual* and as such answer *no* can be correctly inferred. In the table, (*w/o WSD*) is based on the first sense (most common sense) of the words, whereas (*with WSD*) utilizes the real sense of the words. As Table 4 shows, WSD increases the performance. WSD could have higher effect, if more IQAPs contain adjectives with senses different from the first sense.

7 Analysis and Discussions

7.1 Number and Types of Emotions

In our PSSS approach, there is no limitation on the number and types of emotions as we assumed emotions are hidden. In this Section, we perform experiments to predict the number and type of hidden emotions.

Figure 6 and 7 show the results of the hidden models (SHEM and BHEM) on SO prediction and IQAPs inference tasks respectively with different number of emotions. As the Figures show, in both tasks, SHEM achieved high performances with 11 emotions. However, BHEM achieved high performances with six emotions. Now, the question is which emotion number should be considered? To answer this question, we further study the results as follows.

First, for SHEM, there is no significant difference between the performances with six and 11 emotions in the SO prediction task. This is the

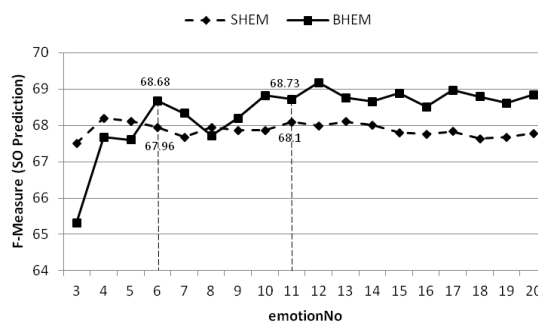


Figure 6. Performance of BHEM and SHEM on SO prediction through different #of emotions

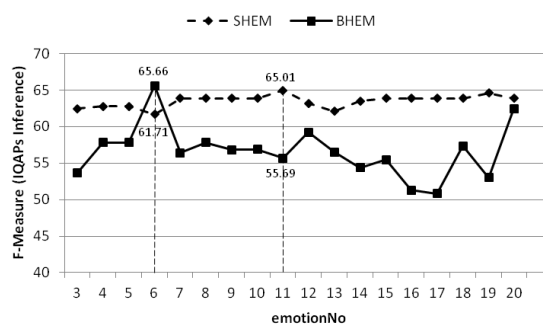


Figure 7. Performance of BHEM and SHEM on IQAPs inference through different #of emotions

same for BHEM. Also, the performances of SHEM on the IQAP inference task with six and 11 emotions are comparable. However, there is a significant difference between the performances of BHEM in six and 11 emotions. So, we consider the dimension in which both hidden emotional models present a reasonable performance over both tasks. This dimension is six here.

Second, as shown in the Figures 6 and 7, in contrast to BHEM, the performance of SHEM does not considerably change with different number of emotions over both tasks. This is because, in SHEM, the emotional vectors of the words are derived from the emotional vectors of the documents after the EM steps, see Equation (14). However, in BHEM, the emotional vectors are directly obtained from the EM steps. Thus, the bridged model is more sensitive than series model to the number of emotions. This could indicate that the bridged model is more accurate than the series model to estimate the number of emotions.

Therefore, based on the above discussion, the estimated number of emotions is six in our development dataset. This number may vary using different development datasets.

In addition to the number of emotions, their types can also be interpreted using our approach. To achieve this aim, we sort the words based on their probability values, $P(w|e)$, with respect to

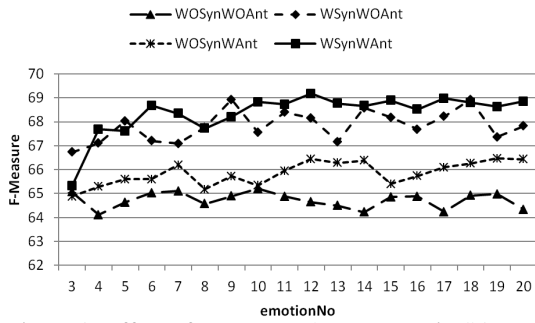


Figure 8. Effect of synonyms & antonyms in SO prediction task with different emotion numbers in BHEM

Emotion#1	Emotion#2	Emotion#3
excellent (1)	unimpressive (1)	disreputable (1)
magnificently(1)	humorlessly (1)	villian (1)
blessed (1)	paltry (1)	onslaught (1)
sublime (1)	humiliating (1)	ugly (1)
affirmation (1)	uncreative (1)	old (1)
tremendous (2)	lackluster (1)	disrupt (1)

Table 5. Sample words in three emotions

each emotion. Then, the type of the emotions can be interpreted by observing the top k words in each emotion. For example, Table 5 shows the top 6 words for three out of six emotions obtained for BHEM. The numbers in parentheses show the sense of the words. The corresponding emotions for these categories can be interpreted as "wonderful", "boring" and "disreputable", respectively.

We also observed that, in SHEM with eleven emotion numbers, some of the emotion categories have similar top k words such that they can be merged to represent the same emotion. Thus, it indicates that the BHEM is better than SHEM to estimates the number of emotions than SHEM.

7.2 Effect of Synsets and Antonyms

We show the important effect of synsets and antonyms in computing the sentiment similarity of words. For this purpose, we repeat the experiment for SO prediction by computing sentiment similarity of word pairs with and without using synonyms and antonyms. Figure 8 shows the results of obtained from BHEM. As the Figure shown, the highest performance can be achieved when synonyms and antonyms are used, while the lowest performance is obtained without using them. Note that, when the synonyms are not used, the entries of enriched matrix are computed using $P(w_i|w_j)$ instead of $P(\text{syn}(w_i)|\text{syn}(w_j))$ in the Equation (15). Also, when the antonyms are not used, the $\text{Max}(\cdot)$ in Equation (18) is 0 and SS is computed using only correlation between words.

The results show that synonyms can improve the performance. As Figure 8 shows, the two

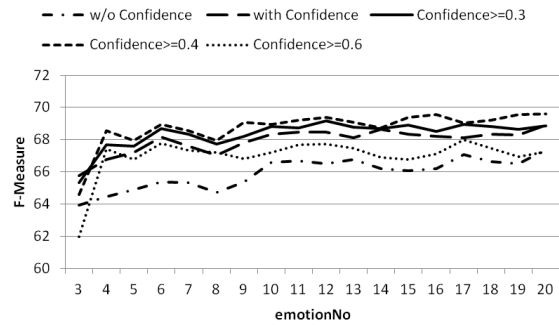


Figure 9. Effect of confidence values in SO prediction with different emotion numbers in BHEM

highest performances are obtained when we use synonyms and the two lowest performances are achieved when we don't use synonyms. This is indicates that the synsets of the words can improve the quality of the enriched matrix. The results also show that the antonyms can improve the result (compare WOSynWAnt with WOSynWOAnt). However, synonyms lead to greater improvement than antonyms (compare WSynWOAnt with WOSynWAnt).

7.3 Effect of Confidence Value

In Section 3.1, we defined a confidence value for each word to improve the quality of the enriched matrix. To illustrate the utility of the confidence value, we repeat the experiment for SO prediction by BHEM using all the words appears in enriched matrix with different confidence thresholds. The results are shown in Figure 9, "w/o confidence" shows the results when we don't use the confidence values, while "with confidence" shows the results when use the confidence values. Also, "confidence>x" indicates the results when we set all the confidence value smaller than x to 0. The thresholding helps to eliminate the effect of low confident words.

As Figure 9 shows, "w/o confidence" leads to the lowest performance, while "with confidence" improves the performance with different number of emotions. The thresholding is also effective. For example, a threshold like 0.3 or 0.4 improves the performance. However, if a large value (e.g., 0.6) is selected as threshold, the performance decreases. This is because a large threshold filters a large number of words from enriched model that decreases the effect of the enriched matrix.

7.4 Convergence Analysis

The PSSS approach is based on the EM algorithm for the BHEM (or SHEM) presented in Table 2. This algorithm performs a predefined

number of iterations or until convergence. To study the convergence of the algorithm, we repeat our experiments for SO prediction and IQAPs inference tasks using BHEM with different number of iterations. Figure 10 shows that after the first 15 iterations the performance does not change dramatically and is nearly constant when more than 30 iterations are performed. This shows that our algorithm will converge in less than 30 iterations for BHEM. We observed the same pattern in SHEM.

7.5 Bridged Vs. Series Model

The bridged and series models are both based on the hidden emotions that were developed to predict the sense sentiment similarity. Although their best results on the SO prediction and IQAPs inference tasks are comparable, they have some significant differences as follows:

- BHEM is considerably faster than SHEM. The reason is that, the input matrix of BHEM (i.e., $W \times R$) is significantly smaller than the input matrix of SHEM (i.e., $W \times D$).
- In BHEM, the emotional vectors are directly computed from the EM steps. However, the emotional vector of a word in SHEM is computed using the emotional vectors of the documents containing the word. This adds noises to the emotional vectors of the words.
- BHEM gives more accurate estimation over type and number of emotions versus SHEM. The reason is explained in Section 7.1.

8 Related Works

Sentiment similarity has not received enough attention to date. Most previous works employed semantic similarity of word pairs to address SO prediction and IQAP inference tasks. Turney and Littman (2003) proposed to compute pair-wised mutual information (PMI) between a target word and a set of seed positive and negative words to infer the SO of the target word. They also utilized Latent Semantic Analysis (LSA) (Landauer et al., 1998) as another semantic similarity measure. However, both PMI and LSA are semantic similarity measure. Similarly, Hassan and Radev (2010) presented a graph-based method for predicting SO of words. They constructed a lexical graph where nodes are words and edges connect two words with semantic similarity obtained from Wordnet (Fellbaum 1998). They propagated the SO of a set of seeds through this graph. However, such approaches did not take into account the sentiment similarity between words.

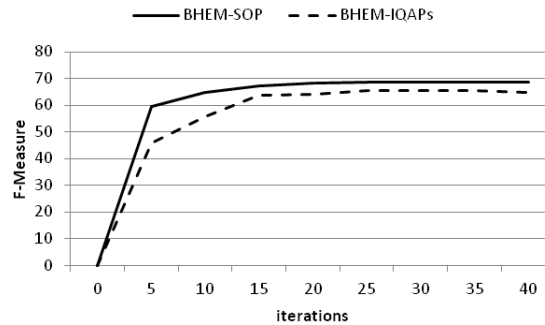


Figure 10. Convergence of BHEM

In IQAPs, Marneffe et al. (2010) inferred the *yes/no* answers using SO of the adjectives. If SO of the adjectives have different signs, then the answer conveys *no*, and Otherwise, if the absolute value of SO for the adjective in question is smaller than the absolute value of the adjective in answer, then the answer conveys *yes*, and otherwise *no*. In Mohtarami et al. (2012), we used two semantic similarity measures (PMI and LSA) for the IQAP inference task. We showed that measuring the sentiment similarities between the adjectives in question and answer leads to higher performance as compared to semantic similarity measures.

In Mohtarami et al. (2012), we proposed an approach to predict the sentiment similarity of words using their emotional vectors. We assumed that the type and number of emotions are pre-defined and our approach was based on this assumption. However, in previous research, there is little agreement about the number and types of basic emotions. Furthermore, the emotions in different dataset can be varied. We relaxed this assumption in Mohtarami et al., (2013) by considering the emotions as hidden and presented a hidden emotional model called SHEM. This paper also consider the emotions as hidden and presents another hidden emotional model called BHEM that gives more accurate estimation of the numbers and types of the hidden emotions.

9 Conclusion

We propose a probabilistic approach to infer the sentiment similarity between word senses with respect to automatically learned hidden emotions. We propose to utilize the correlations between reviews, ratings, and words to learn the hidden emotions. We show the effectiveness of our method in two NLP tasks. Experiments show that our sentiment similarity models lead to effective emotional vector construction and significantly outperform semantic similarity measures for the two NLP task.

References

- Hadi Amiri and Tat S. Chua. 2012. *Mining Slang and Urban Opinion Words and Phrases from cQA Services: An Optimization Approach*. Proceedings of the fifth ACM international conference on Web search and data mining (WSDM). Pp. 193-202.
- Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. Cambridge, MA: MIT Press.
- Ahmed Hassan and Dragomir Radev. 2010. *Identifying Text Polarity Using Random Walks*. Proceeding in the Association for Computational Linguistics (ACL). Pp: 395-403.
- Aminul Islam and Diana Inkpen. 2008. *Semantic text similarity using corpus-based word similarity and string similarity*. ACM Transactions on Knowledge Discovery from Data (TKDD).
- Carroll E. Izard. 1971. *The face of emotion*. New York: Appleton-Century-Crofts.
- Soo M. Kim and Eduard Hovy. 2004. *Determining the sentiment of opinions*. Proceeding of the Conference on Computational Linguistics (COLING). Pp: 1367-1373.
- Thomas K. Landauer, Peter W. Foltz, and Darrell Laham. 1998. *Introduction to Latent Semantic Analysis*. Discourse Processes. Pp: 259-284.
- Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. *Learning Word Vectors for Sentiment Analysis*. Proceeding in the Association for Computational Linguistics (ACL). Pp:142-150.
- Marie-Catherine D. Marneffe, Christopher D. Manning, and Christopher Potts. 2010. *"Was it good? It was provocative." Learning the meaning of scalar adjectives*. Proceeding in the Association for Computational Linguistics (ACL). Pp: 167-176.
- Mitra Mohtarami, Hadi Amiri, Man Lan, Thanh P. Tran, and Chew L. Tan. 2012. *Sense Sentiment Similarity: An Analysis*. Proceeding of the Conference on Artificial Intelligence (AAAI).
- Mitra Mohtarami, Man Lan, and Chew L. Tan. 2013. *From Semantic to Emotional Space in Probabilistic Sense Sentiment Analysis*. Proceeding of the Conference on Artificial Intelligence (AAAI).
- Alena Neviarouskaya, Helmut Prendinger, and Mitsuru Ishizuka. 2007. *Textual Affect Sensing for Sociable and Expressive Online Communication*. Proceedings of the conference on Affective Computing and Intelligent Interaction (ACII). Pp: 218-229.
- Alena Neviarouskaya, Helmut Prendinger, and Mitsuru Ishizuka. 2009. *SentiFul: Generating a Reliable Lexicon for Sentiment Analysis*. Proceeding of the conference on Affective Computing and Intelligent Interaction (ACII). Pp: 363-368.
- Andrew Ortony and Terence J. Turner. 1990. *What's Basic About Basic Emotions*. American Psychological Association. 97(3), 315-331.
- Christopher Potts, C. 2011. *On the negativity of negation*. In Nan Li and David Lutz, eds., Proceedings of Semantics and Linguistic Theory 20, 636-659.
- Peter D. Turney and Michael L. Littman. 2003. *Measuring Praise and Criticism: Inference of Semantic Orientation from Association*. ACM Transactions on Information Systems, 21(4), 315-346.
- Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. *Recognizing contextual polarity in phrase-level sentiment analysis*. Proceeding in HLT-EMNLP. Pp: 347-354.