Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence

# From Semantic to Emotional Space in Probabilistic Sense Sentiment Analysis

**Mitra Mohtarami[1], Man Lan[2], and Chew Lim Tan[1]**
[1]Department of Computer Science, National University of Singapore;
[2]Institute for Infocomm Research
mitra@comp.nus.edu.sg; mlan@i2r.a-star.edu.sg; tancl@comp.nus.edu.sg

## Abstract

This paper proposes an effective approach to model the emotional space of words to infer their *Sense Sentiment Similarity* (SSS). SSS reflects the distance between the words regarding their senses and underlying sentiments. We propose a probabilistic approach that is built on a hidden emotional model in which the basic human emotions are considered as hidden. This leads to predict a vector of emotions for each sense of the words, and then to infer the sense sentiment similarity. The effectiveness of the proposed approach is investigated in two Natural Language Processing tasks: *Indirect yes/no Question Answer Pairs Inference* and *Sentiment Orientation Prediction*.

## Introduction

Sentiment analysis or opinion mining aims to enable computers to derive sentiment from human language. In this paper, we aim to address sense sentiment similarity that aims to infer the similarity between word pairs with respect to their senses and underlying sentiments.

Previous works employed semantic similarity measures to estimate sentiment similarity of word pairs (Kim and Hovy 2004; Turney and Littman 2003). However, it has been shown that although the semantic similarity measures are good for relating semantically related words like "*car*" and "*automobile*" (Islam et al., 2008), but are less effective to capture sentiment similarity (Mohtarami et al., 2012). For example, using Latent Semantic Analysis (Landauer et al., 1998), the semantic similarity of "*excellent*" and "*superior*" is greater than the similarity between "*excellent*" and "*good*". However, the intensity of sentiment in "*excellent*" is more similar to "*superior*" than "*good*". That is, sentiment similarity of "*excellent*" and "*superior*" should be greater than "*excellent*" and "*good*".



This paper shows that not only semantic similarity measures are less effective, considering just the total sentiment of words (as positive or negative) is also not sufficient to accurately infer sentiment similarity between words senses. The reason is that, although the opinion words can be categorized into *positive* and *negative* sentiments with different sentiment intensity values, they carry different human emotions. For instance, consider a fixed set of emotions *e* = [*anger, disgust, sadness, fear, guilt, interest, joy, shame, surprise*] where each dimension ranges from 0 to 1. Given the above emotions, the emotion vectors and the sentiment orientation (SO) of the words "*doleful*", "*rude*" and "*smashed*" will be as follows (Neviarouskaya et al., 2007; Neviarouskaya et al., 2009):

e(*rude*) =[0.2,0.4,0,0,0,0,0,0,0], SO(*rude*)=-0.2-0.4=-0.6
e(*doleful*)=[0,0,0.4,0,0,0,0,0,0], SO(*doleful*)=-0.4
e(*smashed*)=[0,0,0.8,0.6,0,0,0,0,0], SO(*smashed*)=-1.4

All the three words have negative sentiment and SO of "*doleful*" is closer to "*rude*" than "*smashed*". However, the emotional vectors indicate that "*rude*" only carries the emotions "*anger*" and "*disgust*", while "*doleful*" and "*smashed*" carry the same emotion "*sadness*". As such, considering the emotional space of words, the word "*doleful*" should be closer to "*smashed*" than "*rude*".

This paper shows that using emotional vectors of the words is more effective than using semantic similarity measures or considering sentiment of the words (as positive or negative) to infer sense sentiment similarity. To achieve this aim, we propose a probabilistic approach by combining semantic and emotional spaces. Furthermore, we show the utility of sentiment similarity in *Indirect yes/no Question Answer Pairs* (IQAPs) *Inference* and *Sentiment Orientation* (SO) *prediction* tasks explained as follows:

In IQAPs, the answer of the question does not explicitly contain a clear *yes* or *no*, but rather gives information to infer such an answer (e.g. **Q**: *Was she the* ***best*** *one on that old show*? **A**: *She was simply* ***funny***). Clearly, the sentiment words in an IQAP are the pivots to determine the final answer as *yes* or *no*. We show that sentiment similarity between the adjectives in IQAPs can be used to effectively infer the *yes* or *no* answers.

SO prediction aims to determine the sentiment orientation of words. Previous research utilized (a) word relations obtained from WordNet (Hassan and Radev 2010), (b) external resources like review rating (Marneffe et al., 2010), and (c) semantic similarity measures for this purpose (Turney and Littman 2003). We show that sentiment similarity is a more appropriate measure to achieve accurate SO of words.

In summary, the contributions of this paper are follows:

- We propose a series hidden emotional model to predict emotional vectors of senses of words and infer SSS employing the emotional vectors, and
- We show that the sentiment similarity computed using emotional vectors is more accurate than using the SO of the words, and
- We show that such sentiment similarity can be utilized to get accurate SO for each sense of the words.

## Method: Probabilistic Sense Sentiment Similarity

Previous researches showed that there exists a small set of basic (or fundamental) emotions which are central to other emotions (Ortony and Turner 1990; Izard 1971). Regarding the previous researches, Mohtarami et al. (2012) introduced twelve basic emotions that are central and generally accepted: *anger*, *disgust*, *fear*, *guilt*, *sadness*, *shame*, *interest*, *joy*, *surprise*, *desire*, *love*, *courage*. However, in previous research, there is little agreement about the number and types of basic emotions. In this research, we suppose that the number and types of basic emotions are not pre-defined. Then, we propose an effective probabilistic approach in which the emotions are considered as hidden, and the combination of sentiment and semantic spaces are employed to achieve sentiment similarity. We follow two general steps to achieve this aim: (a): *Constructing Emotional Vectors*, and (b): *Predicting Word Pair Sentiment Similarity*.

### Constructing Emotional Vectors

Online review systems provide means for users to write their comments and experiences about different products and services. The review systems usually employ a rating mechanism (mostly in terms of stars) to allow users attach a rating to their reviews. A rating indicates the summarized opinion of a user who ranks a product or service regarding the reasons and feelings written in his review. The range of rating in star-based review systems is often five-star (e.g., Amazon review system) or ten-star (e.g., IMDB review system) in which the maximum (minimum) number of stars presents the user's strong positive (negative) sentiment.

Though positive or negative ratings are assigned to the reviews, there are various feelings/emotions behind the ratings with respect to the reviews. Figure 1 shows rating $r$ from a set of ratings $R=\{r_1,\ldots,r_p\}$ is assigned to a hidden emotion set $E=\{e_1,\ldots,e_k\}$. A document $d$ from a set of documents $D=\{d_1,\ldots,d_N\}$ that contain the word set $W=\{w_1,\ldots,w_M\}$ is associated with the hidden emotion set.

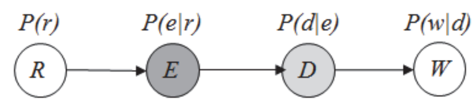


Figure 1: Series hidden emotional model

We aim to employ the ratings and the relation between ratings, documents and words with respect to the hidden emotional model shown in Figure 1 to extract the hidden emotions. Note that the lines indicate the dependency between the elements $r$, $e$, $d$, and $w$. The proposed emotional model is called *Series Hidden Emotional Model* (SHEM).

The entire text collection can be represented 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 base on the joint probability of the observation tuples $(w,d,r)$ as the follows:

$$D = \prod_{r}\prod_{d}\prod_{w} P(w,d,r)^{n(w,d,r)} = \prod_{r}\prod_{d}\prod_{w} P(w,d,r)^{n(w,d)n(d,r)} \tag{1}$$

where, $P(w,d,r)$ is the joint probability of the tuple $(w,d,r)$, and $n(w,d,r)$ is the frequency of the word $w$ that occurs in both document $d$ and rating $r$. We represent $n(w,d,r)$ as the multiplication of $n(w,d)$ and $n(d,r)$ where $n(w,d)$ is the frequency of $w$ in the document $d$, and $n(d,r)$ is one if $r$ is assigned to $d$, and otherwise zero.

To infer the joint probability $P(w,d,r)$ with respect to the Figure 1, we assumes the following assumption ($A1$):

$$A1: \begin{cases} \text{1. implicit conditional independence assumption: } r \text{ and } d \text{ are independent conditioned on the state of the associated hidden variable } e, \text{ and} \\ \text{2. The word } w \text{ is dependent to } d \text{ and independent to } e. \end{cases}$$

The joint probability for this model can be defined as follows based on the hidden emotion $e$, A1 and Bayes' rule:

$$P(w,d,r) = P(w|d)\sum_{e} P(d|e)P(e)P(r|e) \qquad (2)$$

We employ Maximum Likelihood to learn the unobservable probabilities (e.g., $P(d|e)$, $P(r|e)$, $P(e)$) to infer the possible hidden emotions. The log-likelihood of the whole data set $D$ of Equation (1) can be defined as follows:

$$L = \sum_{r}\sum_{d}\sum_{w} n(w,d)n(d,r)\log P(w,d,r) \qquad (3)$$

Substituting Equation (2) in Equation (3) leads to:

$$L = \sum_{r}\sum_{d}\sum_{w} n(w,d)n(d,r)\log[P(w|d)\sum_{e} P(d|e)P(e)P(r|e)] \qquad (4)$$

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

1. E-step: Calculate expectation (posterior probabilities) for hidden variables given the observations by using the current estimates of the parameters, and
2. M-step: Update parameters such that the data log-likelihood (log L) increases using the posterior probabilities in the E-step.

The steps of EM can be computed with respect to the series hidden emotional model. We derive the EM Equations for the series hidden emotional model by employing the assumptions *A1* and Bayes Rule as follows:

**E-step**:

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

**M-step**:

$$P(r|e) = \frac{\sum_d \sum_w n(w,d)n(d,r)\, P(\mathrm{e}|w,d,r)}{\sum_r \sum_d \sum_w n(w,d)n(d,r)\, P(\mathrm{e}|w,d,r)} \qquad (6)$$

$$P(d|e) = \frac{\sum_r \sum_w n(w,d)n(d,r)\, P(\mathrm{e}|w,d,r)}{\sum_d \sum_r \sum_w n(w,d)n(d,r)\, P(\mathrm{e}|w,d,r)} \qquad (7)$$

$$P(e) = \frac{\sum_r \sum_d \sum_w n(w,d)n(d,r)\, P(\mathrm{e}|w,d,r)}{\sum_{\mathrm{e}} \sum_d \sum_r \sum_w n(w,d)n(d,r)\, P(\mathrm{e}|w,d,r)} \qquad (8)$$

Finally, the following algorithms employ the EM steps to construct emotional vectors. The algorithm can be used for series hidden model. The goal is to infer the emotional vector for each word *w*. We achieve this aim by computing the $P(w|e)$. The series model does not contain the probability $P(w|e)$, however, the model can compute this probability using $P(d|e)$ as follows:

$$P(w|e) = \sum_{d} P(w|d)P(d|e) \qquad (9)$$

| |
|---|
| **Input**: Document-Rating D×R, Term-Document W×D<br>**Output**: Emotional vectors $\{e_1,\ldots,e_k\}$ for each *d* and *w*<br><br>**Algorithm:**<br>1. Enriching hidden emotional model: update W×D<br>2. Initialize $P(d|e)$, $P(r|e)$, and $P(e)$ randomly,<br>3. **while** *L* has not converged to a pre-specified value **do**<br>4. E-step; estimate the value of $P(e|w,d,r)$ in Equation 5,<br>5. M-step; estimate the values of $P(r|e)$, $P(d|e)$, and $P(e)$ in Equations 6-8, respectively<br>6. **end while**<br>7. Infer word emotional vector: estimate $P(w|e)$ in Equation 9. |

Table 1: Algorithm of series hidden emotional model

The algorithm for our hidden models is shown in the Table 1. In the algorithm, we employ the document-rate and term-document matrices to infer the unknown probabilities (e.g., $P(w|e)$). Line 1 in the algorithm will be explained in the next subsection.

**Enriched Hidden Emotional Model**

In our hidden model, the term-document and document-rating matrices are employed as inputs to infer the emotional vectors. The matrices just present the knowledge about the frequency of a word in documents or documents in ratings.

Suppose we have prior information about the semantic similarity between some words before using the hidden model. For example, there are two words $w_1$ and $w_2$ in the matrices that are synonyms (thus their emotional vectors should be similar). The question is how this knowledge can be transferred to our model. One simple way is using some post-processing after getting the emotional vectors of $w_1$ and $w_2$, e.g. by averaging their emotional vectors. However, this approach is less effective, since the knowledge about the synonyms $w_1$ and $w_2$ has not yet transferred to the hidden model and this knowledge has not employed in learning step of the model. To utilize the word similarity knowledge, we use the following enriched matrix in which each cell shows the semantic relation between the two words in the corresponding row and column. If we do not have any knowledge about two words or they are not sentimentally co-related, their corresponding cell will be zero. To compute the semantic similarity between each two words, we utilize the synset of the words as follows:

$$w_i w_j = P\big(syn(w_i)|syn(w_j)\big)$$

$$= \frac{1}{|syn(w_i)|}\sum_{\mathrm{i}}^{|syn(w_i)|} \frac{1}{|syn(w_j)|}\sum_{\mathrm{j}}^{|syn(w_j)|} P(w_i|w_j) \qquad (10)$$

where, *syn(w)* is the synset of word *w*. Let *count*($w_i$, $w_j$) be the co-occurrence of the words $w_i$ and $w_j$, and let *count*($w_j$) be the total word count. The probability of $w_i$ given $w_j$ will then be $P(w_i|w_j) = count(w_i, w_j)/ count(w_j)$.

The reason we employ the co-occurrence of the words is as follows. First, we employ the hypothesis that a word can be characterized by its neighbors (Turney and Littman 2003). That is, the emotional vector of a word tends to correspond to the emotional vectors of its neighbors. Second, each entry of the input matrices of our hidden model is based on frequency of a word in whole length of a document or rating. However, this scale is large and may add some noise to our hidden model. The co-occurrence of words in a small window can make our model more accurate.

In addition, the reason we employ the synset of the words is as follows. First, as the synset of a word has the same or nearly the same meaning as the original word, the word can be replaced by any of its synset with no major changes in its emotion. Second, the major advantage of using synset is that we can obtain different emotional vectors for each sense of a word and predict the sentiment similarity at the sense level. Note that, various senses of a word can have diverse meanings and emotions, and consequently different emotional vectors. If two words $w_i$ and $w_j$ are synonyms, their corresponding entry in the enriched matrix will be one.

To improve our hidden model, the enriched matrix W×W is multiplied to the inputs of the model W×D such that the sense of words can be added to the matrices. 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 be indirectly associated to the document through the word $w_j$.

## Predicting Sense Sentiment Similarity

So far, we computed the emotional vectors of the words with respect to their senses using the proposed series hidden emotional model. To infer the sentiment similarity of words, we compare each emotion of a word with corresponding emotion of another. To achieve this aim, we use the correlation coefficient between the emotional vectors of two words to compute the sentiment similarity between them regarding their senses. Let X and Y be the emotional vectors of two words. Equation 11 computes their correlation:

$$corr(X,Y) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{(n-1)S_X S_Y} \qquad (11)$$

where, $n$ is number of emotional categories, $\bar{X}, \bar{Y}$ and $S_X, S_Y$ are the mean and standard deviation values of $\boldsymbol{X}$ and $\boldsymbol{Y}$ respectively.

The problem is that how large the correlation value should be to consider two words as similar in sentiment. We address this issue by utilizing the antonyms of the words (Mohtarami et al., 2012). Since the word and its antonyms have opposite sentiment orientation, we consider two words, $w_i$ 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, ${\sim}w_i$ (${\sim}w_j$) are antonyms of $w_i$ ($w_j$) respectively, and $corr(w_i, w_j)$ is the correlation between the emotional vectors obtained from Equation 11. Finally, we compute the sentiment similarity (*SS*) between two words as follows:

$$SS(w_i, w_j) = corr(w_i, w_j) - Max\{corr(w_i, {\sim}w_j), corr({\sim}w_i, w_j)\} \qquad (12)$$

A positive value of $SS(.,.)$ indicates that the words are sentimentally similar and negative value shows the amount of dissimilarity between the words.

## Applications

This section explains how sentiment similarity can be employed to address the IQAP inference and SO prediction tasks, respectively.

Since the adjectives in the questions and its corresponding answers are the main factors to infer *yes* or *no* answers in IQAPs (Marneffe et al., 2010; Mohtarami et al., 2011; Mohtarami et al., 2012), the following algorithm is employed for interpreting the indirect answer in a given IQAP. First, the sentiment similarity (SS) between the adjective in question and the adjective in answer is computed regarding their correct senses using Equation (12). Then, if the value of *SS* is positive, it means the question and its answer are sentimentally similar and the response is *yes*; and a negative *SS* leads to *no* response.

In second application, we attempt to show that sentiment similarity along with a simple algorithm is able to accurately predict sentiment orientation (SO). To achieve this aim, *SS* is computed from Equation (12) and the algorithm presented by Turney and Littman (2003) is used as follows:

$$SO\_A(w) = \sum_{pword \in Pwords} A(w, pword) - \sum_{nword \in Nwords} A(w, nword) \qquad (13)$$

where, *A(.,.)* is a similarity measure. In equation (13), the SO of a given word is calculated from its similarity with seven positive words like "*excellent*", minus its similarity with seven negative words like "*poor*". As a similarity measure, we utilize our sentiment similarity measure.

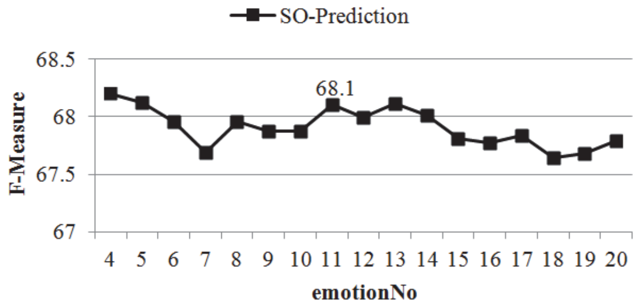


Figure 2. F1 performance through different number of emotions in SO prediction

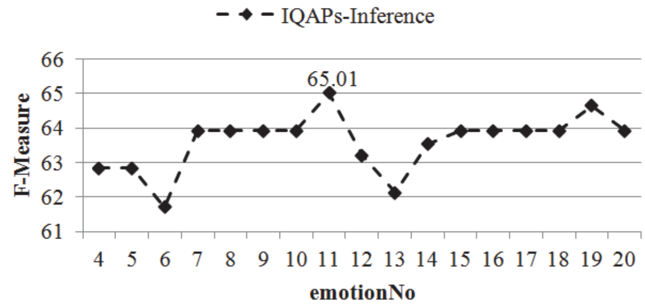


Figure 3. F1 performance through different number of emotions in IQAPs inference

## Results and Discussions

In this section we explain the datasets used, and report the experiments conducted to evaluate our approach.

### Data and Settings

The movie review dataset developed by Maas et al. (2011) is used as the development dataset to compute the required statistics. This dataset contains 50k movie reviews and 90k vocabulary. In the dataset, each review has an associated star rating: one star (most negative) to ten stars (most positive) excluding the ratings 5 and 6 that are more neutral. For the evaluation purpose, we used two datasets: 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. For MPQA dataset, we ignore the neutral words and use the remaining 4k opinion words with their SOs. The IQAPs dataset contains a 125 IQAPs and their corresponding *yes* or *no* labels as the ground truth as described in (Marneffe et al., 2010).

### Evaluation of Sentiment Orientation Prediction

In our probabilistic approach, there is not any assumption on the number and types of the emotions. That is, we assumed that the emotions are hidden. Now, to predict the number of hidden emotions, we repeat our experiment on SO prediction and IQAPs inference tasks using probabilistic SSS (PSSS) with different number of emotions.

Figure 2 and 3 show the results with emotion numbers ranging from 2 to 20. The best result is achieved with eleven emotions in IQAPs inference as shown in Figure 3. In addition, SO prediction obtains a good accuracy of 68.10% with eleven emotions, such that it is not significantly different from the highest accuracy (68.20%) obtained with four emotions. Thus, the estimated number of emotions is eleven in our development dataset. We note that the number of emotions may vary using different datasets. The result of SO prediction is reported in the third row of Table 2 and compared with the baselines.

| Method | Precision | Recall | F1 |
|---|---|---|---|
| SO-PMI | 56.20 | 56.36 | 55.01 |
| ER | 65.68 | 65.68 | 63.27 |
| PSSS-SHEM | **68.69** | **69.37** | **68.10** |

Table 2: F1 performance on SO prediction

In Table 2, the first and second rows present the results obtained by SO-PMI and Expected Rating (ER) of a word respectively. SO-PMI is computed by Equation (13) using PMI as similarity measure. PMI extracts the semantic similarity between words using their co-occurrences. As Table 2 shows PMI produces poor results. This is because our development dataset is relatively small which leads to poor co-occurrence information. ER (Potts, 2011) 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. Our proposed PSSS approach employs series hidden emotional model, and its result is reported in the third row. PSSS outperforms both baselines. The reason is that PSSS presents an approach based on the combination between sentiment space (through the input $D \times R$) and semantic space (through the input $W \times D$). 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 types of emotions can be interpreted using our approach. To achieve this aim, considering the probability $P(w|e)$, we can sort the words based on their probabilities in each emotion. Then, the type of the emotions can be interpreted by seeing the top K of the words in each emotion. To make the interpretation of the emotion types straightforward, the words can be assigned to the emotion with the highest probability. For example, top 5 words for three emotional categories are listed below where the emotions can be interpreted as "contented", "brilliance" and "fear", respectively.

| Emotion#1 | Emotion#2 | Emotion#3 |
|---|---|---|
| fairly | brilliant | aghast |
| reasonably | superb | dismayed |
| universal | unanimous | appalled |
| profit | lucky | questionable |
| decent | zealous | lacking |

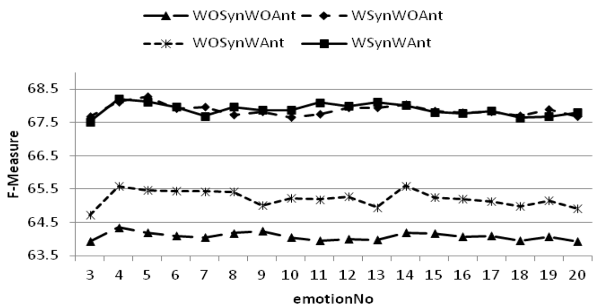

Figure 4. Role of synonyms and antonyms

**Role of synsets and antonyms of words:** We show the important role of using synsets and antonyms of words which we employed in our method. For this purpose, we repeat the experiment for SO prediction by computing sentiment similarity of word pairs without using synonyms and antonyms. Figure 4 shows the results. As it is clear, the highest performance can be achieved when synonyms and antonyms are used, while the lowest performance is obtained without using them. When the synonyms are not used, the entries of enriched matrix are computed using $P(w_i,w_j)$ instead of $P(syn(w_i),syn(w_j))$ in the Equation (10). The results show that the synonyms can improve the result. This is because the enriched matrix is based on co-occurrence of words and synsets of the words can make it more accurate. When the antonyms are not used, the *Max(,)* in Equation (12) returns zero and SS is computed using only correlation between the words. The results show that the antonyms cannot significantly improve the results, when the synsets are used. This demonstrates that the synsets make the model more robust.

### Evaluation of IQAPs Inference

As shown in Figure 3, the best result can be achieved with eleven emotions in IQAPs inference. Table 3 shows the results of PSSS using eleven emotions. The first row presents the result obtained by Marneffe et al. (2010). They computed SO of the adjectives based on the expected ratings (ER), and then employed the SO to infer *yes* or *no*. However, our experiments show that the sentiment similarity based on emotional vectors is more accurate than the similarity based on SO.

The second row of Table 3 shows the result of using PMI as the similarity measure in the algorithm explained the section "Applications". The last rows, *PSSS-SHEM (with or w/o WSD)* indicate the results when we use our *PSSS* with and without WSD respectively. *PSSS-SHEM (w/o WSD)* is based on the first sense of the words, whereas *PSSS-SHEM (with WSD)* utilizes the real sense of the words. As it is clear in Table 3, using WSD and correct sense of the adjectives increase the performance from 64.62% to 65.01%. The efficiency of the WSD could have been more highlighted, if more IQAPs contain adjectives with senses different from their first senses.

| Method | Precision | Recall | F1 |
|---|---|---|---|
| Marneffe et al. (2010) | 60.00 | 60.00 | 60.00 |
| PMI | 60.61 | 58.70 | 59.64 |
| PSSS-SHEM (w/o WSD) | 65.83 | **65.84** | 64.62 |
| PSSS-SHEM (with WSD) | **65.92** | 65.70 | **65.01** |

Table 3. F1 performance on IQAPs

## Related Works

Sentiment similarity has not received enough attention to date. Most previous works employed semantic similarity measures to address the NLP tasks. Turney and Littman (2003) proposed a method for automatically inferring the SO of a word from its semantic similarity with a set of positive and negative seed words. To calculate the semantic similarity, they used PMI. 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 based on Wordnet (Fellbaum 1998). Then, they propagated the SO of a set of seed words through graph.

In IQAPs, Marneffe et al. (2010) attempted to infer the *yes/no* answers using SO of the adjectives. They calculated the probability of rating given adjectives over dataset to compute SO. If the adjectives in an IQAP have different SO (i.e. one positive and negative), the answer conveys *no*. 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*. Mohtarami et al. (2011) used semantic similarity measures (PMI and LSA) to infer *yes/no* from indirect answers in a given IQAP. They showed that measuring the association between the adjectives in question and answer can be a main factor to infer a clear response from an IQAP. It is notable that our experiments showed that we can achieve an accurate SO, and better infer *yes/no* answer from a given IQAP by using sentiment similarity measure.

Mohtarami et al. (2012) proposed an approach to predict the sentiment similarity of words using their emotional vectors. However, they assumed that the type and number of emotions are pre-defined. However, as we discussed before, previous research has little agreement about the number and types of basic emotions. Furthermore, the emotions in different dataset can be varied. In this paper, we treated emotions as hidden and developed algorithms to identify them.

## Conclusion

This paper presents a probabilistic approach to infer sense sentiment similarity without knowledge about type and number of basic emotions. We show that sentiment similarity constructed from emotional vectors is more accurate than the similarity based on overall sentiment of the words. Our sentiment similarity approach can predict the sentiment orientation of words that is comparable with state-of-the-art approaches.

## References

Fellbaum., C. 1998. *WordNet: An Electronic Lexical Database*. Cambridge, MA: MIT Press.

Hassan, A., and Radev, D. 2010. *Identifying Text Polarity Using Random Walks.* Proceeding in the Association for Computational Linguistics (ACL). Pp: 395–403.

Islam, A., and Diana I. 2008. *Semantic text similarity using corpus-based word similarity and string similarity*. ACM Transactions on Knowledge Discovery from Data (TKDD).

Izard, C. E. 1971. *The face of emotion.* New York: Appleton-Century-Crofts.

Kim, S. M., and Hovy, E. 2004. *Determining the sentiment of opinions*. Proceeding of the Conference on Computational Linguistics (COLING). Pp: 1367–1373.

Landauer, T. K., Foltz, P. W., and Laham, D. 1998. *Introduction to Latent Semantic Analysis*. Discourse Processes. 259-284.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. 2011. *Learning Word Vectors for Sentiment Analysis*. Proceeding in the Association for Computational Linguistics (ACL). Pp:142-150.

Marneffe, M., Manning, C., and Potts, C. 2010. *"Was it good? It was provocative." Learning the meaning of scalar adjectives*. Proceeding in the Association for Computational Linguistics (ACL). Pp: 167–176.

Mohtarami, M., Amiri, H., Lan, M., and Tan, C. L. 2011. *Predicting the Uncertainty of Sentiment Adjectives in Indirect Answers*. 20th ACM Conference on Information and Knowledge Management (CIKM). Pp. 2485-2488.

Mohtarami, M., Amiri, H., Lan, M., Tran, T. P., and Tan, C. L. 2012. *Sense Sentiment Similarity: An Analysis*. Proceeding of the Conference on Artificial Intelligence (AAAI).

Neviarouskaya, A., Prendinger, H., and Ishizuka, M. 2009. *SentiFul: Generating a Reliable Lexicon for Sentiment Analysis*. Proceeding of the conference on Affective Computing and Intelligent Interaction (ACII). 363-368.

Neviarouskaya, A., Prendinger, H., and Ishizuka, M. 2007. *Textual Affect Sensing for Sociable and Expressive Online Communication*. Proceedings of the conference on Affective Computing and Intelligent Interaction (ACII). Pp: 218-229.

Ortony, A., and Turner, T. J. 1990. *What's Basic About Basic Emotions*. American Psychological Association. 97(3), 315-331.

Potts, C. 2011. *On the negativity of negation*. In Nan Li and David Lutz, eds., Proceedings of Semantics and Linguistic Theory 20, 636-659.

Turney, P., and Littman, M. 2003. *Measuring Praise and Criticism: Inference of Semantic Orientation from Association*. ACM Transactions on Information Systems, 21(4), 315–346.

Wilson, T., Wiebe, J., and Hoffmann, P. 2005. *Recognizing contextual polarity in phrase-level sentiment analysis*. Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT-EMNLP). Pp: 347–354.