Ranking Reader Emotions Using Pairwise Loss Minimization and Emotional Distribution Regression

Kevin Hsin-Yih Lin and **Hsin-Hsi Chen**
Department of Computer Science and Information Engineering
National Taiwan University
No. 1 Roosevelt Rd. Sec. 4, Taipei, Taiwan
{f93141, hhchen}@csie.ntu.edu.tw

Abstract

This paper presents two approaches to ranking reader emotions of documents. Past studies assign a document to a single emotion category, so their methods cannot be applied directly to the emotion ranking problem. Furthermore, whereas previous research analyzes emotions from the writer's perspective, this work examines readers' emotional states. The first approach proposed in this paper minimizes pairwise ranking errors. In the second approach, regression is used to model emotional distributions. Experiment results show that the regression method is more effective at identifying the most popular emotion, but the pairwise loss minimization method produces ranked lists of emotions that have better correlations with the correct lists.

1 Introduction

Emotion analysis is an increasingly popular research topic due to the emergence of large-scale emotion data on the web. Previous work primarily studies emotional contents of texts from the writer's perspective, where it is typically assumed that a writer expresses only a single emotion in a document. Unfortunately, this premise does not hold when analyzing a document from the reader's perspective, because readers rarely agree unanimously on the emotion that a document instills. Figure 1 illustrates this phenomenon. In the figure,

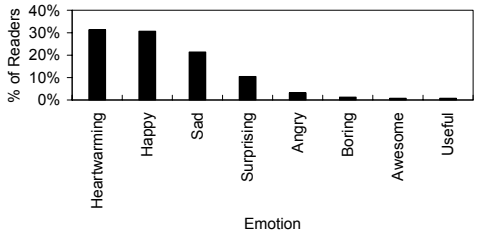

Figure 1. Emotional responses of 626 people after reading a Yahoo! News article about an Iranian refugee mother and her two children who finally reunited with their family in the March of 2007 after been stranded in a Moscow airport for 10 months due to false passports.

readers' responses are distributed among different emotion categories. In fact, none of the emotions in Figure 1 has a majority (i.e., more than 50%) of the votes. Intuitively, it is better to provide a ranking of emotions according to their popularity rather than associating a single reader emotion with a document. As a result, current writer-emotion analysis techniques for classifying a document into a single emotion category are not suitable for analyzing reader emotions. New methods capable of ranking emotions are required.

Reader-emotion analysis has potential applications that differ from those of writer-emotion analysis. For example, by integrating emotion ranking into information retrieval, users will be able to retrieve documents that contain relevant contents and at the same time produce desired feelings. In addition, reader-emotion analysis can assist writers in foreseeing how their work will influence readers emotionally.

Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, pages 136–144,
Honolulu, October 2008.

In this paper, we present two approaches to ranking reader emotions. The first approach is inspired by the success of the pairwise loss minimization framework used in information retrieval to rank documents. Along a similar line, we devise a novel scheme to minimize the number of incorrectly-ordered emotion pairs in a document. In the second approach, regression is used to model reader-emotion distributions directly. Experiment results show that the regression method is more effective at identifying the most popular emotion, but the pairwise loss minimization method produces ordered lists of emotions that have better correlations with the correct lists.

The rest of this paper is organized as follows. Section 2 describes related work. In Section 3, details about the two proposed approaches are provided. Section 4 introduces the corpus and Section 5 presents how features are extracted from the corpus. Section 6 shows the experiment procedures and results. Section 7 concludes the paper.

2 Related Work

Only a few studies in the past deal with the reader aspect of emotion analysis. For example, Lin et al. (2007; 2008) classify documents into reader-emotion categories. Most previous work focuses on the writer's perspective. Pang et al. (2002) design an algorithm to determine whether a document's author expresses a positive or negative sentiment. They discover that using Support Vector Machines (SVM) with word unigram features results in the best performance. Since then, more work has been done to find features better than unigrams. In (Hu et al., 2005), word sentiment information is exploited to achieve better classification accuracy.

Experiments have been done to extract emotional information from texts at granularities finer than documents. Wiebe (2000) investigates the subjectivity of words, whereas Aman and Szpakowicz (2007) manually label phrases with emotional categories. In 2007, the SemEval-2007 workshop organized a task on the unsupervised annotation of news headlines with emotions (Strapparava and Mihalcea, 2007).

As for the task of ranking, many machine-learning algorithms have been proposed in information retrieval. These techniques generate ranking functions which predict the relevance of a document. One class of algorithms minimizes the errors resulting from ordering document pairs incorrectly. Examples include (Joachims, 2002), (Freund et al., 2003) and (Qin et al., 2007). In particular, the training phase of the Joachims' Ranking SVM (Joachims, 2002) is formulated as the following SVM optimization problem:

$$\min_{\boldsymbol{w},\,\xi_{i,j,k}} \ \tfrac{1}{2}\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w} + C\sum \xi_{i,j,k}$$

subject to:

$$\begin{aligned} &\forall (q_k,d_i),(q_k,d_j) \in V \mid s_{k,i} > s_{k,j}: \\ &\quad \boldsymbol{w}^{\mathrm{T}}(\Phi(q_k,d_i) - \Phi(q_k,d_j)) \geq 1 - \xi_{i,j,k} \qquad (1) \\ &\forall i \forall j \forall k : \xi_{i,j,k} \geq 0 \end{aligned}$$

where V is the training corpus, $\Phi(q_k, d_i)$ is the feature vector of document d_i with respect to query q_k, $s_{k,i}$ is the relevance score of d_i with respect to q_k, $\boldsymbol{w}$ is a weight vector, C is the SVM cost parameter, and $\xi_{i,j,k}$ are slack variables. The set of constraints at (1) means that document pairwise orders should be preserved.

Unfortunately, the above scheme for exploiting pairwise order information cannot be applied directly to the emotion ranking task, because the task requires us to rank emotions within a document rather than provide a ranking of documents. In particular, the definitions of $\Phi(q_k,d_i)$, $\Phi(q_k,d_j)$, $s_{k,i}$ and $s_{k,j}$ do not apply to emotion ranking. In the next section, we will show how the pairwise loss minimization concept is adapted for emotion ranking.

3 Ranking Reader Emotions

In this section, we provide the formal description of the reader-emotion ranking problem. Then we describe the pairwise loss minimization (PLM) approach and the emotional distribution regression (EDR) approach to ranking emotions.

3.1 Problem Specification

The reader emotion ranking problem is defined as follows. Let $D = \{d_1, d_2, \ldots, d_N\}$ be the document space, and $E = \{e_1, e_2, \ldots, e_M\}$ be the emotion space. Let $f_i : E \rightarrow \Re$ be the emotional probability function of $d_i \in D$. That is, $f_i(e_j)$ outputs the fraction of readers who experience emotion e_j after reading document d_i. Our goal is to find a function $r : D \rightarrow E^M$ such that $r(d_i) = (e_{\pi(1)}, e_{\pi(2)}, \ldots, e_{\pi(M)})$ where π is

a permutation on {1, 2, …, M}, and $f_i(e_{\pi(1)}) \geq f_i(e_{\pi(2)}) \geq \ldots \geq f_i(e_{\pi(M)})$.

3.2 Pairwise Loss Minimization

As explained in Section 2, the information retrieval framework for exploiting pairwise order information cannot be applied directly to the emotion ranking problem. Hence, we introduce a novel formulation of the emotion ranking problem into an SVM optimization problem with constraints based on pairwise loss minimization.

Whereas Ranking SVM generates only a single ranking function, our method creates a pairwise ranking function $g_{jk} : D \rightarrow \Re$ for each pair of emotions e_j and e_k, aiming at satisfying the maximum number of the inequalities:

$$\forall d_i \in D \mid f_i(e_j) > f_i(e_k) : g_{jk}(d_i) > 0$$
$$\forall d_i \in D \mid f_i(e_j) < f_i(e_k) : g_{jk}(d_i) < 0$$

In other words, we want to minimize the number of incorrectly-ordered emotion pairs. We further require $g_{jk}(d_i)$ to have the linear form $\boldsymbol{w}^{\mathrm{T}}\Omega(d_i) + b$, where $\boldsymbol{w}$ is a weight vector, b is a constant, and $\Omega(d_i)$ is the feature vector of d_i. Details of feature extraction will be presented in Section 5.

As Joachims (2002) points out, the above type of problem is NP-Hard. However, an approximate solution to finding g_{ik} can be obtained by solving the following SVM optimization problem:

$$\min_{\boldsymbol{w}, b, \xi_i} \ \tfrac{1}{2}\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w} + C\sum \xi_i$$

subject to:

$$\forall d_i \in Q \mid f_i(e_j) > f_i(e_k) : \boldsymbol{w}^{\mathrm{T}}\Omega(d_i) + b \geq 1 - \xi_i$$
$$\forall d_i \in Q \mid f_i(e_j) < f_i(e_k) : -(\boldsymbol{w}^{\mathrm{T}}\Omega(d_i) + b) \geq 1 - \xi_i$$
$$\forall i : \xi_i \geq 0$$

where C is the SVM cost parameter, ξ_i are slack variables, and Q is the training corpus. We assume each document $d_i \in Q$ is labeled with $f_i(e_j)$ for every emotion $e_j \in E$.

When formulated as an SVM optimization problem, finding g_{jk} is equivalent to training an SVM classifier for classifying a document into the e_j or e_k category. Hence, we use LIBSVM, which is an SVM implementation, to obtain the solution.[1]

[1] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

Input: Set of emotion ordered pairs P
1. $G \leftarrow$ a graph with emotions as vertices and no edge
2. **while** ($P \neq \varnothing$)
3. remove (e_j,e_k) with the highest confidence from P
4. **if** adding edge (e_j,e_k) to G produces a loop
5. **then** add (e_k,e_j) to G
6. **else** add (e_j,e_k) to G
7. **return** topological sort of G

Algorithm 1. Merge Pairwise Orders.

We now describe how we rank the emotions of a previously unseen document using the $M(M-1)/2$ pairwise ranking functions g_{jk} created during the training phase. First, all of the pairwise ranking functions are applied to the unseen document, which generates the relative orders of every pair of emotions. These pairwise orders need to be combined together to produce a ranked list of all the emotions. Algorithm 1 does exactly this.

In Algorithm 1, the confidence of an emotion ordered pair at Line 3 is the probability value returned by a LIBSVM classifier for predicting the order. LIBSVM's method for generating this probability is described in (Wu et al., 2003). Lines 4 and 5 resolve the problem of conflicting emotion ordered pairs forming a loop in the ordering of emotions. The ordered list of emotions returned by Algorithm 1 at Line 7 is the final output of the PLM method.

3.3 Emotional Distribution Regression

In the second approach to ranking emotions, we use regression to model f_i directly. A regression function $h_j : D \rightarrow \Re$ is generated for each $e_j \in E$ by learning from the examples $(\Omega(d_i), f_i(e_j))$ for all documents d_i in the training corpus.

The regression framework we adopt is Support Vector Regression (SVR), which is a regression analysis technique based on SVM (Schölkopf et al., 2000). We require h_j to have the form $\boldsymbol{w}^{\mathrm{T}}\Omega(d_i) + b$. Finding h_j is equivalent to solving the following optimization problem:

$$\min_{\boldsymbol{w}, b, \xi_{i,1}, \xi_{i,2}} \ \tfrac{1}{2}\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w} + C\sum(\xi_{i,1} + \xi_{i,2})$$

subject to:

$$\forall d_i \in Q :$$
$$f_i(e_j) - (\boldsymbol{w}^{\mathrm{T}}\Omega(d_i) + b) \geq \varepsilon - \xi_{i,1}$$
$$(\boldsymbol{w}^{\mathrm{T}}\Omega(d_i) + b) - f_i(e_j) \geq \varepsilon - \xi_{i,2}$$
$$\forall i : \xi_{i,1}, \xi_{i,2} \geq 0$$

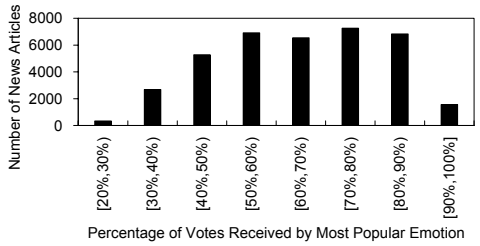

Figure 2. News articles in the entire corpus grouped by the percentage of votes received by the most popular emotion.

where C is the cost parameter, ε is the maximum difference between the predicted and actual values we wish to maintain, $\xi_{i,1}$ and $\xi_{i,2}$ are slack variables, and Q is the training corpus. To solve the above optimization problem, we use SVM^{light}'s SVR implementation.[2]

When ranking the emotions of a previously unseen document d_k, we sort the emotions $e_j \in E$ in descending order of $h_j(d_k)$.

4 Constructing the Corpus

The training and test corpora used in this study comprise Chinese news articles from Yahoo! Kimo News[3], which allows a user to cast a vote for one of eight emotions to express how a news article makes her feel. Each Yahoo! news article contains a list of eight emotions at the bottom of the webpage. A reader may select one of the emotions and click on a submit button to submit the emotion. As with many websites which collect user responses, such as the Internet Movie Database, users are not forced to submit their responses. After submitting a response, the user can view a distribution of emotions indicating how other readers feel about the same article. Figure 1 shows the voting results of a Yahoo! news article.

The eight available emotions are *happy*, *sad*, *angry*, *surprising*, *boring*, *heartwarming*, *awesome*, and *useful*. *Useful* is not a true emotion. Rather, it means that a news article contains practical information. The value $f_i(e_j)$ is derived by normalizing the number of votes for emotion e_j in document d_i by the total number votes in d_i.

The entire corpus consists of 37,416 news articles dating from January 24, 2007 to August 7, 2007. News articles prior to June 1, 2007 form the training corpus (25,975 articles), and the remaining ones form the test corpus (11,441 articles). We collect articles a week after their publication dates to ensure that the vote counts have stabilized.

As mentioned earlier, readers rarely agree unanimously on the emotion of a document. Figure 2 illustrates this. In 41% of all the news articles in the entire corpus, the most popular emotion receives less than 60% of the votes.

5 Extracting Features

After obtaining news articles, the next step is to determine how to convert them into feature vectors for SVM and SVR. That is, we want to instantiate Ω. For this purpose, three types of features are extracted.

The first feature type consists of Chinese character bigrams, which are taken from the headline and content of each news article. The presence of a bigram is indicated by a binary feature value.

Chinese words form the second type of features. Unlike English words, consecutive Chinese words in a sentence are not separated by spaces. To deal with this problem, we utilize Stanford NLP Group's Chinese word segmenter to split a sentence into words.[4] As in the case of bigrams, binary feature values are used.

We use character bigram features in addition to word features to increase the coverage of Chinese words. A Chinese word is formed by one or more contiguous Chinese characters. As mentioned earlier, Chinese words in a sentence are not separated by any boundary symbol (e.g., a space), so a Chinese word segmentation tool is always required to extract words from a sentence. However, a word segmenter may identify word boundaries erroneously, resulting in the loss of correct Chinese words. This problem is particularly severe if there are a lot of out-of-vocabulary words in a dataset. In Chinese, around 70% of all Chinese words are Chinese character bigrams (Chen et al., 1997). Thus, using Chinese character bigrams as features will allow us to identify a lot of Chinese words, which when combined with the words extracted by the word segmenter, will give us a wider coverage of Chinese words.

The third feature type is extracted from news metadata. A news article's metadata are its news

[2] http://svmlight.joachims.org/
[3] http://tw.news.yahoo.com
[4] http://nlp.stanford.edu/software/segmenter.shtml

category, agency, hour of publication, reporter, and event location. Examples of news categories include sports and political. Again, we use binary feature values. News metadata are used because they may contain implicit emotional information.

6 Experiments

The experiments are designed to achieve the following four goals: (i) to compare the ranking performance of different methods, (ii) to analyze the pairwise ranking quality of PLM, (iii) to analyze the distribution estimation quality of EDR, and (iv) to compare the ranking performance of different feature sets. The Yahoo! News training and test corpora presented in Section 4 are used in all experiments.

6.1 Evaluation Metrics for Ranking

We employ three metrics as indicators of ranking quality: ACC@*k*, NDCG@*k* and SACC@*k*.

ACC@*k* stands for accuracy at position *k*. According to ACC@*k*, a predicted ranked list is correct if the list's first *k* items are identical (i.e., same items in the same order) to the true ranked list's first *k* items. If two emotions in a list have the same number of votes, then their positions are interchangeable. ACC@*k* is computed by dividing the number of correctly-predicted instances by the total number of instances.

NDCG@*k*, or normalized discounted cumulative gain at position *k* (Järvelin and Kekäläinen, 2002), is a metric frequently used in information retrieval to judge the quality of a ranked list when multiple levels of relevance are considered. This metric is defined as

$$\mathrm{NDCG}@k = z_k \sum_{i=1}^{k} \frac{rel_i}{\log_2(i+1)}$$

where rel_i is the relevance score of the predicted item at position i, and z_k is a normalizing factor which ensures that a correct ranked list has an NDCG@*k* value of 1. In the emotion ranking problem, rel_i is the percentage of reader votes received by the emotion at position i. Note that the $\log_2(i+1)$ value in the denominator is a discount factor which decreases the weights of items ranked later in a list. NDCG@*k* has the range [0, 1], where 1 is the best. In the experiment results, NDCG@*k* values are averaged over all instances in the test corpus.

NDCG@*k* is used because ACC@*k* has the disadvantage of not taking emotional distributions into account. Take Figure 1 as an example. In the figure, *heartwarming* and *happy* have 31.3% and 30.7% of the votes, respectively. Since the two percentages are very close, it is reasonable to say that predicting *happy* as the first item in a ranked list may also be acceptable. However, doing so would be completely incorrect according to ACC@*k*. In contrast, NDCG@*k* would consider it to be partially correct, and the extent of correctness depends on how much *heartwarming* and *happy*'s percentages of votes differ. To be exact, if *happy* is predicted as the first item, then the corresponding NDCG@1 would be 30.7% / 31.3% = 0.98.

The third metric is SACC@*k*, or set accuracy at *k*. It is a variant of ACC@*k*. According to SACC@*k*, a predicted ranked list is correct if the set of its first *k* items is the same as the true ranked list's set of first *k* items. In effect, SACC@*k* evaluates a ranking method's ability to place the top *k* most important items in the first *k* positions.

6.2 Tuning SVM and SVR Parameters

SVM and SVR are employed in PLM and EDR, respectively. Both SVM and SVR have the adjustable *C* cost parameter, and SVR has an additional ε parameter. To estimate the optimal *C* value for a combination of SVM and features, we perform 4-fold cross-validation on the Yahoo! News training corpus, and select the *C* value which results in the highest binary classification accuracy during cross-validation. The same procedure is used to estimate the best *C* and ε values for a combination of SVR and features. The *C*-ε pair which results in the lowest mean squared error during cross-validation is chosen. The candidate *C* values for both SVM and SVR are 2^{-10}, 2^{-9}, …, 2^{-6}. The candidate ε values for SVR are 10^{-2} and 10^{-1}. All cross-validations are performed solely on the training data. The test data are not used to tune the parameters. Also, SVM and SVR allow users to specify the type of kernel to use. Linear kernel is selected for both SVM and SVR.

6.3 Nearest Neighbor Baseline

The nearest neighbor (NN) method is used as the baseline. The ranked emotion list of a news article in the test corpus is predicted as follows. First, the

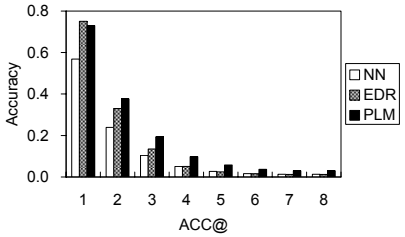

Figure 3. ACC@k

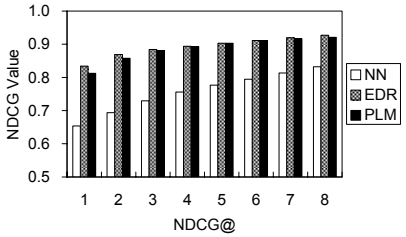

Figure 4. NDCG@k

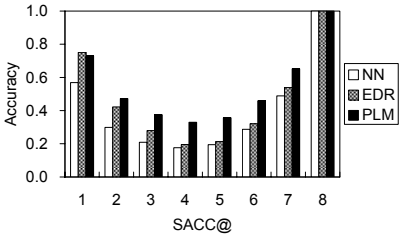

Figure 5. SACC@k

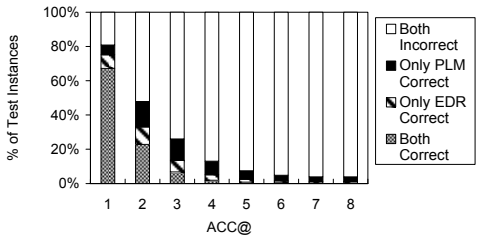

Figure 6. Performance of PLM and EDR.

test news article is compared to every training news article using cosine similarity, which is defined as

$$\cos(d_i, d_j) = \frac{|D_i \cap D_j|}{\sqrt{|D_i| \times |D_i|}}$$

where d_i and d_j are two news articles, and D_i and D_j are sets of Chinese character bigrams in d_i and d_j, respectively. The ranked emotion list of the training article having the highest cosine similarity with the test article is used as the predicted ranked list.

6.4 Comparison of Methods

Figures 3 to 5 show the performance of different ranking methods on the test corpus. For both PLM and EDR, all of the bigram, word, and news metadata features are used.

In Figure 3, EDR's ACC@1 (0.751) is higher than those of PLM and NN, and the differences are statistically significant with p-value < 0.01. So, EDR is the best method at predicting the most popular emotion. However, PLM has the best ACC@k for $k \geq 2$, and the differences from the other two methods are all significant with p-value < 0.01. This means that PLM's predicted ranked lists better resemble the true ranked lists.

Figure 3 displays a sharp decrease in ACC@k values as k increases. This trend indicates the hardness of predicting a ranked list correctly. Looking from a different angle, the ranking task under the ACC@k metric is equivalent to the classification of news articles into one of $8!/(8-k)!$ classes, where we regard each unique emotion sequence of length k as a class. In fact, computing ACC@8 for a ranking method is the same as evaluating the method's ability to classify a news article into one of 8! = 40,320 classes. So, producing a completely-correct ranked list is a difficult task.

In Figure 4, all of PLM and EDR's NDCG@k improvements over NN are statistically significant with p-value < 0.01. For some values of k, the difference in NDCG@k between PLM and EDR is not significant. The high NDCG@k values (i.e., greater than 0.8) of PLM and EDR imply that although it is difficult for PLM and EDR to generate completely-correct ranked lists, these two methods are effective at placing highly popular emotions to the beginning of ranked lists.

In Figure 5, PLM outperforms the other two methods for $2 \leq k \leq 7$, and the differences are all statistically significant with p-value < 0.01. For small values of k (e.g., $2 \leq k \leq 3$), PLM's higher SACC@k values mean that PLM is better at placing the highly popular emotions in the top positions of a ranked list.

To further compare PLM and EDR, we examine their performance on individual test instances. Figure 6 shows the percentage of test instances where both PLM and EDR give incorrect lists, only PLM gives correct lists, only EDR gives ranked lists, and both methods give correct lists. The "Only PLM Correct" and "Only EDR Correct" categories are nonzero, so neither PLM nor EDR is always better than the other.

In summary, EDR is the best at predicting the most popular emotion according to ACC@1, NDCG@1 and SACC@1. However, PLM generates ranked lists that better resemble the correct ranked lists according to ACC@k and SACC@k

Method	Average τ_b	Average p-value
PLM	0.584	0.068
EDR	0.474	0.114
NN	0.392	0.155

Table 1. Kendall's τ_b statistics.

	He	Su	Sa	Us	Ha	Bo	An
Aw	0.80	0.75	0.78	0.77	0.82	0.76	0.79
He		0.79	0.81	0.78	0.81	0.89	0.81
Su			0.82	0.78	0.80	0.82	0.82
Sa				0.78	0.80	0.84	0.82
Us					0.82	0.91	0.82
Ha						0.83	0.79
Bo							0.80

Table 2. Classification accuracies of SVM pairwise emotion classifiers on the test corpus. He = *heartwarming*, Su = *surprising*, Sa = *sad*, Us = *useful*, Ha = *happy*, Bo = *boring*, and An = *angry*.

Figure 7. Accuracy of pairwise emotion classification and the corresponding average discrimination value.

for $k \geq 2$. Further analysis shows that neither method is always better than the other.

6.5 Pairwise Ranking Quality of PLM

In this subsection, we evaluate the performance of PLM in predicting pairwise orders.

We first examine the quality of ranked lists generated by PLM in terms of pairwise orders. To do this, we use Kendall's τ_b correlation coefficient, which is a statistical measure for determining the correlation between two ranked lists when there may be ties between two items in a list (Liebetrau, 1983). The value of τ_b is determined based on the number of concordant pairwise orders and the number of discordant pairwise orders between two ranked lists. Therefore, this measure is appropriate for evaluating the effectiveness of PLM at predicting pairwise orders correctly. τ_b has the range [-1, 1], where 1 means a perfect positive correlation, and -1 means two lists are the reverse of each other. When computing τ_b of two ranked lists, we also calculate a p-value to indicate whether the correlation is statistically significant.

We compute τ_b statistics between a predicted ranked list and the corresponding true ranked list. Table 1 shows the results. In Table 1, numbers in the "Average τ_b" and "Average p-value" columns are averaged over all test instances. The statistics for EDR and NN are also included for comparison. From the table, we see that PLM has the highest average τ_b value and the lowest average p-value, so PLM is better at preserving pairwise orders than EDR and NN methods. This observation verifies that PLM's minimization of pairwise loss leads to better prediction of pairwise orders.

We now look at the individual performance of the 28 pairwise emotion rankers g_{jk}. As mentioned in Section 3.2, each pairwise emotion ranker g_{jk} is equivalent to a binary classifier for classifying a document into the e_j or e_k category. So, we look at their classification accuracies in Table 2. In the table, accuracy ranges from 0.75 for the *awesome-surprising* pair to 0.91 for the *useful-boring* pair.

From the psychological perspective, the relatively low accuracy of the *awesome-surprising* pair is expected, because *awesome* is *surprising* in a positive sense. So, readers should have a hard time distinguishing between these two emotions. And the SVM classifier, which models reader responses, should also find it difficult to discern these two emotions. Based on this observation, we suspect that the pairwise classification performance actually reflects the underlying emotional ambiguity experienced by readers. To verify this, we quantify the degree of ambiguity between two emotions, and compare the result to pairwise classification accuracy.

To quantify emotional ambiguity, we introduce the concept of discrimination value between two emotions e_j and e_k in a document d_i, which is defined as follows:

$$\frac{\left|f_i(e_j) - f_i(e_k)\right|}{f_i(e_j) + f_i(e_k)}$$

where f_i is the emotional probability function defined in Section 3.1. Intuitively, the larger the discrimination value is, the smaller the degree of ambiguity between two emotions is.

Figure 7 shows the relationship between pairwise classification accuracy and the average discrimination value of the corresponding emotion

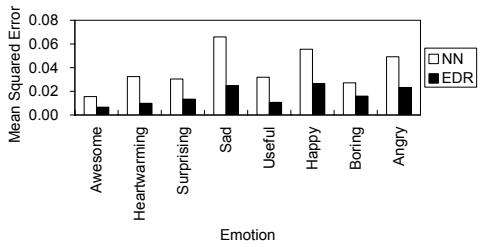

Figure 8. Mean squared error of NN and EDR for estimating the emotional distributions of the test corpus.

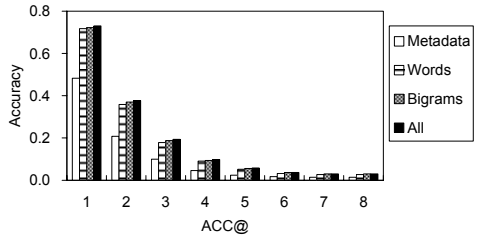

Figure 9. PLM performance using different features.

pair. The general pattern is that as accuracy increases, the discrimination value also increases. To provide concrete evidence, we use Pearson's product-moment correlation coefficient, which has the range of [-1, 1], where 1 means a perfect positive correlation (Moore, 2006). The coefficient for the data in Figure 7 is 0.726 with p-value < 0.01. Thus, pairwise emotion classification accuracy reflects the emotional ambiguity experienced by readers.

In summary, PLM's pairwise loss minimization leads to better pairwise order predictions than EDR and NN. Also, the pairwise classification results reveal the inherent ambiguity between emotions.

6.6 Distribution Estimation Quality of EDR

In this subsection, we evaluate EDR's performance in estimating the emotional probability function f_i.

With the prior knowledge that a news article's f_i values sum to 1 over all emotions, and f_i is between 0 and 1, we adjust EDR's f_i predictions to produce proper distributions. It is done as follows. A predicted f_i value greater than 1 or less than 0 is set to 1 and 0, respectively. Then the predicted f_i values are normalized to sum to 1 over all emotions.

NN's distribution estimation performance is included for comparison. For NN, the predicted f_i values of a test article are taken from the emotional distribution of the most similar training article.

Figure 8 shows the mean squared error of EDR and NN for predicting f_i. In the figure, the error generated by EDR is less than those by NN, and all the differences are statistically significant with p-value < 0.01. Thus, EDR's use of regression leads to better estimation of f_i than the NN.

6.7 Comparison of Features

Figure 9 shows each of the three feature type's ACC@k for predicting test instances' ranked lists when PLM is used. The feature comparison graph for EDR is not shown, because it exhibits a very similar trend as PLM. For both PLM and EDR, bigrams are better than words, which are in turn better than news metadata. In Figure 9, the combination of all three feature sets achieves the best performance. For both PLM and EDR, the improvements in ACC@k of using all features over words and metadata are all significant with p-value < 0.01, and the improvements over bigrams are significant for $k \leq 2$. Hence, in general, it is better to use all three feature types together.

7 Conclusions and Future Work

This paper presents two methods to ranking reader emotions. The PLM method minimizes pairwise loss, and the EDR method estimates emotional distribution through regression. Experiments with significant tests show that EDR is better at predicting the most popular emotion, but PLM produces ranked lists that have higher correlation with the correct lists. We further verify that PLM has better pairwise ranking performance than the other two methods, and EDR has better distribution estimation performance than NN.

As for future work, there are several directions we can pursue. An observation is that PLM exploits pairwise order information, whereas EDR exploits emotional distribution information. We plan to combine these two methods together. Another research direction is to improve EDR by finding better features. We would also like to integrate emotion ranking into information retrieval.

Acknowledgments

We are grateful to the Computer and Information Networking Center, National Taiwan University, for the support of high-performance computing facilities. The research in this paper was partially supported by National Science Council, Taiwan, under the contract NSC 96-2628-E-002-240-MY3.

References

Saima Aman and Stan Szpakowicz. 2007. *Identifying Expressions of Emotion in Text*. In Proceedings of 10th International Conference on Text, Speech and Dialogue, Lecture Notes in Computer Science 4629, 196-205. Springer, Plzeň, CZ.

Aitao Chen, Jianzhang He, Liangjie Xu, Frederic Gey, and Jason Meggs. 1997. *Chinese Text Retrieval wihtout using a Dictionary*. In Proceedings of 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 42-49. Association for Computing Machinery, Philadelphia, US.

Yoav Freund, Raj D. Iyer, Robert E. Schapire, and Yoram Singer. 2003. *An Efficient Boosting Algorithm for Combining Preferences*. Journal of Machine Learning Research, 4, 933-969.

Yi Hu, Jianyong Duan, Xiaoming Chen, Bingzhen Pei, and Ruzhan Lu. 2005. *A New Method for Sentiment Classification in Text Retrieval*. In Proceedings of 2nd International Joint Conference on Natural Language Processing, 1-9. Jeju Island, KR.

Kalervo Järvelin and Jaana Kekäläinen. *Cumulative Gain-based Evaluation of IR Techniques*. 2002. ACM Transactions on Information Systems, 20(4), 422-446.

Thorsten Joachims. 2002. *Optimizing Search Engines using Clickthrough Data*. In Proceedings of 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Association for Computing Machinery, Edmonton, CA.

Albert M. Liebetrau. 1983. *Measures of Association*. Sage Publications, Newbury Park, US.

Kevin H. Lin, Changhua Yang, and Hsin-Hsi Chen. 2007. *What Emotions do News Articles Trigger in their Readers?* In Proceedings of 30th ACM SIGIR Conference, 733-734. Association for Computing Machinery, Amsterdam, NL.

Kevin H. Lin, Changhua Yang, and Hsin-Hsi Chen. 2008. *Emotion Classification of Online News Articles from the Reader's Perspective*. In Proceedings of International Conference on Web Intelligence. Institute of Electrical and Electronics Engineers, Sydney, AU.

David Moore. 2006. *The Basic Practice of Statistics*. W.H. Freeman and Company, New York, US.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. *Thumbs up? Sentiment Classification Using Machine Learning Techniques*. In Proceedings of 2002 Conference on Empirical Methods in Natural Language Processing, 79-86. Association for Computational Linguistics, Philadelphia, US.

Tao Qin, Tie-Yan Liu, Wei Lai, Xu-Dong Zhang, De-Sheng Wang, and Hang Li. 2007. *Ranking with Multiple Hyperplanes*. In Proceedings of 30th ACM SIGIR Conference, 279-286. Association for Computing Machinery, Amsterdam, NL.

Bernhard Schölkopf, Alex J. Smola, Robert C. Williamson, and Peter L. Barlett. 2000. *New Support Vector Algorithms*. Neural Computation, 12(5), 1207-1245.

Carlo Strapparava and Rada Mihalcea. 2007. *SemEval-2007 Task 14: Affective Text*. In Proceedings of 4th International Workshop on Semantic Evaluations. Prague, CZ.

Janyce M. Wiebe. 2000. *Learning Subjective Adjectives from Corpora*. In Proceedings of 17th Conference of the American Association for Artificial Intelligence, 735-740. AAAI Press, Austin, US.

Ting-Fan Wu, Chih-Jen Lin, and Ruby C. Weng. *Probability Estimates for Multi-class Classification by Pairwise Coupling*. 2004. Journal of Machine Learning Research, 5, 975-1005.