Embarrassed or Awkward?
Ranking Emotion Synonyms for ESL Learners' Appropriate Wording

Wei-Fan Chen
Academia Sinica
viericwf@iis.sinica.edu.tw

Mei-Hua Chen
Department of Foreign Language and Literature,
Tunghai University
chen.meihua@gmail.com

Lun-Wei Ku
Academia Sinica
lwku@iis.sinica.edu.tw

Abstract

We introduce a novel framework based on the probabilistic model for emotion wording assistance. The example sentences from the online dictionary, *Vocabulary.com* are utilized as the training data; and the writings in a designed ESL's writing task are the testing corpus. The emotion events are captured by extracting patterns of the example sentences. Our approach learns the joint probability of contextual emotion events and the emotion words from the training corpus. After extracting patterns in the testing corpus, we then aggregate their probabilities to suggest the emotion word that describes the ESL's context most appropriately. We evaluate the proposed approach by the NDCG@5 of the suggested words for the writings in the testing corpus. The experiment result shows our approach can more appropriately suggest the emotion words compared to SVM, PMI and two representative on-line reference tools, PIGAI and *Thesaurus.com*.

1 Introduction

Most English-as-a-second-language (ESL) learners have been found to have difficulties in emotion vocabulary (Pavlenko, 2008). With limited lexical knowledge, learners tend to use common emotion words such as angry and happy to describe their feelings. Moreover, the learner's first language seems to lead to inappropriate word choices (Altarriba and Basnight-Brown, 2012). Many learners consult the thesaurus for synonyms of emotion words; typically, the synonyms suggested come with little or no definition or usage information. Moreover, the suggested synonyms seldom take into account contextual information. As a result, the thesaurus does not always help language learners select appropriately nuanced emotion words, and can even mislead learners into choosing improper words that sometimes convey the wrong message (Chen *et al.*, 2013). Take *embarrassed* and *awkward* for example: although they both describe situations where people feel uneasy or uncomfortable, in practice they are used in different scenarios. According to *Vocabulary.com*, *embarrassed* is more self-conscious and can result from shame or wounded pride: for instance, *He was too embarrassed to say hello to his drunken father on the street*. On the other hand, *awkward* would be "socially uncomfortable" or "unsure and constrained in manner": *He felt awkward and reserved at parties.* These examples illustrate not only the nuances between synonymous emotion words, but also the difficulty for language learners in determining proper words. There is a pressing need for a reference resource providing a sufficient number of emotion words and their corresponding usage information to help language learners expand their knowledge of emotion words and learn proper emotional expressions.

To address this issue, we propose a novel approach to help differentiate synonyms of emotion words based on contextual clues—Ranking Emotional SynOnyms for language Learners' Vocabulary Expansion (RESOLVE). This involves first the learning of emotion event scores between an event and an emotion word from a corpus: these

Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications, 2015, pages 144–153,
Denver, Colorado, June 4, 2015. ©2015 Association for Computational Linguistics

scores quantify how appropriate an emotion word is to describe a given event. Subsequently, based on the emotion event in the learner's text, RESOLVE suggests a list of ranked emotion words.

2 Related Work

Previous studies related to RESOLVE can be divided into four groups: **paraphrasing**, **emotion classification**, **word suggestion** and **writing assessment**. The aim of paraphrasing research is how to express the same information in various ways. Such alternative expressions of the same information rely on paraphrase pairs which map an expression to a previously learned counterpart, or inference rules that re-structure the original sentences. Most work uses machine translation techniques such as statistical machine translation or multiple-sequence alignment to extract paraphrase pairs from monolingual corpora (Barzilay and McKeown, 2001; Keshtkar and Inkpen, 2010), or bilingual corpora (Bannard and Callison-Burch, 2005; Callison-Burch, 2008; Chen *et al.*, 2012). Approaches based on inference rules, on the other hand, derive these rules by analyzing the dependency relations of paraphrase sentences (Lin and Pantel, 2001; Dinu and Lapata, 2010). Alternative expressions can be achieved by applying inference rules to rephrase the original sentence. In general, the focus of paraphrasing is sentence variation, which involves sentence re-structuring, phrase alternation and word substitution. Generating an alternative sentence without changing the sentence's original meaning is the main concern. For RESOLVE, in contrast, rather than attempting preservation, the focus is on appropriate in-context word substitution. There are several online paraphrasing tools. PREFER[1] (Chen *et al.*, 2012) is an online paraphrase reference interface that generates phrase-level paraphrases using a combination of graph and PageRank techniques. Chen shows a significant improvement in learner paraphrase writing performance. For RESOLVE, instead of pursuing participant improvements in semantically equivalent rephrasing, the aim is to suggest contextually appropriate wording. Microsoft Contextual Thesaurus[2] (MCT) is similar to PREFER: it is an online reference tool that smartly rephrases an input sentence into various alternative expressions, using both word-level and phrase-level substitution. However, we know of no study that evaluates learning effectiveness when using MCT. Finally, SPIDER (Barreiro, 2011) targets document-level editing; it relies on derivations from dictionaries and grammars to paraphrase sentences, aiming at reducing wordiness and clarifying vague or imprecise terms. In short, rather than offering better suggestions, paraphrasing tools provide equivalent expressions.

Emotion classification concerns approaches to detect the underlying emotion of a text. Related work typically attempts this using classifiers. These classifiers are trained with features such as n-grams (Tokuhisa *et al.*, 2008), word-level pointwise mutual information (PMI) values (Agrawal *et al.*, 2012; Bullinaria *et al.*, 2007; and Church *et al.*, 1990) or a combination of word POS and sentence dependency relations (Ghazi *et al.*, 2012). The remained works of emotion classification in above mentioned research to deal with emotions aroused by events inspires us to relate events to emotion words in RESOLVE. In addition, in terms of emotion classification, RESOLVE classifies texts into fine-grained classes where each emotion word can be viewed as a single class; in contrast, most emotion classification work focuses only on coarse-grained (6 to 10 classes) emotion labeling. It is a challenging work.

Word suggestion involves guessing a possible replacement for a given word in a sentence, or finding word collocations. A representative research task for word suggestion is the SemEval 2007 English Lexical Substitution task: the problem is to find a word substitute for the designated word given a sentence. Zhao *et al.* (2007) first uses rules to find possible candidates from WordNet and verifies the sentence after substitution using Web search results; Dahl *et al.* (2007) utilizes a more traditional n-gram model but uses statistics from web 5-grams. Although closely related to our work, this task is different in several ways. First, the word for which a substitute is required is already an appropriate word, as it appears in a sentence from a well-written English corpus, the Internet Corpus of English[3]. However, the goal of our work is to determine whether a word selected by ESL learners is appropriate, and if necessary to

[1] http://service.nlpweb.org/PREFER

[2] http://labs.microsofttranslator.com/thesaurus/

[3] http://corpus.leeds.ac.uk/internet.html

suggest appropriate alternatives. Observation of our corpus has shown that typically, the word selected by ESL learners is not the most appropriate one. This is in contrast to the cited related works in which the original in-context wording is usually the most appropriate one. However, in our research the context often does not support the way the ESL learner's word(s) are used. Second, the context of the given word in SemEval is a sentence, while in this work it is a document. Third, annotators for SemEval were limited to at most three possible substitutions, all of which were to be equally appropriate, while in our work annotators are asked to assign ranks to all candidates (synonyms of the given word). Fourth, in SemEval the words to be substituted come from various syntactic or semantic categories, while we only suggest appropriate emotion words to the learners.

For writing assessment, existing works are known as automatic essay assessment (AEA) systems, which analyze user compositions in terms of wording, grammar and organization. PIGAI[4], targeted at generating suggested revisions, suggests unranked synonyms for words. However, unranked synonyms easily confuse Chinese learners (Ma, 2013). E-rater (Leading *et al.* 2005), a writing evaluation system developed by the Educational Testing Service (ETS), offers a prompt-specific vocabulary usage score, a scoring feature which evaluates the word choice and compares words in the writing with samples in low- to high-quality writings. Ma shows that students' scores on PIGAI increase after using PIGAI, and that these results are in proportion to the frequency they use PIGAI. As for E-rater, to our best knowledge, its focus is on helping judges to score writing rather than on assisting learners. In contrast, the purpose of RESOLVE is to directly assist language learners in finding appropriate wording, especially for emotion words.

As context and events are crucial to appropriate emotion wording, both have been taken into account in the development of RESOLVE. For context, learner writings describing emotions have been utilized to extract contextual clues. For events, Jeong and Myaeng (2012) find that in the well-annotated TimeBank corpus, 65% of the event conveyance was accomplished using verbs; thus we detect events from verb phrases. In contrast to paraphrasing and emotion analysis, the goal of RESOLVE is to distinguish the nuances among emotion synonyms in order to aid in language learning: this makes it a novel research problem.

3 Method

We postulate that patterns can describe emotion events, and that event conveyance is accomplished primarily using verbs (Jeong and Myaeng, 2012). In RESOLVE, verb-phrase patterns are selected for use in differentiating emotion word synonyms, that is, candidate emotion words, using their relationships with these patterns. Figure 1 shows the two-stage RESOLVE framework. First we learn corpus patterns (patterns extracted from the corpus) and their emotion event scores, *EES*, for all emotion words, and then, given the target emotion word, we rank the candidate emotion words to suggest appropriate wording using the writing patterns (patterns extracted from learner writing) and associated emotion event scores learned in the first stage. To determine the similarity between corpus patterns and writing patterns, we also propose a pattern matching algorithm which takes into account the cost of wildcard matching. Finally, to verify the effectiveness of RESOLVE in aiding precise wording, a learning experiment is designed. In an example RESOLVE scenario, the learner writes the following: "I love guava but one day I ate a rotten guava with a maggot inside, which made me **disgust**." She is not sure about the wording so she turns to RESOLVE for help. She is given a ranked word suggestion list: *repugnance*, *disgust*, *repulsion*, *loathing* and *revulsion*; which are more appropriate than the list *Theasurus.com* provides: *antipathy*, *dislike*, *distaste*, *hatred* and *loathing*.

[4] http://www.pigai.org

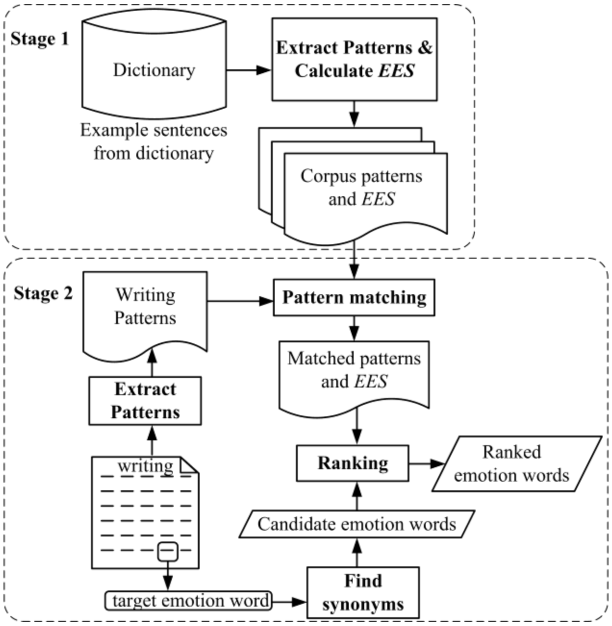

Figure 1: RESOLVE framework.

3.1 Stage One: Learning Corpus Patterns for All Emotion Words

In this stage, we learn patterns and their relations to emotion words from the corpus. Sentences are first pre-processed, after which patterns are extracted from the corpus and their emotion event scores calculated.

Pre-processing. As compound sentences can be associated with more than one emotion event, they must be pre-processed before we extract patterns. Compound sentences are first broken into clauses according to the Stanford phrase structure tree output (Klein and Manning, 2003). In the experiments, these clauses are treated as sentences.

Pattern Extraction. Emotion events are characterized by verb-phrase patterns, derived from the output of the Stanford dependency parser (De Marneffe *et al.*, 2006). This parser generates the grammatical relations of word pairs and determines the ROOT, which is often the verb, after parsing each sentence. We describe the extraction steps given the sentence "*We gave the poor girl a new book*.". A total of 746,919 patterns were extracted in this process.

Step1: Identify the ROOT (*gave*) and all its dependents based on the parsing result.

Step2: Replace the words having no dependency relation to the *ROOT* with wildcards; consecutive wildcards were combined into one. (*we gave * girl * book*)

Step3: Filter out less informative dependents (i.e., those nodes that are not the children of the ROOT in the dependency parse tree) by replacing with wildcards the dependents in the following relations to the ROOT: *subj, partmod, comp, parataxis, advcl, aux, poss, det, cc, advmod* and *dep*. (* *gave * girl * book*)

Step4: Generalize the grammatical objects by replacing them with their top-level semantic class in *WordNet*. (* *gave * <person> * <artifact>*)

Step5: Lemmatize the verbs using *WordNet*. (* *give * <person> * <artifact>*)

Step6: Removing the starting and ending wildcards. (*give * <person> * <artifact>*)

Emotion Event Score Calculation. Once the patterns are extracted, RESOLVE learns their emotion event scores (EES) to quantize their relations to each emotion word. Here we discover an interesting issue: the extracted pattern may summarize an emotion event, but it may also tell the emotion it bears directly with emotion words. To determine whether patterns containing emotion words have different characteristics and effects on performance, we term them self-containing patterns. Hence two pattern sets are used in experiments: one that includes all extracted patterns (P_{all}), and the other that excludes all self-containing patterns ($P_{-scPattern}$).

As shown in equation (1), we define the emotion event score ($EES_{p,e}$) of a pattern *p* for an emotion word *e* by the conditional probability of *e* given *p*.

$$EES_{p,e} = P(e \mid p) \qquad (1)$$

3.2 Stage Two: Ranking Synonyms of the Emotion Word to Suggest Appropriate Wording

In previous stage we built a pattern set for each emotion word. In this stage, there are four tasks: enumerate candidate emotion words for the target emotion word, extract writing patterns, match the writing patterns to the corpus patterns of the candidates, and rank the candidates. To enumerate the candidate emotion words, RESOLVE first looks up synonyms of the target emotion word in *WordNetSynsets* and in *Merriam Webster's Dictionary*.

Pattern Matching. For each candidate e_i, RESOLVE compares writing patterns $P_{writing}=(pw_1,pw_2,...pw_N)$ with corpus patterns P_{corpus}, and returns the matching corpus patterns $P_{match}=(p_1,p_2,...p_N)$ and their corresponding emotion event scores; where N is number of clauses in the writing. Edit distance is utilized to calculate the similarity between a writing pattern and a corpus pattern, where the matching corpus pattern is defined as that with the maximum pattern similarity to the writing pattern (in a one-to-one matching). The emotion scores of this matched corpus pattern for different emotion words will be used as the writing pattern scores.

We propose a variation of edit distance which accepts wildcards (that is, edit-distance with wildcards, *EDW*) that allows for partial matching, including similar patterns, and hence increases hit rates. Therefore, we add a *wildcard replacement cost* (WRC) to the edit cost (*general edit cost*, GEC) in the traditional definition of the edit distance. For this purpose, a two-dimensional vector (GEC, WRC) which considers two edit costs separately is used to measure the *EDW* between patterns S_1 and S_2. *EDW* is defined as

$$EDW\left(S_1 \rightarrow S_2\right) = D_{I,J} = \left(GEC,\ WRC\right) \qquad (2)$$

where $S_1=\{s_1(1), s_1(2), s_1(3),...,s_1(I)\}$ and $S_2=\{s_2(1), s_2(2), s_2(3),..., s_2(J)\}$ are tokens of the corpus pattern S1 and the writing pattern S_2; I and J are the lengths of S_1 and S_2; i and j are the indices of S_1 and S_2; and $D_{I,J}$ is recursively calculated from 1 to I and 1 to J using the edit distance formula. Note that $D_{0,0}$ is (0,0). A wildcard may be replaced with one or more tokens, or vice versa. When calculating *EDW*, if there is a wildcard replacement, the replacement cost is added to the WRC; for other cases, the edit cost is added to the GEC.

We here define the value of the WRC. The traditional edit-distance algorithm takes into account only single-token costs, whereas wildcards in our patterns may replace more than one token. Wildcard insertion and deletion costs hence depend on the number of tokens a wildcard may replace. After some experiments, we empirically choose e (Euler's number) as the cost of wildcard insertion and deletion. Note that e is also very close to the mean of the number of words replaced by one wildcard (positively skewed distribution). Table 1 shows the costs of all EDW operations.

Operation	Cost
Wildcard Insertion (ø → *)	e
Wildcard Deletion (* →ø)	e
Wildcard Replacement (* →token)	1
Wildcard Replacement (token → *)	e

Table 1: Edit distance costs for EDW operations.

Empirically, if no exact pattern is found, to represent the pattern we seek a more general pattern rather than a specific one. A general pattern's meaning includes the meaning of the original pattern, but a specific pattern's meaning is part of the original. For example, consider the pattern "*eat * tomorrow morning quickly.*" If unable to find an exactly matching pattern, it would be better to use "*eat * tomorrow * quickly*" rather than "*eat breakfast tomorrow morning quickly*" to represent it. Hence "*→token" wildcard replacements ("* →morning" in the example) should be assigned a lower cost than "token→*" wildcard replacements ("breakfast→*" in the example), as a wildcard token may represent several general tokens: "token→*" wildcard replacement (token →*) is equivalent to inserting more than zero tokens and "*→token" wildcard replacements are equivalent to deleting more than zero tokens. Therefore, we define the cost of "*→token" wildcard replacement as 1 and "token→*" wildcard replacement as e.

$$\begin{aligned} p_{match} &= \underset{p_{corpus} \in P_{corpus}}{\arg\max}\ similarity\left(p_{corpus} \rightarrow p_{wiriting}\right) \\ &= \underset{p_{corpus} \in P_{corpus}}{\arg\max}\ \text{-}\sqrt{GEC^2 + WRC^2} \end{aligned} \qquad (3)$$

The Euler equation, equation (3), takes into account both GEC and WRC to calculate the similarity of two patterns. The matching corpus pattern is that with the maximum similarity.

Candidate Emotion Word Ranking. The scoring function for ranking candidates $S = \{e_1, e_2, ..., e_I\}$ depends on the conditional probability of candidate e_i given writing patterns and candidates as defined in equation (4), which equals equation (5), assuming the patterns in $P_{writing}$ are mutually independent.

$$P\left(e_i \mid P_{writing}, S\right) = P\left(e_i \mid pw_1, pw_2, ..., pw_N, S\right) \qquad (4)$$

$$P\left(e_i \mid pw_1, pw_2, ..., pw_N, S\right) \propto \left(\prod_n P\left(e_i \mid pw_n, S\right)\right) \cdot P\left(e_i \mid S\right)^{1-N} \tag{5}$$

The second term in equation (5), $P(e_i|S)^{1-N}$, denotes the learner's preference with respect to writing topics. As we have no learner corpus, we assume that there are no such preferences and thus that $P(e_i|S)$ is uniformly distributed among e_i in S. As a result, when ranking e_i, $P(e_i|S)^{1-N}$ can be omitted. In addition, for the scores of the writing patterns we must use the scores of the matching corpus pattern found by the EDW algorithm for the corpus. Therefore, we rewrite the first term of equation (5) as follows.

$$\begin{aligned}&\prod_n P\left(e_i \mid pw_n, S\right)\\ &= \prod_n \frac{P\left(pw_n, e_i, S\right)}{P\left(p_n, e_i, S\right)} \cdot \frac{P\left(p_n, S\right)}{P\left(pw_n, S\right)} \cdot P\left(e_i \mid p_n, S\right)\\ &\propto \prod_n \frac{P\left(pw_n, e_i, S\right)}{P\left(p_n, e_i, S\right)} \cdot P\left(e_i \mid p_n, S\right)\end{aligned} \tag{6}$$

$P(e_i|p_n,S)$ in equation (6) can be calculated by *EES*, and the similarity value from equation (3) is utilized in equation (7) to estimate the first term. Equation (8), its logarithmic form, is the final scoring function for ranking.

$$\begin{aligned}&scr\left(e_i, P_{writing}, S\right)\\ &= \prod_n e^{similarity\left(p_n \to pw_n\right)} \cdot P\left(e_i \mid p_n, S\right)\end{aligned} \tag{7}$$

$$\begin{aligned}&\ln\left(scr\left(e_i, P_{writing}, S\right)\right)\\ &= \sum_n \left(\ln\left(\frac{EES_{p_n, e_i}}{\sum_{e_i \in S} EES_{p_n, e_i}}\right) - \sqrt{GEC_n{}^2 + WRC_n{}^2}\right)\end{aligned} \tag{8}$$

Modified EES. After observing the corpus characteristics, we further modified *EES* by adding the weighting factors ICZ_e (the Inverse CorpussiZe-ratio for emotion word e, where the corpus size denotes the number of patterns) and CTP^l_p (the emotion Category Transition Penalty for pattern p, where l denotes the level of the emotion word hierarchy, as explained later) in equation (9). *ICZ* in equation (10) normalizes the effect of the emotion word corpus size. When an emotion word appears more frequently, more example sentences are collected, resulting in a larger corpus. This can lead to a suggestion bias toward commonly seen emotion words.

$$EES'_{p,e} = EES_{p,e} \cdot ICZ_e \cdot \prod_{l=1}^{3}\left(1 - CTP^l_p\right) \tag{9}$$

$$ICZ_e = \log\left(P\left(e\right)^{-1}\right) \tag{10}$$

The other weighting factor, *CTP*, takes into account emotion word similarity. As mentioned, emotion words are derived from *WordNet-Affect* and then extended via *WordNetSynset* and *Webster Synonyms*; as shown in Figure 2, we build a three-layered hierarchy of emotion words. Level 1 is the six major emotion categories in *WordNet-Affect* (*anger, disgust, fear, joy, sadness,* and *surprise*), level 2 is the 1,000 emotion words from *WordNet-Affect*, and level 3 is the synonyms of the level-2 emotion words.

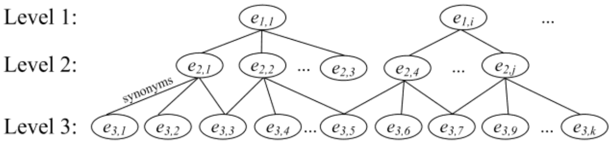

Figure 2: The emotion word hierarchy.

$$CTP^l_p = \frac{\dfrac{\left(P\left(p\right)\right)^2}{\sum_c \left(P\left(c, p\right)\right)^2} - 1}{m_{level}}, 0 \le CTP^l_p \le 1 \tag{11}$$

Intuitively, patterns that co-occur with many different emotion words are less informative. To assign less importance to these patterns, *CTP* estimates how often a pattern transits among emotion categories and adjusts its score accordingly in equation (11), where *m* is the number of categories in each level; *c* is the emotion category. High-*CTP* patterns appear in more emotion categories or are evenly distributed among emotion categories and are hence less representative. Note that categories in lower levels (for instance level 1) are less similar, and transitions among these make patterns less powerful.

4 Experiment

4.1 Emotion Words and Corpus

The corpora used in this study include *WordNet-Affect* (Strapparava and Valitutti, 2004), *WordNetSynset* (Fellbaum, 1999), *Merriam Webster Dictionary*, and *Vocabulary.com*. The WordNet-

Affect emotion list contains 1,113 emotion terms categorized into six major emotion categories: *anger*, *disgust*, *fear*, *joy*, *sadness*, and *surprise* (Ekman, 1993). 113 of the 1,113 terms were excluded because they were emotion phrases as opposed to words; thus a total of 1,000 emotion words were collected. Then, to increase coverage, synonyms of these 1,000 emotion words from *WordNetSynset* and *Merriam Webster Dictionary* were included. Thus we compiled a corpus with 3,785 emotion words. For each of these 3,785 emotion words there was an average of 13.1 suggested synonyms, with a maximum of 57 and a minimum of 1. Moreover, we extracted from *Vocabulary.com* a total of 2,786,528 example sentences, each containing emotion words. The maximum number of example sentences for a given emotion word was 1,255; the minimum was 3.

4.2 Testing Data and Gold Standard

A writing task was designed for the evaluation. To create the testing data, 240 emotion writings written by ESL learners were collected. The participants were non-native English speakers (native Chinese speakers), all undergraduates or higher. Each writing was a short story about one of the six emotions defined by Ekman, and each had three requirements: (1) a length of at least 120 words; (2) a consistent emotion throughout the story; and (3) a sentence at the end that contains an emotion word (hereafter referred to as the *target emotion word*) summarizing the feeling of the writing. The target emotion word and its synonyms were taken as candidates of the most appropriate word (hereafter termed *candidate emotion words*). From these, RESOLVE proposes for each writing the five most appropriate words.

To create the gold standard, two native-speaker judges ranked the appropriateness of the emotion word candidates for each target emotion word given the writing. The judges scored the candidates ranging from 0 (worst) to 6 (best) based on contextual clues. When two or more words were considered equally appropriate, equal ranks were allowed, i.e., skips were allowed in the ranking. For example, given the synonym list *angry, furious, enraged, mad* and *annoyed*, if the judge considered enraged and furious to be equally appropriate, followed by angry, mad and annoyed, then the ranking scores from highest to lowest would be *enraged*-6, *furious*-6, *angry*-4, *mad*-3 and *annoyed*-2, respectively. In addition, words not in the top five but that fit the context were assigned 1 whereas those that did not fit the context were assigned 0.

In order to gauge the quality of judges' ranks, Cohen's KAPPA value was utilized to measure the inter-judge agreement. KAPPA (k) is calculated by considering the ranking score to be either zero (0) or non-zero (1-6). In addition, a weighted KAPPA value for ranked evaluation (k_w) was adopted (Sim and Wright, 2005) to quantify the agreement between the native scores. On average, k=0.51, and k_w=0.68; both values indicate substantial inter-judge agreement.

5 Performance of RESOLVE

In this section, we first evaluate the performance of RESOLVE from several aspects: (1) the performance of EDW and modified EES, (2) a comparison of RESOLVE with commonly-adopted mutual information and machine learning algorithms for classification, and (3) a comparison of RESOLVE with tools for ESL learners. Then we utilize and compare the pattern sets P_{all} and $P_{-scPattern}$ (no self-containing patterns) introduced in Section 4.1. We adopt NDCG@5 as the evaluation metric, which evaluates the performance when viewing this work as a word suggestion problem.

5.1 EDW and Modified EES

We evaluate the effect of the pattern-matching algorithm EDW, EES modified by three layers of CTP weighting, and ICZ weighting. First we compare EDW matching with wildcard matching. For the baseline, we use conventional wildcard matching with neither ICZ nor CTP. The results in Table 2 show that EDW outperforms the baseline wildcard matching algorithm. In addition, using ICZ to account for the influence of the corpus size improves performance. Level-1 CTP performs best. Thus for the remaining experiments we use EDW and EES modified by ICZ weighting and level-1 CTP.

RESOLVE Components	NDCG@5
Baseline	0.5107
EDW	0.5138
EDW + level-1 CTP	0.5150
EDW + level-1 CTP + ICZ	**0.5529**
EDW + level-1, 2 CTP + ICZ	0.5104
EDW + level-1, 2, 3 CTP + ICZ	0.5098

Table 2: Performance with various components.

5.2 Comparison to MI/ML Methods

After demonstrating that the proposed EDW and modified EES for RESOLVE yield the best performance, we compare RESOLVE to representative methods in related work to demonstrate its superiority. As mentioned in Section 2, related works view similar research problems as emotion classification problems or word suggestion problems. Commonly-adopted approaches for the former are based on mutual information (MI) and the latter on machine learning (ML). To represent these two types of approaches, we selected PMI and SVM, respectively, to which we compare the performance of RESOLVE.

PMI, SVM and RESOLVE all used the same corpus. Note that NAVA words (noun, adjective, verb and adverb) are the major sentiment-bearing terms (Agrawal and An, 2012). Hence for comparison with the feature set of extracted patterns we selected NAVA words as the additional feature set. For the PMI approach we calculated PMI values (1) between NAVA words and emotion words, (2) between patterns and emotion words. The PMI values between features from the writing and one emotion word candidate are then summed as the ranking score of the candidate. For the SVM approach, we used libsvm (Chang *et al.*, 2011). We used a linear kernel to train for a classifier for each emotion by selecting all positive samples and an equal number of randomly-selected negative samples. We ran tests using various SVM parameter settings and found the performance differences to be within 1%. PMI, SVM and RESOLVE were all trained on the prepared three feature sets. SVM simply classifies each emotion word candidate as fitting the context or not. The confidence value of each answer is used for ranking.

From Table 3, we found the best features for the PMI and SVM approaches are NAVA words. NDCG@5 (BD) shows the binary decision performance when giving a score of 1 to all candidates with ranking scores from 1 to 6, and 0 otherwise. Note that it is possible that SVM when using NAVA words is too sparse to ensure satisfactory performance, as the number of corpus-extracted patterns exceeds one million; thus the result is not shown here, as this leads to excessive feature counts for SVM. Experimental results show that RESOLVE achieves the best performance; the significance test shows that RESOLVE (pattern) significantly outperforms PMI (NAVA) and SVM (NAVA) at tail *p*-values of less than 0.001.

Feature		PMI	SVM	RESOLVE
NAVA word	NDCG@5	**0.4275**	0.5122	0.5048
	NDCG@5(BD)	0.4778	**0.5229**	0.5236
Pattern	NDCG@5	0.4126	N/A	**0.5529**
	NDCG@5(BD)	0.4530	N/A	0.5627

Table 3: NDCG@5 for various feature sets.

As to RESOLVE, recall that there are two configurations for testing the effectiveness of self-containing patterns: RESOLVE including self-containing patterns (RESOLVE-P_{all}), and RESOLVE excluding self-containing patterns (RESOLVE-$P_{-scPattern}$). Six different emotion categories are analyzed individually to reveal their different characteristics (De Choudhury *et al.*, 2012). Table 4 shows the NDCG@5 averaged by the number of writings in six emotion categories for PMI (NAVA), SVM (NAVA), and RESOLVE-P_{all} and RESOLVE-$P_{-scPattern}$. A further analysis of the writings shows that when expressing *disgust* or *sadness*, extensive uses of emotion words are found. Therefore, RESOLVE-P_{all} yields better performance. The remaining four emotions are expressed through descriptions of events rather than using emotion words. These results conform to the conclusion from (De Choudhury, Counts and Gamon, 2012): negative moods tend to be described in limited context. Based on the finding in Table 4, RESOLVE-P_{all} is used for emotion writings about *disgust* and *sadness*, and RESOLVE-$P_{-scPattern}$ is used for writings about *anger, fear, joy* and *surprise* when building the final conditional RESOLVE system.

Emotion	PMI (NAVA)	SVM (NAVA)	RESOLVE $-P_{all}$	RESOLVE $-P_{-scPattern}$
Anger	0.3295	0.4706	0.4886	**0.5071**
Disgust	0.3103	0.3738	**0.3773**	0.2584
Fear	0.4064	0.5381	0.5168	**0.6152**
Joy	0.4849	**0.5764**	0.4456	0.5708
Sadness	0.2863	0.3495	**0.3999**	0.3194
Surprise	0.7346	0.7651	0.8037	**0.8400**

Table 4 NDCG@5 for six emotion categories.

5.3 Comparison to Tools for ESL Learners

In the final part of the system evaluation, we show the effectiveness of RESOLVE by evaluating the performance of the most commonly-used tools by ESL learners. One traditional and handy tool is the thesaurus. For this evaluation we selected Roget's Thesaurus[5]. Another tool is online language learning systems, of which PIGAI is the most well-known online rating system for writing for Chinese ESL learners. This system can also suggest to learners several easily-confused words as substitutes for several system-selected words. For evaluation, we posted the experimental writing to PIGAI to check whether there were any suggested substitutes for the target emotion word. Replacement suggestions were found for the target emotion word in 71 out of 240 writings. Therefore, we compared the performance of PIGAI and RESOLVE on these 71 writings. Note that what the thesaurus and PIGAI suggested are both appropriate word sets, where words are listed in alphabetic order. Learners must select by themselves (or most conveniently, simply select the first one). Table 5 shows that RESOLVE provides a better set of top-5 suggestions than both the thesaurus and PIGAI.

Tool	NDCG@5	NDCG@5 (BD)	Precision@5 (BD)
PIGAI (71/240)	0.3300	0.3095	0.8732
RESOLVE (71/240)	**0.4755**	**0.4728**	**0.9789**
Thesaurus	0.3708	0.4237	0.9146
RESOLVE	**0.5529**	**0.5627**	**0.9479**

Table 5: Performance using ESL learner tools.

6 Conclusion

We presented a probabilistic model that can suggest emotion word based on the context. The modified *EES* that considered the distribution of emotion word help our algorithm rank the candidate emotion words better. Besides, the matching algorithm, EDW, can find the most similar emotional event from the writings. Furthermore, the example sentences can be used as our training corpus without any handcraft annotations. The evaluation shows that the proposed approach can more appropriately suggest emotion words than other models and reference tools like PIGAI and Thesaurus.

Acknowledgements

Research of this paper was partially supported by National Science Council, Taiwan, under the contract MOST 101-2628-E-001-005-MY3.

References

Agrawal, A., and An, A. 2012. Unsupervised Emotion Detection from Text using Semantic and Syntactic Relations. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2012 IEEE/WIC/ACM International Conferences on* (Vol. 1, pp. 346-353). IEEE.

Altarriba, J., and Basnight-Brown, D. M. 2012. The acquisition of concrete, abstract, and emotion words in a second language. *International Journal of Bilingualism*, *16*(4), 446-452.

Bannard, C., and Callison-Burch, C. 2005. Paraphrasing with bilingual parallel corpora. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics* (pp. 597-604). Association for Computational Linguistics.

Barreiro, A. 2011. SPIDER: A System for Paraphrasing in Document Editing and Revision—Applicability in Machine Translation Pre-editing. In *Computational Linguistics and Intelligent Text Processing* (pp. 365-376). Springer Berlin Heidelberg.

Barzilay, R., and Lee, L. 2003. Learning to paraphrase: an unsupervised approach using multiple-sequence alignment. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1* (pp. 16-23). Association for Computational Linguistics.

Barzilay, R., and McKeown, K. R. 2001. Extracting paraphrases from a parallel corpus. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics* (pp. 50-57). Association for Computational Linguistics.

[5] http://Thesaurus.com

Bullinaria, J. A., and Levy, J. P. 2007. Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior research methods*, *39*(3), 510-526.

Callison-Burch, C. 2008. Syntactic constraints on paraphrases extracted from parallel corpora. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (pp. 196-205). Association for Computational Linguistics.

Chang, C. C., and Lin, C. J. 2011. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *2* (3), 27.

Chen, M. H., Huang, S. T., Chang, J. S., and Liou, H. C. 2013. Developing a corpus-based paraphrase tool to improve EFL learners' writing skills. *Computer Assisted Language Learning*, (ahead-of-print), 1-19.

Chen, M. H., Huang, S. T., Huang, C. C., Liou, H. C., and Chang, J. S. 2012. PREFER: using a graph-based approach to generate paraphrases for language learning. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP* (pp. 80-85). Association for Computational Linguistics.

Church, K. W., and Hanks, P. 1990. Word association norms, mutual information, and lexicography. *Computational linguistics*, *16*(1),

Dahl, G., Frassica, A. M., and Wicentowski, R. (2007, June). SW-AG: Local context matching for English lexical substitution. In *Proceedings of the 4th International Workshop on Semantic Evaluations* (pp. 304-307). Association for Computational Linguistics. 22-29.

De Choudhury, M., Counts, S., and Gamon, M. 2012. Not all moods are created equal! exploring human emotional states in social media. In *Sixth International AAAI Conference on Weblogs and Social Media*.

De Marneffe, M. C., MacCartney, B., and Manning, C. D. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC* (Vol. 6, pp. 449-454).

Dinu, G., and Lapata, M. 2010. Topic models for meaning similarity in context. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters* (pp. 250-258). Association for Computational Linguistics.

Ekman, P. 1993. Facial expression and emotion. *American Psychologist*, *48* (4), 384.

Fellbaum, C. 1999. *WordNet*. Blackwell Publishing Ltd.

Ghazi, D., Inkpen, D., and Szpakowicz, S. 2012. Prior versus Contextual Emotion of a Word in a Sentence. In *Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis* (pp. 70-78). Association for Computational Linguistic

Jeong, Y., and Myaeng, S. H. 2012. Using Syntactic Dependencies and WordNet Classes for Noun Event Recognition. In *The 2nd Workhop on Detection, Representation, and Exploitation of Events in the Semantic Web in Conjunction with the 11th International Semantic Web Conference* (pp. 41-50).

Keshtkar, F., and Inkpen, D. 2010. A corpus-based method for extracting paraphrases of emotion terms. In *Proceedings of the NAACL HLT 2010 Workshop on Computational approaches to Analysis and Generation of emotion in Text* (pp. 35-44). Association for Computational Linguistics.

Klein, D., and Manning, C. D. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1* (pp. 423-430). Association for Computational Linguistics.

Leading, L. L., Monaghan, W., and Bridgeman, B. 2005. E-rater as a Quality Control on Human Scores.

Lin, D. 1998. An information-theoretic definition of similarity. In *ICML* (Vol. 98, pp. 296-304).

Lin, D., and Pantel, P. 2001. DIRT—discovery of inference rules from text. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 323-328). ACM.

Ma, K. (2013). Improving EFL Graduate Students' Proficiency in Writing through an Online Automated Essay Assessing System. *English Language Teaching*, *6*(7).

Pavlenko, A. 2008. Emotion and emotion-laden words in the bilingual lexicon. *BILINGUALISM LANGUAGE AND COGNITION*, *11*(2), 147.

Sim, J., and Wright, C. C. 2005. The kappa statistic in reliability studies: use, interpretation, and sample size requirements. *Physical therapy*, *85*(3), 257-268.

Strapparava, C., and Valitutti, A. 2004. WordNet Affect: an Affective Extension of WordNet. In *LREC* (Vol. 4, pp. 1083-1086).

Tokuhisa, R., Inui, K., and Matsumoto, Y. 2008. Emotion classification using massive examples extracted from the web. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1* (pp. 881-888). Association for Computational Linguistics.

Zhao, S., Zhao, L., Zhang, Y., Liu, T., and Li, S. (2007, June). Hit: Web based scoring method for english lexical substitution. In *Proceedings of the 4th International Workshop on Semantic Evaluations* (pp. 173-176). Association for Computational Linguistics.