EmotiWord: Affective Lexicon Creation with Application to Interaction and Multimedia Data

Nikos Malandrakis[1], Alexandros Potamianos[1], Elias Iosif[1], and Shrikanth Narayanan[2]

[1] Dept. of ECE, Technical Univ. of Crete, 73100 Chania, Greece
{nmalandrakis,potam,iosife}@telecom.tuc.gr
[2] SAIL Lab, Dept. of EE, Univ. of Southern California, Los Angeles, CA 90089, USA
shri@sipi.usc.edu

Abstract. We present a fully automated algorithm for expanding an affective lexicon with new entries. Continuous valence ratings are estimated for unseen words using the underlying assumption that semantic similarity implies affective similarity. Starting from a set of manually annotated words, a linear affective model is trained using the least mean squares algorithm followed by feature selection. The proposed algorithm performs very well on reproducing the valence ratings of the Affective Norms for English Words (ANEW) and General Inquirer datasets. We then propose three simple linear and non-linear fusion schemes for investigating how lexical valence scores can be combined to produce sentence-level scores. These methods are tested on a sentence rating task of the SemEval 2007 corpus, on the ChIMP politeness and frustration detection dialogue task and on a movie subtitle polarity detection task.

Keywords: language understanding, emotion, affect, affective lexicon.

1 Introduction

Affective text analysis, the analysis of the emotional content of lexical information is an open research problem that is very relevant for numerous natural language processing, web and multimodal dialogue applications. Emotion recognition from multimedia streams (audio, video, text) and emotion recognition of users of interactive applications is another area where the affective analysis of text plays an important, yet still limited role [10,9,2].

The requirements of different applications lead to the definition of affective sub-tasks, such as emotional category labeling (assigning label(s) to text fragments, e.g., "sad"), polarity recognition (classifying into positive or negative) and subjectivity identification (separating subjective from objective statements). The wide-range of application scenarios and affective tasks has lead to the fragmentation of research effort. The first step towards a general task-independent solution to affective text analysis is the creation of an appropriate affective lexicon, i.e., a resource mapping each word (or term) to a set of affective ratings.

E. Salerno, A.E. Çetin, and O. Salvetti (Eds.): MUSCLE 2011, LNCS 7252, pp. 30–41, 2012.

A number of affective lexicons for English have been manually created, such as the General Inquirer [15], and Affective norms for English Words (ANEW) [3]. These lexica, however, fail to provide the required vocabulary coverage; the negative and positive classes of the General Inquirer contain just 3600 words, while ANEW provides ratings for just 1034 words. Therefore, computational methods are necessary to create or expand an already existing lexicon. Well-known lexica resulting from such methods are SentiWordNet [5] and WORDNET AFFECT [17]. However, such efforts still suffer from limited coverage.

A variety of methods have been proposed for expanding known lexica with continuous affective scores, or, simply for assigning binary "positive - negative" labels to new words, also known as semantic orientation [7]. The underlying assumption at the core of these methods is that *semantic similarity can be translated to affective similarity.* Therefore given some metric of the similarity between two words one may derive the similarity between their affective ratings. A recent survey of metrics of semantic similarity is presented in [8]. The semantic similarity approach, pioneered in [20], uses a set of words with known affective ratings, usually referred as *seed words*, as a starting point. Then the semantic similarity between each new word in the lexicon and the seed words is computed and used to estimate the affective rating of new words. Various methods have been proposed to select the initial set of words: seed words may be the lexical labels of affective categories ("anger","happiness"), small sets of words with unambiguous (affective) meaning or, even, all words in an affective lexicon. Having a set of seed words and the appropriate similarity measure, the next step is devising a method of combining them to estimate the affective score or category.

Once the affective lexicon has been expanded to include all words in our vocabulary, the next step is the combination of word ratings to estimate affective scores for larger lexical units, phrases or sentences. Initially the subset of affect-bearing words has to be selected, depending on their part-of-speech tags [4], affective rating and/or sentence structure [1]. Then word-level ratings are combined, typically in a simple fashion, such as the arithmetic mean. More complex approaches take into account sentence structure, word/phrase level interactions such as valence shifters [14] and large sets of manually created rules [4,1].

In this paper, we aim to create an affective lexicon with fine-grained/pseudo-continuous valence ratings. This lexicon can be readily expanded to cover unseen words with no need to consult any linguistic resources. The work builds on [20]. The proposed method requires only a small number (a few hundred) labeled seed words and a web search engine to estimate the similarity between the seed and unseen words. Further, to improve the quality of the affective lexicon we propose a machine learning approach for training a valence estimator. The affective lexicon created is evaluated against manually labeled corpora both at the word and the sentence level, achieving state-of-the-art results despite the lack of underlying syntactic or pragmatic information in our model. We also investigate the use of small amounts of in-domain data for adapting the affective models. Results show that domain-independent models perform very well for certain tasks, e.g., for frustration detection in spoken dialogue systems.

2 Affective Rating Computation

Similarly to [20], we start from an existing, manually annotated lexicon. Then we automatically select a subset of seed words, using established feature selection algorithms. The rating (in our case valence) for an unseen word is estimated as the linear combination of the ratings of seed words weighted by the semantic similarity between the unseen and seed words. Semantic similarity weights are computed using web hit-based metrics. In addition, a linear weight is used that regulates the contribution of each seed word in the valence computation formula. The weight of each seed word is optimized to minimize the mean square estimation error over all words in the training set.

Introducing a trainable weight for each seed word is motivated by the fact that semantic similarity does not fully capture the relevance of a seed word for valence computation. For instance, consider an unseen word and a lexicon that consists of two seed words that are equally (semantically) similar to the unseen word. Based on the assumption that semantic similarity implies affective similarity both seed words should be assigned the same feature weight. However, there is a wide range of factors affecting the relevance of each seed word, e.g., words that have high affective variance (many affective senses) might prove to be worse features that affectively unambiguous words. Other factors might include the mean valence of seed words and the degree of centrality (whether they are indicative samples of their affective area). Instead of evaluating the effect of each factor separately, we choose to use machine learning to estimate a single weight per seed word using Least Mean Squares estimation (LMS).

2.1 Word Level Tagging - Metrics of Semantic Similarity

We aim at characterizing the affective content of words in a continuous valence range of $[-1, 1]$ (from very negative to very positive), *from the reader perspective*. We hypothesize that the valence of a word can be estimated as a linear combination of its semantic similarities to a set of seed words and the valence ratings of these words, as follows:

$$\hat{v}(w_j) = a_0 + \sum_{n=1}^{N} a_i \; v(w_i) \; d(w_i, w_j), \tag{1}$$

where w_j is the word we mean to characterize, $w_1 ... w_N$ are the seed words, $v(w_i)$ is the valence rating for seed word w_i, a_i is the weight corresponding to word w_i (that is estimated as described next), and $d(w_i, w_j)$ is a measure of semantic similarity between words w_i and w_j.

Assume a training corpus of K words with known ratings and a set of $N < K$ seed words for which we need to estimate weights a_i, we can use (1) to create a system of K linear equations with $N + 1$ unknown variables as shown next:

$$\begin{bmatrix} 1 & d(w_1,w_1)v(w_1) & \cdots & d(w_1,w_N)v(w_N) \\ \vdots & \vdots & \vdots & \vdots \\ 1 & d(w_K,w_1)v(w_1) & \cdots & d(w_K,w_N)v(w_N) \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_N \end{bmatrix} = \begin{bmatrix} 1 \\ v(w_1) \\ \vdots \\ v(w_K) \end{bmatrix} \quad (2)$$

where $a_1 ... a_N$ are the N seed word weights and a_0 is an additional parameter that corrects for affective bias in the seed word set. The optimal values of these parameters can be estimated using LMS. Once the weights of the seed words are estimated the valence of an unseen word w_j can be computed using (1).

To select N seeds out of K labeled words, we need to perform feature selection or simply randomly select a subset. Here, we rank all K words by their "worth" as features, using a wrapper feature selector working on a "best-first" forward selection strategy. The performance metric used for feature evaluation is Mean Square Error. Thus, we simply pick the N seeds that are best at predicting the affective rating of the rest of the training data. The seeds typically correspond to 10% to 50% of the training set (for more details see also Sections 4 and 5).

The valence estimator defined in (1) employs a metric $d(w_i, w_j)$ that computes the semantic similarity between words w_i and w_j. In this work, we use hit-based similarity metrics that estimate the similarity between two words/terms using the frequency of co-occurence within larger lexical units (sentences, documents). The underlying assumption is that terms that often co-occur in documents are likely to be related. A popular method to estimate co-occurrence is to pose conjunctive queries including both terms to a web search engine; the number of returned hits is an estimate of the frequency of co-occurrence [8]. Hit-based metrics do not depend on any language resources, e.g., ontologies, and do not require downloading documents or snippets, as is the case for context-based semantic similarities. In this work, we employ four well-established hit-based metrics namely, Dice coefficient, Jaccard coefficient, point-wise mutual information (PMI), and Google-based Semantic Relatedness (Google). Due to space limitations details are omitted, but the reader may consult [8].

2.2 Sentence Level Tagging

The principle of compositionality [13] states that the meaning of a phrase or sentence is the sum of the meaning of its parts. The generalization of the principle of compositionality to affect could be interpreted as follows: to compute the valence of a sentence simply take the average valence of the words in that sentence. The affective content of a sentence $s = w_1 w_2 ... w_N$ in the simple linear model is:

$$v_1(s) = \frac{1}{N} \sum_{i=1}^{N} v(w_i). \quad (3)$$

This simple linear fusion may prove to be inadequate for affective interpretation given that non-linear affective interaction between words (especially adjacent words) in the same sentence is common. It also weights equally words that have

a strong and weak affective content and tends to give lower absolute valence scores to sentences that contain many neutral (non-content) words. Thus we also consider a normalized weighted average, in which words that have high absolute valence values are weighted more, as follows:

$$v_2(s) = \frac{1}{\sum_{i=1}^{N} |v(w_i)|} \sum_{i=1}^{N} v(w_i)^2 \cdot \text{sign}(v(w_i)), \tag{4}$$

where sign(.) is the signum function. Alternatively we consider non-linear fusion, in which the word with the highest absolute valence value dominates the meaning of the sentence, as follows. Equations (3), (4) and (5) are the fusion schemes used in the following experiments, where they are referred to as: average (avg), weighted average (w.avg) and maximum (max).

$$v_3(s) = \max_i(|v(w_i)|) \cdot \text{sign}(v(w_z)), \quad \text{where} \quad z = \arg\max_i(|v(w_i)|) \tag{5}$$

3 Corpora and Experimental Procedure

Experimental Corpora: (1) ANEW: The main corpus used for creating the affective lexicon is the ANEW dataset. It consists of 1034 words, rated in 3 continuous dimensions of arousal, valence and dominance. In this work, we only use the valence ratings provided. ANEW was used for both training (seed words) and evaluation using cross-validation (as outlined below). (2) General Inquirer: The second corpus used for evaluation of the affective lexicon creation algorithm is the General Inquirer corpus that contains 2005 negative and 1636 positive words. It was created by merging words with multiple entries in the original lists of 2293 negative and 1914 positive words. It is comparable to the dataset used in [20,21]. (3) SemEval: For sentence level tagging evaluation (no training is performed here, only testing) the SemEval 2007: Task 14 corpus is used [16]. It contains 1000 news headlines manually rated in a fine-grained valence scale $[-100, 100]$, which is rescaled to a $[-1, 1]$ for our experiments. (4) Subtitles: For the movie subtitle evaluation task, we use the subtitles of the corpus presented in [11]. It contains the subtitles of twelve thirty minute movie clips, a total of 5388 sentences. Start and end times of each utterance were extracted from the subtitles and each sentence was given a continuous valence rating equal to the average of the multimodal affective curve for the duration of the utterance. (5) ChIMP: The ChIMP database was used to evaluate the method on spontaneous spoken dialog interaction. It contains 15585 manually annotated spoken utterances, with each utterance labeled with one of three emotional state tags: neutral, polite, and frustrated [22].

Semantic Similarity Computation: In our experiments we utilized four different similarity metrics based on web co-occurrence, namely, Dice coefficient,

Jaccard coefficient, point-wise mutual information (PMI) and Google-based Semantic Relatedness (Google). To compute the co-occurrence hit-count we use conjunctive "AND" web queries [8] so we are looking for co-occurrence anywhere inside a web document. All queries were performed using the Yahoo! search engine. The number of seed words determines the number of queries that will be required. Assuming that N seed words are selected, $N + 1$ queries will be required to rate each new word.

3.1 Affective Lexicon Creation

The ANEW dataset was used for both training and testing using 5-fold cross-validation. On each fold, 80% of the ANEW words was used for training and 20% for evaluation. For each fold, the words in the training set were ranked based on their value as features using a wrapper (for feature selection) on a 3-fold cross-validation prediction experiment conducted within the training set. On each fold 66% of the training data were used to predict the ratings of the remaining 33%, using our method. The selection started with no seed words, then more seed words were added iteratively - in each iteration all non-seeded words in the training set were tested as seeds, then the one that produced the lowest mean square error was added to the previous seed word set. The order in which words were added to the selection seed word set was their ranking. Then the N first were used as seed words. We provide results for a wide range of N values, from 1 to 500 features. Then the semantic similarity between each of the N features and each of the words in the test set ("unseen" words) was computed, as discussed in the previous section. Next for each value of N, the optimal weights of the linear equation system matrix in (2) were estimated using LMS. For each word in the test set the valence ratings were computed using (1).

A toy training example using $N = 10$ features and the Google Semantic Relatedness hit-based metric is shown in Table 3.1. The seed words are presented in the order they were selected by the wrapper feature selection method. The last row in the table corresponds to the bias term a_0 in (1) that takes a small positive value. The third column $v(w_i)$ shows the manually annotated valence of word w_i, while the fourth column a_i shows the corresponding linear weight computed by the LMS algorithm. Their product (final column) $v(w_i) \times a_i$ is a measure of the affective "shift" of the valence of each word per "unit of similarity" to that seed word (see also (1)).

Table 1. Training sample using 10 seed words

Order	w_i	$v(w_i)$	a_i	$v(w_i) \times a_i$
1	mutilate	-0.8	0.75	-0.60
2	intimate	0.65	3.74	2.43
3	poison	-0.76	5.15	-3.91
4	bankrupt	-0.75	5.94	-4.46
5	passion	0.76	4.77	3.63
6	misery	-0.77	8.05	-6.20
7	joyful	0.81	6.4	5.18
8	optimism	0.49	7.14	3.50
9	loneliness	-0.85	3.08	-2.62
10	orgasm	0.83	2.16	1.79
-	w_0 *(offset)*	1	0.28	0.28

In addition to the ANEW corpus, the General Inquirer corpus was used (only) for testing. The features and corresponding linear weights were trained on the (whole of the) ANEW corpus and used as seeds to estimate continuous valence ratings for the General Inquirer corpus. Our goal here was not only to evaluate the proposed algorithm, but also investigate whether using seeds from one annotated corpus can robustly estimate valence ratings in another corpus.

The following objective evaluation metrics were used to measure the performance of the affective lexicon expansion algorithm: (i) Pearson correlation between the manually labeled and automatically computed valence ratings, (ii) mean square error (assuming that the manually labeled scores were the ground truth) and (iii) binary classification accuracy of positive vs negative relations, i.e., continuous ratings are produced, converted to binary decisions and compared to the ground truth.

3.2 Sentence Level Tagging

The SemEval 2007: Task 14 corpus was used for evaluating the various fusion methods for turning word into sentence ratings. All unseen words in the SemEval corpus are added to the lexicon using the affective lexicon expansion algorithm outlined above. The model used to create the required ratings is trained using all of the words in the ANEW corpus as training samples and 300 of them as seed words, i.e., $N = 300$ for this experiment. Then the ratings of the words are combined to create the sentence rating using linear fusion (3), weighted average fusion (4) or non-linear max fusion (5). In the first experiment (labeled "all words"), all words in a sentence are taken into account to compute the sentence score. In the second experiment (labeled "content words"), only the verbs, nouns, adjectives and adverbs are used.

Similarly for the subtitles corpus, all required word ratings were created by a model trained with ANEW data, then combined into sentence ratings. The temporal dependencies of the subtitles are disregarded (sentences occurring close to each other are likely to have similar content), each sentence is handled out of context. For this experiment, all words in the subtitles are used.

In order to adapt the affective model to the ChIMP task, the discrete sentence level valence scores were mapped as follows: frustrated was assigned a valence value of -1, neutral was 0 and polite was 1. To bootstrap the valence scores for each word in the ChIMP corpus, we used the average sentence-level scores for all sentence where that word appeared. Finally, the ANEW equation system matrix was augmented with all the words in the ChIMP corpus and the valence model in (2) was estimated using LMS. Note that for this training process a 10-fold cross validation experiment was run on the ChIMP corpus sentences. The relative weight of the ChIMP corpus adaptation data was varied by adding the respective lines multiple times to the augmented system matrix, e.g., adding each line twice gives a weight of $w = 2$. We tested weights of $w = 1$, $w = 2$, and using only the samples from ChIMP as training samples (denoted as $w = \infty$). The valence boundary between frustrated and other classes was selected based on the a-priori probability distribution for each class, and is simply the Bayesian

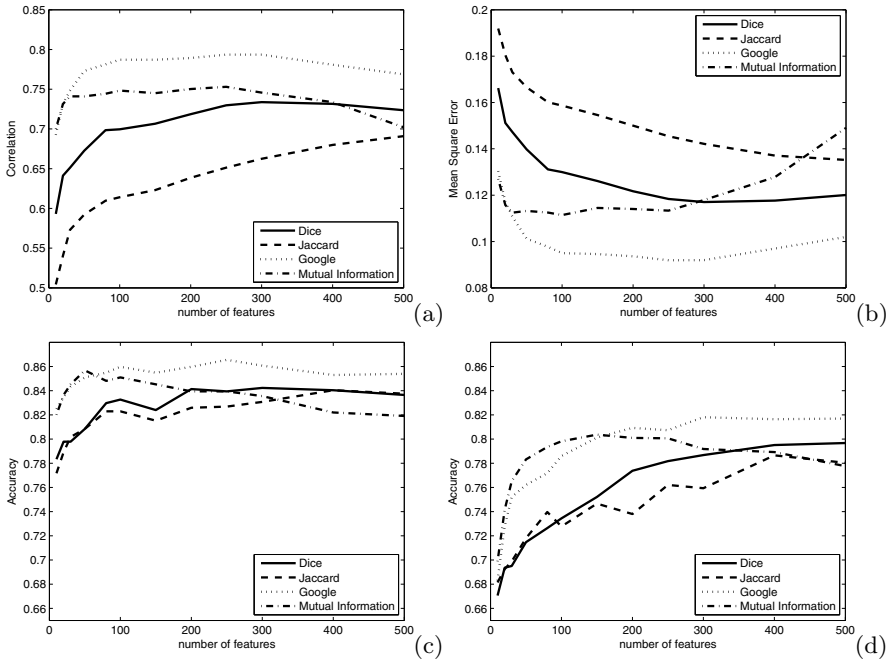

Fig. 1. Performance as a function of number of seed words for the four similarity metrics: (a) correlation between the automatically computed and manually annotated scores (ANEW corpus), (b) mean square error (ANEW), (c) binary (positive vs negative) classification accuracy (ANEW), (d) binary classification accuracy (General Inquirer)

decision boundary (similarly between polite and other classes). The focus of this experiment is the effectiveness of the adaptation procedure, therefore only results using Google Semantic Relatedness and all words of each sentence are presented.

In order to evaluate the performance of the sentence level affective scores we use the following metrics: (i) Pearson correlation between the manually labeled and automatically computed scores, (ii) classification accuracy for the 2-class (positive, negative) and 3-class (positive, neutral, negative) problems, and (iii) precision, recall and F-score for the 2-class (positive vs negative) problem.

4 Results for Affective Lexicon Creation

Figure 1 shows the performance of the affective lexicon creation algorithm on the ANEW (a)-(c) and General Inquirer corpora (d) as a function of the number of features N (seed words) and for each semantic similarity metric. Overall, the results obtained are very good both in terms of correlation and binary classification accuracy. Performance improves up to about $N = 200$ to 300 seed words and then levels off or falls slightly. Note, that good performance is achieved even with few features (less than 100 features) especially when using the mutual information or Google semantic relatedness metrics. Although, more features should

Table 2. Correlation, class. accuracy for SemEval dataset wrt metrics and fusion.

Correlation						
Similarity Metric	All Words: fusion scheme			Content Words: fusion scheme		
	avg	w.avg	max	avg	w.avg	max
Dice	0.48	0.46	0.44	0.5	0.47	0.45
Jaccard	0.37	0.3	0.28	0.45	0.39	0.36
PMI	0.37	0.31	0.25	0.4	0.34	0.27
Google	0.44	0.44	0.42	0.47	0.46	0.44
3-Class (pos., neutral, neg.) / 2-Class (pos., neg.) Classification Accuracy						
Similarity Metric	All Words: fusion scheme			Content Words: fusion scheme		
	avg	w.avg	max	avg	w.avg	max
Dice	0.60/0.65	0.59/0.68	0.53/0.70	0.60/0.68	0.60/0.69	0.54/0.71
Jaccard	0.59/0.59	0.42/0.58	0.31/0.58	0.60/0.67	0.57/0.68	0.47/0.67
PMI	0.60/0.60	0.54/0.61	0.34/0.61	0.59/0.64	0.54/0.64	0.37/0.63
Google	0.60/0.67	0.59/0.68	0.51/0.69	0.60/0.69	0.59/0.69	0.51/0.69

in principle help, only $K = 827$ training words exist in each fold of the ANEW corpus making it hard to robustly estimate the feature weights (over-fitting). A larger seed vocabulary would enable us to use even more features effectively and possibly lead to improved performance . The best performing semantic similarity metric is Google Semantic Relatedness, followed by the mutual information and Dice metrics. This trend is consistent in all experiments shown in Fig. 1. Note that PMI is not very well suited to this task, since it is the only one among the metrics that is unbounded. This is a possible explanation for the different performance dynamics it exhibits, peaking much earlier (around $N = 50$ to 150) and then trailing off. However, PMI it still one of the top performers. In terms of absolute numbers, the results reported here are at least as good as the state-of-the-art. Correlation results for the ANEW corpus shown in Fig. 1(a) are between 0.72 and 0.78. Binary classification results for the ANEW corpus shown in Fig. 1(c) are between 82% and 86% for the various metrics. Binary classification accuracy for the General Inquirer corpus shown in Fig. 1(d) is up to 82% using Google Semantic Relatedness. Compare this to 82.8% accuracy quoted in [20] and [21] using NEAR queries for computing semantic similarity, 82.1% in [6], and 81.9% in [19]. Note that the latter two methods achieve higher performance when using part of the GI corpus itself as training. Overall, the results are impressive given that ANEW ratings were used to seed the GI corpus and the differences in the manual tagging procedure for each corpus.

5 Results for Sentence Level Tagging

SemEval Dataset: The performance for the SemEval sentence level affective tagging is summarized in Table 2. Results are shown for 3-way and 2-way classification experiments.

For 3-class experiments the following mapping from continuous to discrete values was performed: $[-1, -0.5) \rightarrow -1$ (negative), $[-0.5, 0.5) \rightarrow 0$ (neutral),

and $[0.5, 1] \rightarrow 1$ (positive). Similarly for 2-class: $[-1, 0) \rightarrow -1$ (negative) and $[0, 1] \rightarrow 1$ (positive). The correlation and 3-way accuracy numbers are higher than those reported in [16] (47.7% and 55.1% respectively) and 2-way accuracy is comparable to that achieved by [12] (71%). The relative performance of the various similarity metrics is different than for the word rating task. Google semantic relatedness performed best at the word level, but is the second-best performer at the sentence level. Conversely, Dice that was an average performer on words, works the best on sentences. When comparing the three fusion schemes, linear fusion performs well in terms of correlation, 2-way and 3-way classification. Non-linear fusion schemes (weighted average and, especially, max) work best for the binary classification task, giving more weight to words with extreme valence scores. Finally, using only content words (instead of all words) improves results, yet the gain is minimal in almost all cases. This indicates that non-content words already have been assigned low word rating scores. Overall, the results are good, significantly higher than those reported in [18] and on par with the best systems reported in [16] and [12], when evaluating performance on *all* the sentences in the dataset. Best results in terms of correlation of 0.5 are achieved using linear fusion. 3-way classification accuracy (although higher than any system reported in [16]) is still poor, since the a-priori probability for the neutral class is 0.6 and our best performance, achieved using linear (or weighted average) fusion, barely matched that at 60%. Finally, non-linear max fusion achieves the best results for 2-way classification at 71%.

Subtitles Dataset: Table 3 shows performance in the subtitles dataset. The results are fairly low, with accuracy under 60%, and very small correlation. The relatively poor results show the added complexity of this task compared to unimodal polarity detection in text. More likely it points to the significance of factors we ignored: interactions across sentences and across modalities. It is reasonable to assume that context acts as a modifier on the affective interpretation of each utterance. Furthermore, here, it is not just lexical context that contributes to the ground truth, but also multimedia context: from the voice tone of the actor, to his facial expression, to the movie's setting.

ChIMP Corpus: In Table 4, the two-class sentence-level classification accuracy is shown for the ChIMP corpus (polite vs other: "P vs O", frustrated vs other: "F vs O"). For the adaptation experiments on the ChIMP corpus, the parameter w denotes the weighting given to the in-domain ChIMP data, i.e., number of

Table 3. Performance on subtitles dataset for Google metric wrt fusion

Performance Measurement	Fusion scheme		
	avg	w.avg	max
Correlation	0.05	0.05	0.06
2-Class Classification Accuracy (positive, negative)	0.58	0.56	0.56
Precision (positive vs negative)	0.62	0.62	0.61
Recall (positive vs negative	0.85	0.84	0.82
F-score (positive vs negative)	0.72	0.71	0.70

Table 4. Sentence classification accuracy for ChIMP baseline and adapted tasks

Sentence Classification Accuracy	Fusion scheme		
	avg	w.avg	max
Baseline: P vs O / F vs O	0.70/0.53	0.69/0.62	0.54/0.66
Adapt $w = 1$: P vs O / F vs O	0.74/0.51	0.70/0.58	0.67/0.57
Adapt $w = 2$: P vs O / F vs O	0.77/0.49	0.74/0.53	0.71/0.53
Adapt $w = \infty$: P vs O / F vs O	0.84/0.52	0.82/0.52	0.75/0.52

times the adaptation equations were repeated in the system matrix (2). Results are shown for the three fusion methods (average, weighted average, maximum). For the ChIMP politeness detection task, performance of the baseline (unsupervised) model is lower than that quoted in [22] for lexical features. Performance improves significantly by adapting the affective model using in-domain ChIMP data reaching up to 84% accuracy for linear fusion (matching the results in [22]). The best results for frustration detection are achieved with the baseline model and max fusion schemes at 66% (at least as good as those reported in [22]).

6 Conclusions

We proposed an affective lexicon creation/expansion algorithm that estimates a continuous valence score for unseen words from a set of manually labeled seed words and semantic similarity ratings. The lexicon creation algorithm achieved very good results on the ANEW and General Inquirer datasets using 200-300 seed words, achieving correlation scores of up to 0.79 and over 80% binary classification accuracy. In addition, preliminary results on sentence level valence estimation show that simple fusion schemes achieve performance that is at least at a par with the state-of-the-art for the SemEval task. For politeness detection it was shown that adaptation of the affective model and linear fusion achieves the best results. For frustration detection, the domain-independent model and max fusion gave the best performance. Polarity detection on the movie subtitles dataset proved the most challenging, showing the importance of context and multimodal information for estimating affect. Overall, we have shown that an unsupervised domain-independent approach is a viable alternative to training domain-specific language models for the problem of affective text analysis.

Although the results are encouraging, this work represents only the first step towards applying machine learning methods to affective text analysis. Future research direction could involve more complex models (e.g., non-linear, kernel) to model lexical affect, alternative semantic similarity or relatedness metrics, and better fusion models that incorporate syntactic and pragmatic information to improve sentence-level valence scores. Overall, the proposed method creates reasonably accurate ratings and is a good starting point for future research. An important advantage of the proposed method is it's simplicity: the only requirements for constructing EmotiWord are a few hundred labeled seed words and a web search engine.

References

1. Andreevskaia, A., Bergler, S.: CLaC and CLaC-NB: Knowledge-based and corpus-based approaches to sentiment tagging. In: Proc. SemEval, pp. 117–120 (2007)
2. Ang, J., Dhillon, R., Krupski, A., Shriberg, E., Stolcke, A.: Prosody-based automatic detection of annoyance and frustration in human-computer dialog. In: Proc. ICSLP, pp. 2037–2040 (2002)
3. Bradley, M., Lang, P.: Affective norms for english words (ANEW): Stimuli, instruction manual and affective ratings. Technical report C-1. The Center for Research in Psychophysiology, University of Florida (1999)
4. Chaumartin, F.R.: UPAR7: A knowledge-based system for headline sentiment tagging. In: Proc. SemEval, pp. 422–425 (2007)
5. Esuli, A., Sebastiani, F.: Sentiwordnet: A publicly available lexical resource for opinion mining. In: Proc. LREC, pp. 417–422 (2006)
6. Hassan, A., Radev, D.: Identifying text polarity using random walks. In: Proc. ACL, pp. 395–403 (2010)
7. Hatzivassiloglou, V., McKeown, K.: Predicting the Semantic Orientation of Adjectives. In: Proc. ACL, pp. 174–181 (1997)
8. Iosif, E., Potamianos, A.: Unsupervised Semantic Similarity Computation Between Terms Using Web Documents. IEEE Transactions on Knowledge and Data Engineering 22(11), 1637–1647 (2009)
9. Lee, C.M., Narayanan, S.S., Pieraccini, R.: Combining acoustic and language information for emotion. In: Proc. ICSLP, pp. 873–876 (2002)
10. Lee, C.M., Narayanan, S.S.: Toward detecting emotions in spoken dialogs. IEEE Transactions on Speech and Audio Processing 13(2), 293–303 (2005)
11. Malandrakis, N., Potamianos, A., Evangelopoulos, G., Zlatintsi, A.: A supervised approach to movie emotion tracking. In: Proc. ICASSP, pp. 2376–2379 (2011)
12. Moilanen, K., Pulman, S., Zhang, Y.: Packed feelings and ordered sentiments: Sentiment parsing with quasi-compositional polarity sequencing and compression. In: Proc. WASSA Workshop at ECAI, pp. 36–43 (2010)
13. Pelletier, F.J.: The principle of semantic compositionality. Topoi 13, 11–24 (1994)
14. Polanyi, L., Zaenen, A.: Contextual valence shifters. In: Computing attitude and affect in text: Theory and Applications, pp. 1–10. Springer (2006)
15. Stone, P., Dunphy, D., Smith, M., Ogilvie, D.: The General Inquirer: A Computer Approach to Content Analysis. The MIT Press (1966)
16. Strapparava, C., Mihalcea, R.: Semeval-2007 task 14: Affective text. In: Proc. SemEval, pp. 70–74 (2007)
17. Strapparava, C., Valitutti, A.: WordNet-Affect: an affective extension of WordNet. In: Proc. LREC, vol. 4, pp. 1083–1086 (2004)
18. Taboada, M., Brooke, J., Tofiloski, M., Voll, K., Stede, M.: Lexicon-based methods for sentiment analysis. Computational Linguistics 1, 1–41 (2010)
19. Takamura, H., Inui, T., Okumura, M.: Extracting semantic orientations of words using spin model. In: Proc ACL, pp. 133–140 (2005)
20. Turney, P., Littman, M.L.: Unsupervised Learning of Semantic Orientation from a Hundred-Billion-Word Corpus. Technical report ERC-1094 (NRC 44929). National Research Council of Canada (2002)
21. Turney, P., Littman, M.L.: ACM Trans. on Information Systems. Measuring praise and criticism: Inference of semantic orientation from association 21, 315–346 (2003)
22. Yildirim, S., Narayanan, S., Potamianos, A.: Detecting emotional state of a child in a conversational computer game. Computer Speech and Language 25, 29–44 (2011)