Distributional Semantic Models for Affective Text Analysis

Nikolaos Malandrakis, *Student Member, IEEE*, Alexandros Potamianos, *Senior Member, IEEE*, Elias Iosif, *Student Member, IEEE*, and Shrikanth Narayanan, *Fellow, IEEE*

***Abstract*—We present an affective text analysis model that can directly estimate and combine affective ratings of multi-word terms, with application to the problem of sentence polarity/semantic orientation detection. Starting from a hierarchical compositional method for generating sentence ratings, we expand the model by adding multi-word terms that can capture non-compositional semantics. The method operates similarly to a bigram language model, using bigram terms or backing off to unigrams based on a (degree of) compositionality criterion. The affective ratings for n-gram terms of different orders are estimated via a corpus-based method using distributional semantic similarity metrics between unseen words and a set of seed words. N-gram ratings are then combined into sentence ratings via simple algebraic formulas. The proposed framework produces state-of-the-art results for word-level tasks in English and German and the sentence-level news headlines classification SemEval'07-Task14 task. The inclusion of bigram terms to the model provides significant performance improvement, even if no term selection is applied.**

***Index Terms*—Affect, affective lexicon, distributional semantic models, emotion, lexical semantics, natural language understanding, opinion mining, polarity detection, sentiment analysis, valence.**

I. Introduction

AFFECTIVE text analysis, the analysis of the emotional content of text, is an open research problem, relevant for numerous natural language processing (NLP), web and multimodal dialogue applications. One popular application is *sentiment analysis/opinion mining*, that aims to identify the emotion expressed in news stories [1], blogs and public forums [2] or product reviews [3], [4]. Generally opinion mining is restricted to separating positive from negative views (polarity detection), or positive, negative and neutral views. Opinion mining is focused on the emotion expressed by the writer (writer-perspective), rather than the emotion experienced by the reader. Emotion recognition from multimedia streams (audio, video, text) and emotion recognition of users of interactive applications (i.e., spoken language transcripts) is another task where the affective analysis of text plays an important, yet still limited role [5]–[7]. Other applications may focus on the reader/media consumer perspective, such as multimedia content analysis through subtitles [8] or news headlines analysis [9]. The requirements of different applications lead to the definition of different sub-tasks, such as emotional category labeling (assigning text a label, such as "sad"), polarity recognition (classifying into positive or negative) and subjectivity identification (separating subjective from objective statements). The affective task is further defined by the emotional representation used (e.g., basic emotions or valence) and the scope of analysis (word, sentence, document analysis).

Given the wide range of applications, affective text analysis has been a popular topic of research in the NLP and Semantic Web communities in recent years. The bulk of the research has focused on hierarchical lexical models, starting from words and working their way up to sentences or documents. Hierarchical lexical models require affective lexica that provide affective ratings for each word/term in the evaluation corpus. Manually annotated lexica, like the General Inquirer [10] and Affective norms for English Words (ANEW) [11] are too small for most applications, containing only 3600 and 1034 words, respectively. Computational methods are necessary to create or expand an already existing lexicon, creating much larger resources like SentiWordNet [12] and WORDNET AFFECT [13]. However, there are still limitations, e.g., WordNet-based efforts can not produce ratings for words not included in WordNet, including multi-word terms and proper nouns; the latter being particularly important when creating ratings for news' headlines.

Given an affective lexicon of sufficient coverage, sentence-level affective ratings are created by combining word-level ratings. Often, when lexica with continuous affective ratings are available, sentence-level ratings are estimated as simple numerical combinations of word ratings (typically the arithmetic mean). There have been attempts of using syntactic rules with encouraging results, e.g., [14], though such approaches are, so far, limited to using binary or tertiary word affective ratings. There has been very little research on modeling directly the affective content of multi-word terms [15], especially compounds with non-compositional semantic context, i.e., multi-word terms where their meaning cannot be expressed as a combination of the meaning of their constituent words.

We propose a method for affective lexicon expansion that can create ratings for both single and multi-word terms. By explicitly modeling the affective content of multi-word terms one can implicitly model the non-compositional semantics of modifiers and compounds (despite the lack of syntactic rules in the model). For instance, the proposed model generates a rating for

Manuscript received December 05, 2012; revised March 22, 2013; accepted July 16, 2013. Date of publication August 15, 2013; date of current version October 14, 2013. This work was supported in part by the IST Programme of the EU under contract number 296170 (PortDial project). The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Renato De Mori.

N. Malandrakis was with the Department of Electronic and Computer Engineering, Technical University of Crete, Chania 73100, Greece. He is now with the Department of Electrical Engineering and Signal and Image Processing Institute (SAIL), University of Southern California, Los Angeles, CA 90089 USA (e-mail: malandra@usc.edu).

A. Potamianos and E. Iosif are with the Department of Electronic and Computer Engineering, Technical University of Crete, Chania 73100, Greece (e-mail: potam@telecom.tuc.gr; iosife@telecom.tuc.gr).

S. Narayanan is with the Department of Electrical Engineering and Signal and Image Processing Institute (SAIL), University of Southern California, Los Angeles, CA 90089 USA (e-mail: shri@sipi.usc.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASL.2013.2277931

"not good" despite the lack of any explicit handling of negations. Given these fine-grained/pseudo-continuous valence ratings for all words and multi-word expressions contained in each sentence, we then use fusion algorithms to combine them into sentence ratings. The main contributions of this work are summarized below:

- We generalize the affective lexicon expansion method proposed in [16] to handle multi-word terms. The lexicon expansion method is language-agnostic, scalable and requires no resources other than a small affective lexicon to bootstrap the process. We also shed some light on the criteria for selecting good candidates (seed words) for the bootstrap lexicon.
- We show that context based semantic similarity estimated on a corpus of web snippets (with good coverage for all words in a language) significantly and consistently outperforms co-occurrence based metrics for affective modeling tasks. Also a detailed evaluation of co-occurrence, context and proximity as features for semantic similarity estimation for affective modeling applications is reported.
- Motivated by the language modeling literature, we propose a framework for combining n-gram ratings of varying order and utilizing multi-word term detection methods, to estimate sentence-level affective ratings. We use a structure similar to an n-gram language model with back-off and propose multi-word term selection criteria (for activating the back-off strategy).

The structure of this paper is as follows: Section II offers a brief review of prior research. Section III details our framework of word, n-gram and sentence rating creation. Section IV explains our experimental/validation procedure. Section V contains our experimental results and Section VI concludes the paper and proposes future research directions.

II. Prior Work

The task of assigning affective ratings, such as binary "positive – negative" labels, also known as semantic orientation [17] is an active research area. The underlying assumption for most semantic orientation algorithms is that *semantic similarity can be translated to affective similarity*. Therefore, given some metric of similarity between two words one may derive the similarity between their affective ratings. In [18], a set of words with known affective ratings together with the semantic similarities between these words and an unseen word are used to estimate affective ratings for the new word. The reference words that are used to bootstrap the affective model are usually referred to as *seed words*. The nature of the seed words can vary; they may be the lexical labels of affective categories (e.g., "anger," "happiness"), small sets of words with unambiguous meaning or even all words in a large lexicon. Having a set of seed words and an appropriate similarity measure, the next step is devising a method of combining these to create the final rating. In most cases the desired rating is some form of binary label like "fear" – "not fear," in which case a classification scheme, like *nearest neighbor* may be used to provide the final result. Alternatively, continuous/pseudo-continuous ratings may be estimated via algebraic combination of similarities and ratings of seed words [19].

TABLE I
The 14 Seeds Used in the Experiments by Turney and Littman.

positive:	good, superior, positive, correct, fortunate, nice, excellent
negative:	bad, inferior, negative, wrong, unfortunate, nasty, poor

In [18], [20], hit counts from conjunctive "NEAR" web queries are used to measure co-occurrence of words in web documents; semantic similarity is estimated for hits via point-wise mutual information. The estimated valence $\hat{v}(w_j)$ of each new word w_j is expressed as a linear combination of the valence ratings $v(w_i)$ of the seeds w_i and the semantic similarities between the new word and each seed $d(w_i, w_j)$ as;

$$\hat{v}(w_j) = \sum_{i=1}^{N} v(w_i) \cdot d(w_i, w_j), \tag{1}$$

The seeds used are $N = 14$ adjectives (7 pairs of antonyms) shown in Table I and their ratings are assumed to be binary (-1 or 1)[1].

WordNet-based methods [22], [23], [13], [24] start with a small set of annotated words, usually with binary ratings. These sets are then expanded by exploiting synonymy, hypernymy and hyponymy relations (traversal of the WordNet network) along with simple rules. Various approaches are then used to calculate the similarity between unseen words and the seed words, including using contextual similarity between glosses [22] and synset distance metrics [24]. The main benefit of resource-based methods is the ability to create ratings *per sense* of each word, however ratings can only be produced for words in WordNet.

Most of the aforementioned work utilizes the notion of semantic similarity between words or terms in order to infer affective ratings. Semantic similarity metrics can be roughly categorized into: i) ontology-based similarity measures, e.g., [25], where similarity features are extracted from ontologies (usually WordNet), ii) context-based similarity measures [26], where similarity of context is used to estimate semantic similarity between words or terms, iii) co-occurrence based similarity metrics where the frequency of co-occurrence of terms in (web) documents is the main feature used for estimating semantic similarity [18], [21], and iv) combinations of the aforementioned methods [27]. Context-based methods form the basis of distributional semantic models (DSM), distinguished into unstructured and structured types [28]. Unstructured approaches do not consider the linguistic structure of context: a window is centered on the target word and the surrounding contextual features within the window are extracted [29], [30]. For structured approaches the extracted contextual features correspond to syntactic relationships, which are typically extracted by dependency parsing and represented as word tuples [31], [28]. Recently corpus-based methods (especially context-based metrics) where shown to perform at par with ontology-based metrics [30], especially when using semantic networks as generalizations of distributional semantic models [32].

Having created an affective lexicon, the next step is the combination of these word ratings to create ratings for larger lexical units, phrases or sentences. Initially the affect-bearing words need to be selected, depending on their part-of-speech tags [33], affective rating and/or the sentence structure [34]. Then their individual ratings are combined, typically in a simple fashion, such as a numeric average. More complex approaches involve

[1] The method is shown to work very well in terms of binary (positive/negative) classification, achieving an 82.8% accuracy on the General Inquirer dataset. This method depends on the, now defunct, Altavista NEAR queries. As shown in [20], [21] the method performs much worse when using conjunctive AND queries instead.

taking into account sentence structure, word/phrase level interactions such as valence shifters [35] and large sets of manually created rules [33], [34]. In [14] a supervised method is used to train the parameters of multiple hand-selected rules of composition. However these complex methods have shown little improvement over simpler distributional approaches. Furthermore, the application of syntactic rules becomes prohibitively complex when using continuous word/sentence ratings: even the simplest of rules would require multiple parameters/cases.

III. Affective Model

As in [18], we start from an existing, manually annotated lexicon. A subset of words is automatically selected from the lexicon to serve as seed words for the affective model. The affective rating for a new word/term is estimated as a linear combination of the products between semantic similarities and affective ratings of the seed words. We modify the method in [18] by adding: i) weights to the equation, one per seed word, so as to adjust each seed word's contribution to the final output, and ii) a function (kernel) that modifies the semantic similarity score contribution to the model. The weights are selected so as to minimize the mean square training error.

The trainable weights are meant to capture the relevance of each seed word in the affective model. For instance, a seed word with high affective (or semantic) variance might be a less robust predictor of the affective scores of unseen words. Words with high affective variance typically have multiple part-of-speech tags and word senses, or their valence rating is highly context-dependent. In addition, a set of seed words might not provide a detailed and representative description of the affective/semantic space, e.g., selecting only words with positive valence scores significantly hurts performance of the model[2]. Rather than attempting to estimate the individual contribution of each parameter to the relevance of seed words in our model, we use machine learning to automatically estimate linear weights for each seed word. The weights are estimated in order to minimize estimation error on the bootstrap affective lexicon using the Least Squares Estimation (LSE) algorithm, as detailed below. A simplified version of the affective model was first proposed in [16].

A. Word Level Tagging

We want to characterize the affective content of words in a continuous valence range of $[-1, 1]$ (from very negative to very positive), *from the reader (i.e., perceiver) perspective*. We model the valence of each word as a linear combination of its semantic similarities to a set of seed words and the valence ratings of these words:

$$\hat{v}(w_j) = a_0 + \sum_{i=1}^{N} a_i \; v(w_i) \; f(d(w_i, w_j)), \tag{2}$$

where w_j is the word we aim to characterize, $w_1, \ldots, w_N$ 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), $d(w_i, w_j)$ is a measure of semantic similarity between words w_i and w_j (see Section III-A) and $f(\bullet)$ is a simple function from Table II. The function $f(\bullet)$ serves to non-linearly rescale the similarity metric $d(w_i, w_j)$ and will be henceforth referred to as the kernel of the affective model.

TABLE II
The Functions of Similarity Used.

linear	$f(d(\bullet)) = d(\bullet)$
exp	$f(d(\bullet)) = e^{d(\bullet)}$
log	$f(d(\bullet)) = log(d(\bullet))$
sqrt	$f(d(\bullet)) = \sqrt{d(\bullet)}$

$$\begin{bmatrix} 1 & f(d(w_1, w_1))v(w_1) & \cdots & f(d(w_1, w_N))v(w_N) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & f(d(w_K, w_1))v(w_1) & \cdots & f(d(w_K, w_N))v(w_N) \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_N \end{bmatrix} = \begin{bmatrix} 1 \\ v(w_1) \\ \vdots \\ v(w_K) \end{bmatrix} \tag{3}$$

Assuming we have a training corpus of K words with known ratings (the manually annotated affective lexicon we start from) and a set of $N < K$ seed words (a subset of the lexicon) for which we need to estimate weights a_i, we can use (2) to create a system of K linear equations with $N+1$ unknown variables as shown in (3); the N weights $a_1, \ldots, a_N$ and the extra weight a_0 which is the shift (bias). The optimal values of these variables can be estimated using LSE. Once the weights of the seed words are estimated the valence of an unseen word w_j can be computed using (2). Note that no additional training corpus or data are required here, the weights are estimated on the affective lexicon used to bootstrap the model.

The valence estimator defined in (2) uses a metric $d(w_i, w_j)$ of the semantic similarity between words w_i and w_j. In this work, we use both co-occurrence based and context-based similarity metrics.

1) Co-Occurrence Based Similarity Metrics: estimate the similarity between two words/terms using the frequency of co-existence within larger lexical units (sentences, documents). The underlying assumption is that terms that co-exist often are likely to be related semantically. One popular method to estimate co-occurrence is to pose conjunctive queries to a web search engine; the number of returned hits is an estimate of the frequency of co-occurrence [30]. Co-occurrence based metrics do not depend on annotated language resources like ontologies nor require downloading documents or snippets, as is the case for context-based semantic similarities.

In the equations that follow, $w_i, \ldots, w_{i+n}$ are the query words, $\{D; w_i, \ldots, w_{i+n}\}$ is the set of results $\{D\}$ returned for these query words. The number of documents in each result set is noted as $|D; w_i, \ldots, w_{i+n}|$. We investigate the performance of four different co-occurrence based metrics, defined next.

Jaccard Coefficient: computes similarity as:

$$J(w_i, w_j) = \frac{|D; w_i, w_j|}{|D; w_i| + |D; w_j| - |D; w_i, w_j|}. \tag{4}$$

Dice Coefficient: is a variation of the Jaccard coefficient, defined as:

$$C(w_i, w_j) = \frac{2\,|D; w_i, w_j|}{|D; w_i| + |D; w_j|}. \tag{5}$$

Mutual Information: [27] is an info-theoretic measure that derives the similarity between w_i and w_j via the dependence

[2] For more details on the effect of these factors on seed selection, see Section V-C.

between their number of occurrences. Point-wise Mutual Information (PMI) is defined as:

$$I(w_i, w_j) = \log \frac{\frac{|D;w_i,w_j|}{|D|}}{\frac{|D;w_i|}{|D|}\frac{|D;w_j|}{|D|}}. \qquad (6)$$

Mutual information is unbounded and can take any value in $[-\infty, +\infty]$. Positive values translate into similarity, negative values into dissimilarity (presence of one word tends to *exclude* the other) and zero into independence, lack of relation.

Google-Based Semantic Relatedness : Normalized Google Distance is proposed in [36], [37] and defined as:

$$E(w_i, w_j) = \frac{\max\{L\} - \log|D; w_i, w_j|}{\log|D| - \min\{L\}}, \qquad (7)$$

where $L = \{\log \mid D; w_i \mid, \; \log \mid D; w_j \mid\}$. This metric is unbounded, taking values in $[0, +\infty]$. In [38], a bounded (in [0, 1]) similarity metric is proposed based on Normalized Google Distance called Google-based Semantic Relatedness, defined as:

$$G(w_i, w_j) = e^{-2E(w_i, w_j)}. \qquad (8)$$

2) Context-Based Similarity Metrics: compute similarity between feature vectors extracted from term context, i.e., using a "bag-of-words" context model, using a metric like cosine similarity or Kullback-Leibler divergence. The basic assumption behind these metrics is that similarity of context implies similarity of meaning, i.e., terms that appear in similar lexical environment (left and right contexts) have a close semantic relation [39], [26]. "Bag-of-words" [40] models assume that the feature vector consists of words or terms that occur in text independently of each other. The context-based metrics presented here employ a context window of fixed size (H words) for feature extraction. Specifically, the right and left contexts of length K are considered for each occurrence of a word or term of interest w in the corpus, i.e., $[v_{K,L} \ldots v_{2,L} \; v_{1,L}]w[v_{1,R} \; v_{2,R} \ldots v_{K,R}]$ where $v_{i,L}$ and $v_{i,R}$ represent the ith word to the left and to the right of w, respectively. The feature vector for word or term w is defined as $T_{w,H} = (t_{w,1}, t_{w,2} \ldots t_{w,V})$ where $t_{w,i}$ is a non-negative integer and H is the context window size. Note that the length of the feature vector is equal to the vocabulary size V, i.e., all words in the vocabulary are features. The ith feature value $t_{w,i}$ reflects the (frequency of) occurrence of vocabulary word v_i within the left or right context window of (all occurrences of) the term w. The value of $t_{w,i}$ may be defined as a (normalized or unnormalized) function of the frequency of occurrence of feature i in the context of w. Once the feature weighting scheme is selected, the "bag-of-words"-based metric S^H computes the similarity between two words or terms, w_1 and w_2, as the cosine similarity of their corresponding feature vectors, $T_{w_1,H}$ and $T_{w_2,H}$ as follows, [40]:

$$S^H(w_1, w_2) = \frac{\sum_{i=1}^{V} t_{w_1,i} t_{w_2,i}}{\sqrt{\sum_{i=1}^{V} (t_{w_1,i})^2} \sqrt{\sum_{i=1}^{V} (t_{w_2,i})^2}} \qquad (9)$$

where H is the context window length and V is the vocabulary size. The cosine similarity metric assigns 0 similarity score when w_1, w_2 have no common context (completely dissimilar words), and 1 for identical words. Various feature weighting schemes can be used to compute the value of $t_{w,i}$. The binary weighting metric used in this work assigns weight $t_{w,i} = 1$ when the ith word in the vocabulary exists at the left or right context of at least one instance of the word w, and 0 otherwise. Alternative weighting schemes such as tf-idf are more popular, but we opt for binary weights that perform best in semantic similarity tasks [30], [41] and are computationally simpler.

B. Multi-Word Term Tagging

So far we have used the terms "word" and "term" interchangeably when referring to the targets of the method described in Section III-A. The method has no requirement that would limit us to estimating word ratings or even limit us to the English language: it can work for any term of any length and for any language as long as we have a starting affective lexicon and an appropriately large text corpus. When applying to bigrams, only the semantic similarity metric has to be extended to handle both unigrams and bigrams. In principle, the co-occurrence and context-based metrics $d(\cdot)$ used for unigrams can be also used to estimate the semantic similarity between n-grams[3].

C. Sentence Level Tagging

We assume that the affect rating of sentence $s = w_1 w_2 \ldots w_N$ can be estimated via the composition [42] of the affective scores of its constituent words w_i. The simplest fusion model (and also by far the most popular) is a simple linear combination of the partial ratings:

$$v_a(s) = b_0 + b_1 \, \frac{1}{N} \sum_{i=1}^{N} v(w_i), \qquad (10)$$

where b_0 and b_1 are trainable weights corresponding to an offset and unigrams w_i respectively. Linear fusion assumes that words should be weighted equally independently of their strong or weak affective content. As a result, a sentence containing only a few strongly polarized terms might end up having low absolute valence (due to averaging). Next, we propose a weighted average scheme, where terms with higher absolute valence values are weighted more:

$$v_w(s) = b_0 + b_1 \, \frac{1}{\sum_{i=1}^{N} |v(w_i)|} \sum_{i=1}^{N} v(w_i)^2 \cdot \text{sign}(v(w_i)), \qquad (11)$$

where $\text{sign}(.)$ is the signum function. One could also generalize to higher powers or to other non-linear scaling functions. Next, we consider non-linear min-max fusion, where the term with the highest absolute valence value dominates the meaning of the sentence:

$$v_{\text{m}}(s) = b_0 + b_1 \; v(w_z), \quad z = \arg\max_i(|v(w_i)|), \qquad (12)$$

where $\arg\max$ is the argument of the maximum. One could also consider combinations of linear and non-linear fusion methods, as well as, syntactic- and pragmatic-dependent fusion rules.

The use of the simple fusion schemes proposed above with only the words of each sentence, carries the implicit assumption of a compositional model of semantics and affect. Specifically, estimating the affective score of a sentence is assumed to

[3] The generalization is straightforward for context-based metrics and indeed such metrics have been successfully used to estimate the semantic similarity between multi-word terms [30]. However, for co-occurrence based metrics that use word counts, the mean and dynamic range of similarity scores is very different between unigrams and bigrams, making their fusion a challenge (see also Section III-D). No bigram seed words are necessary to bootstrap the model.

be simply a problem of appropriately scaling the contribution of each word's affective score to estimate a sentence level score. Although the compositionality assumption might be reasonable in many cases (and as we shall see in Section V produces good results), there are many cases of compound expressions where their semantic and affective content cannot be accurately estimated as a (weighted) sum of its words. Such examples include: 1) modifiers such as negation that can alter the meaning and (reverse) affective scores, and 2) idiomatic multi-word expressions that cannot be semantically parsed word-for-word[4]. To address these concerns, we extend the above models to using terms (of length n) instead of just words: a model using n-grams instead of unigrams (words) will attempt to combine the partial ratings of all *overlapping* n-grams within a sentence.

D. Fusion of n*-Gram Models*

In this section, we attempt to improve on the performance of unigram- and bigram-only affective models by utilizing them as building blocks to create models that employ unigrams and bigrams. The proposed fusion algorithms are motivated by language modeling. Here, instead of n-gram probabilities (for language models), we are combining affective scores. The main fusion strategies, however, are similar: 1) interpolation of the valence scores of the unigrams and bigrams (or higher-order), and 2) back-off from bigrams to the unigrams when a certain criterion is satisfied. Much like language modeling the back-off criterion should be related to n-gram counts. For affect, additional criteria may be devised that are related with the "degree of compositionality" (semantic or affective) of each n-gram. For bigrams that appear rarely in our corpus it may be advantageous to back-off to a unigram where adequate statistics to accurately estimate affective scores exist.

1) Interpolation: For sentence s that consists of the word sequence $w_1 w_2 \dots w_N$ we create a unigram λ_1 and bigram λ_2 affective model, respectively, that estimate the sentence level affective score as follows[5]:

$$v(s|\lambda_1) = \frac{1}{N}\sum_{i=1}^{N} v(w_i)$$
$$v(s|\lambda_2) = \frac{1}{N-1}\sum_{i=1}^{N-1} v(w_i w_{i+1}) \tag{13}$$

where the valence $v(w_i)$ of word w_i and the valence $v(w_i w_{i+1})$ of bigram $w_i w_{i+1}$ are both estimated using (2). We combine the scores of the unigram and bigram models as follows:

$$v_{\mathrm{in}}(w_i w_j) = b_1\ v(w_i w_j|\lambda_1) + b_2\ v(w_i w_j|\lambda_2), \tag{14}$$
$$v_{\mathrm{in}}(s) = b_0 + \frac{1}{N}\left[\frac{b_1}{2}(v(w_1)+v(w_N)) + \sum_{i=1}^{N-1} v_{\mathrm{in}}(w_i w_{i+1})\right] \tag{15}$$

where b_i are linear weights that can be estimated via machine learning on held-out data and the term $(1/2)b_1(v(w_1)+v(w_N))$ serves the need to use each unigram in the sentence an equal amount of times (by adding the ratings of the first and last unigram). It is straightforward to extend the proposed method to higher order n-gram models.

2) Back-Off: Here instead of interpolating the affective scores of different n-gram models, we propose a criterion for alternating between the unigram and bigram model [43]. Specifically we define the selection criterion $c(i,j)$ for bigram $w_i w_j$; we utilize bigram $w_i w_j$ if $c(i,j)$ is larger than some threshold t or revert to the unigrams w_i and w_j otherwise, i.e.,

$$v_{\mathrm{bo}}(w_i w_j) = \begin{Bmatrix} b_1\ v(w_i w_j|\lambda_1), & c(i,j) \le t \\ b_2\ v(w_i w_j|\lambda_2), & c(i,j) > t \end{Bmatrix}, \tag{16}$$

where b_1 and b_2 are the trainable weights of the unigram and bigram models respectively. After performing term selection, we combine the scores:

$$v_{\mathrm{bo}}(s) = b_0 + \frac{1}{N}\left[\frac{1}{2}b_1(v(w_1)+v(w_N)) + \sum_{i=1}^{N-1} v_{\mathrm{bo}}(w_i w_{i+1})\right]. \tag{17}$$

The criterion $c(i,j)$ for selecting the appropriate n-gram model utilizes both the frequency of occurrence of the n-gram in our corpus and the degree of compositionality of the n-gram. Specifically, the following criteria are proposed:

1) The probability of occurrence of the bigram $w_i w_j$:

$$c_{\mathrm{p}}(i,j) = p(w_i w_j). \tag{18}$$

2) A mutual information-like criterion that measures the probability of co-occurrence of words w_i and w_j (a simple measure of compositionality):

$$c_{\mathrm{m}}(i,j) = \frac{p(w_i w_j)}{p(w_i)p(w_j)}. \tag{19}$$

3) The absolute difference between the valence scores estimated via the bigram and unigram models (a measure of affective compositionality):

$$c_{\mathrm{nc}}(i,j) = |v(w_i w_j) - 0.5[v(w_i) + v(w_j)]|. \tag{20}$$

Note that the n-gram frequency-based criterion $c_{\mathrm{p}}()$ can be combined with the degree of compositionality criteria $c_{\mathrm{m}}()$ and/or $c_{\mathrm{nc}}()$ producing the following criteria:

$$c_{ts}(i,j) = p(w_i w_j)\ \log c_{\mathrm{m}}(i,j),$$
$$c_{\mathrm{pnc}}(i,j) = p(w_i w_j)\ \log c_{\mathrm{nc}}(i,j). \tag{21}$$

The thresholds $t_{\mathrm{p}}, t_{\mathrm{m}}, \dots, t_{\mathrm{pnc}}$ are estimated for each criterion on held-out data.

3) Weighted Interpolation: Weighted interpolation extends the interpolation and back-off models. Similarly to the back-off model we use a compositionality criterion $c(i,j)$ for bigram $w_i w_j$, however in weighted interpolation bigram and corresponding unigram ratings are interpolated when $c(i,j)$ is over a threshold t:

$$v_{\mathrm{wi}}(w_i w_j) = \begin{Bmatrix} \mathcal{V}_1, & c(i,j) \le t \\ \mathcal{V}_1 + \mathcal{V}_2, & c(i,j) > t \end{Bmatrix}, \tag{22}$$

where $\mathcal{V}_1 = b_1\, v(w_i w_j|\lambda_1)$ and $\mathcal{V}_2 = b_2\, v(w_i w_j|\lambda_2)$. The final collection of terms will include all unigrams in the sentence and

[4] Note that deviation from the expected meaning and affective content of a multi-word expression may also occur due to contextual or pragmatic constraints, e.g., "wicked" can have high positive valence in certain contexts. However, such semantic/affective variability can occur both for words and multi-word expressions and are not treated directly here.

[5] For simplicity, we only present the equations for the simple linear model. It is straightforward to generalize to non-linear fusion schemes.

some bigrams (appropriately weighted). As before, we combine the selected terms to produce the final sentence rating:

$$v_{\mathrm{wi}}(s) = b_0 + \frac{1}{N}\left[\frac{1}{2}b_1(v(w_1)+v(w_N)) + \sum_{i=1}^{N-1} v_{\mathrm{wi}}(w_i w_{i+1})\right]. \tag{23}$$

IV. Experimental Procedure

Next we present the corpora used for training and evaluation of the proposed algorithms. In addition, the experimental procedure for semantic similarity computation, affective lexicon creation and sentence-level affective score computation is outlined.

A. Corpora

The main corpus used for creating the affective lexicon is the *Affective Norms for English Words* (ANEW) dataset. ANEW consists of 1034 words, rated in 3 continuous dimensions of arousal, valence and dominance. In this work, we only use the valence ratings provided in ANEW[6]. Looking at quantized values, the dataset contains 586 positive and 448 negative words.

The second corpus used for evaluation of the affective lexicon creation algorithm is the *General Inquirer* (GINQ) corpus that contains 2005 negative and 1636 positive words. The General Inquirer corpus 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 [18], [20]. After removing the words that overlap with ANEW, we are left with 1443 positive and 1754 negative words.

To evaluate the lexicon creation method on a non-English dictionary, we used the *Berlin Affective Word List Reloaded* (BAWL-R) dataset. BAWL-R contains 2902 German words annotated in continuous scales (we use only valence). In quantized form, the set contains 1636 positive and 1266 negative words.

For the sentence level tagging task the *SemEval 2007: Task 14* corpus is used [9]. This SemEval corpus contains news headlines, 250 in the development set which are used for training and 1000 in the testing set which are used for evaluation. The headlines are manually rated in a fine-grained valence scale of $[-100, 100]$, which is rescaled to $[-1, 1]$ for our experiments. In quantized form the set contains 474 positive and 526 negative samples.

B. Corpus Creation and Semantic Similarity

In our experiments we utilized four different similarity metrics based on web co-occurrence, mentioned in Section III-A, namely, *Dice coefficient, Jaccard coefficient, point-wise mutual information (PMI) and Google-based Semantic Relatedness* as well as a single contextual similarity metric, cosine similarity with binary weights.

All similarity metrics employed require a corpus in order to calculate frequency statistics or collect context. In this work we use three corpora derived from the web and created by submitting queries to the Yahoo! search engine and collecting the response.

The first corpus is the web, which is only used to compute co-occurrence based similarities. Co-occurrence based similarity metrics require the individual (IND) words' number of occurrences as well as the number of times that the two words co-exist within a set distance. Usually this distance is unlimited (anywhere within a document); this method is used by the AND operator of web search engines. However it is possible to limit that distance, e.g., the Altavista NEAR operator used to obtain co-occurrence in [20] limited co-occurrence to a distance of 10 words. The alternative we used was the Yahoo! NEAR operator, which was an undocumented feature of the Yahoo! engine. This corpus will be henceforth referred as "web."

Using the web directly poses practical challenges. The vast number of queries required can have a significant cost (in terms of both time and monetary cost). More importantly, the desirable distance-limited joint queries are not supported by most search engines: we obtained enough data for our experiments from the Yahoo! engine, however as of this writing the Yahoo! engine no longer supports the NEAR operator. To alleviate these problems we created two more corpora by posing IND queries to the Yahoo! search engine and collected the top $|D|$ (if available) snippets (the short excerpts (page samples) shown under each result, typically one or two sentences automatically selected by the search engine) for each word. Each snippet contains at least two sentences from the result: the title and a preview of the content. The second and third corpora were built using this process.

The second corpus is task dependent: we created a vocabulary that contained all words in (all) our evaluation corpora, posed a single IND query for each of them to Yahoo! and collected 1000 snippets (where available) from each query. The corpus contains 14 million sentences and was indexed using the Lucene indexing engine [44], effectively creating a local search engine. This 14 m corpus was used to compute both co-occurrence based and context-based similarities. Using the 14 m snippet corpus one can emulate hit counts obtained by NEAR queries (on the "web" corpus), e.g., by estimating co-occurrence counts within the same sentence. To compute context-based similarities, the left and right contexts of all occurrences of w_1 and w_2 are examined and the corresponding feature vectors are constructed. The parameters of context based metrics are the number $|D|$ of web documents used and the size K of the context window. In all experiments presented in this work $|D| = 1000$, whereas the values used for K are 1, 2, 5 and 10. This corpus will be noted henceforth as 14 m.

The third corpus is created similarly to the second one, by collecting snippets, however it is task-independent: to create it we used a vocabulary of the English language, specifically the one that comes with the Aspell spellchecker [45] containing 135,433 words. For each of them we posed an IND query and collected up to 500 snippets. The final corpus contains 116 million sentences and will be noted henceforth as 116 m. As with 14 m, the downloaded text was indexed with Lucene and used to compute both co-occurrence based and context-based similarities. This corpus was created[7] as part of the EU-IST PORTDIAL project http://www.portdial.eu/.

C. Affective Lexicon and Word Affective Ratings

The following tasks and associated experimental setup have been used for model training and performance evaluation in this work:

[6] The method is applicable to arousal and dominance, however for the purposes of this work we focus on the more popular dimension of valence. Valence was selected over arousal and dominance due to its greater applicability and larger volume of prior work enabling comparisons.

[7] The main motivation behind creating this large task independent corpus is that the performance of semantic similarity metrics has been shown to improve due to better coverage of rare word senses of common words and more uniform word occurrence probabilities. For more details see [32].

TABLE III
TRAINING SAMPLE USING 10 SEED WORDS.

w_i	$v(w_i)$	a_i	$v(w_i) \times a_i$
triumphant	0.96	1.48	1.42
rape	-0.94	0.72	-0.67
love	0.93	0.57	0.53
suicide	-0.94	3.09	-2.91
paradise	0.93	1.77	1.65
funeral	-0.90	0.53	-0.48
loved	0.91	1.53	1.40
rejected	-0.88	0.50	-0.44
joy	0.90	1.00	0.90
murderer	-0.87	1.99	-1.73
w_0 *(offset)*	1	-0.06	-0.06

- ANEW-CV: 10-fold cross-validation on the ANEW dataset, i.e., model training and evaluation on the ANEW dataset.
- GINQ-PD: model training on the ANEW dataset, evaluation on GINQ dataset.
- BALWR-CV: 10-fold cross-validation on the BAWL-R dataset.

In all cases the seed words were selected from the training set (training fold in the case of cross-validation), therefore on cross-validation experiments the seeds are different for each fold. Given a set of candidate seeds (in most cases the entire training set), we applied a simple method to select the desired seeds. It seems, looking at Turney and Littman's method [18], but also confirmed by our experiments, that good seeds need to have a high absolute valence rating. It also proved beneficial to ensure that the seed set is as close to *balanced* (sum of seed valence is zero) as possible. Therefore our selection method started by sorting the positive and negative seeds separately by their valence rating. Then positive and negative seeds were added to the seed set iteratively so as to minimize the absolute value of the sum of their valence ratings, yet maximize their absolute valence ratings (or frequencies), until the required number N was reached. More on seed selection is given in Section V-C.

The semantic similarity between each of the N seed words 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 (3) were estimated using LSE. Finally, for each word in the test set the valence ratings were computed using (2) and evaluated against the ground truth.

A toy training example using $N = 10$ features and the Google semantic relatedness co-occurrence based metric is shown in Table III. The second column $v(w_i)$ shows the manually annotated valence of word w_i, while the third column a_i shows the corresponding linear weight computed by the LSE 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 (2)). The last row in the table corresponds to the bias term a_0 in (2) that takes a small positive value. Note that the coefficients a_i take positive values and are not bounded in $[0, 1]$, although similarity metrics are bounded at $[0, 1]$ and target valence values are also bounded in $[-1, 1]$. There is no obvious intuition behind the a_i scores, e.g., it is not clear why "suicide" should receive much higher weighting than "funeral." The weights might be related to the semantic and affective variance of the seed words.

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 and (ii) binary classification accuracy of positive vs. negative relations, i.e., continuous ratings are produced, converted to binary decisions and compared to the ground truth. Statistical significance testing was conducted using the paired sample t-test (right-sided) for the cross-validation experiments and McNemar's test for non cross-validation. Unless mentioned otherwise, we set the statistical significance threshold at $p < 0.001$.

$$\underbrace{\underset{0.57}{watching}\ \underset{0.71}{cute}\ \underset{0.50}{puppies}\ \underset{0.00}{makes}\ \underset{-0.11}{me}\ \underset{0.82}{happy}}_{\text{[linear: } 0.41\text{, weighted average: } 0.64\text{, max: } 0.82\text{]}}$$

Fig. 1. Example of word rating fusion, showing the per-word ratings and the phrase ratings produced by the three unigram fusion schemes.

D. Sentence Affective Ratings

The *SemEval'07-Task 14* corpus was used to evaluate the various n-gram fusion methods. All unseen words/terms in the sentence corpus were added to the lexicon using the affective lexicon expansion algorithm outlined above (3983 unigrams and 6630 bigrams overall). The model used to create the required ratings was trained using all of the words in the ANEW corpus as training samples and N of them as seed words. Then the ratings of each term in the sentence were combined to create the sentence rating. We employed content word selection when considering unigram terms: unigrams that were not nouns, verbs, adjectives or adverbs were ignored. To identify content words part-of-speech tagging was performed using *TreeTagger* [46]. A toy example can be seen in Fig. 1.

In order to evaluate the performance of the sentence level affective scores we used the classification accuracy for the 2-class (positive, negative) problem. Statistical significance testing was conducted using McNemar's test.

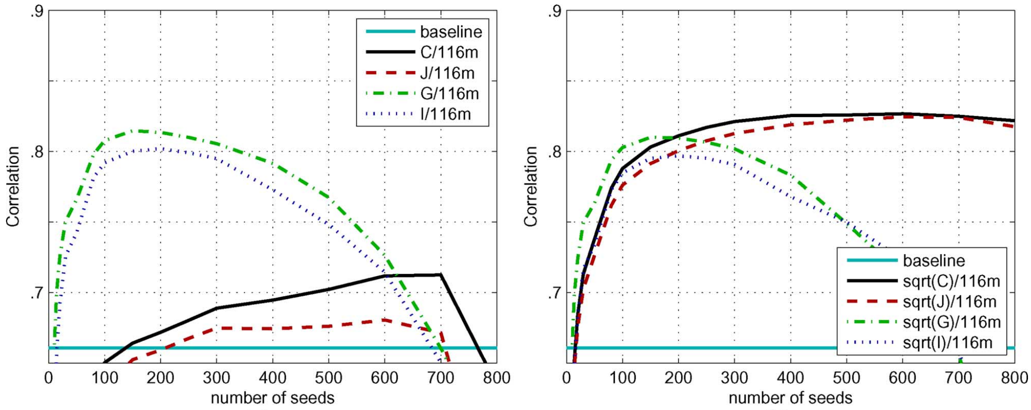
V. RESULTS

In this section, we evaluate the proposed algorithms on a variety of word- and sentence-level affective tasks. The following issues are investigated: i) the relative performance of the co-occurrence and context-based semantic similarity metrics for estimating continuous valence ratings of words, ii) the effect of corpus size and type on performance, iii) how to select seed words for the affective model and iv) the performance of various unigram- and bigram-level fusion strategies (interpolation, back-off, weighted interpolation) for obtaining sentence-level affective ratings.

A. Baseline Performance

The baseline performance for word-level affective tasks is that of the method proposed in [18], [20]. The 14 words shown in Table I were used as seeds, as well as, co-occurrence similarities (mutual information metric I) estimated via NEAR[8] web queries. In [20], the binary classification accuracy for the GINQ dataset was reported at 82.8%; our implementation yielded somewhat higher performance at 84%. In addition, the same setup was run for the ANEW task achieving 0.66 correlation and 82% binary accuracy performance. We do not report baseline performance for the BAWL-R experiment (since no seed words were proposed for German in [20]).

[8] Note that using NEAR conjuctive queries was essential to achieving good performance using this method.



(a) (b)

Fig. 2. Performance of the affective lexicon creation algorithm using similarities based on co-occurrence counts from the 116 m corpus. Correlation for the ANEW-CV experiment using: (a) a linear kernel and (b) a square root kernel.

B. *Similarity Metric Selection*

The first and arguably most important parameter of the affective model detailed in Section III-A is the semantic similarity metric used. The related method in [20] uses the mutual information similarity metric I estimated via NEAR web queries. In our initial experiments, we have observed significant performance differences between various co-occurrence based metrics, e.g., see [16]. In addition to the type of similarity metric used, the similarity estimation method also significantly affects performance; most importantly the size and type of corpus used to calculate statistics and term proximity. All experiments reported in this section are for the ANEW-CV task using correlation with human judgments as the performance metric.

The co-occurrence based similarity metrics used in this work are the same as those used in [16], however, the method for estimating co-occurrence counts has been updated. Specifically, we can estimate co-occurrence counts either using web hits (web) or on a corpus of snippets created via web queries over vocabulary lists of various sizes (14 m, 116 m). Performance of the Dice C, Jaccard J, mutual information I and Google G co-occurrence metrics, when calculated over the 116 m corpus[9], was evaluated on ANEW-CV, using a linear model kernel, is shown in Fig. 2(a) as a function of the number of seed words[10]. The relative performance of the similarity metrics are similar to those reported in [16], with I and G performing significantly better ($p < 10^{-16}$) than J and C. All metrics perform better than the baseline (see Section V-A) provided that a few hundred seeds are used to bootstrap the model. Note however, that metrics J and C are more robust to seed selection process, performance is flat over a wide range of number of seeds. This can be rectified by using the model kernels defined in Table II. Performance using a square root kernel is shown in Fig. 2(b). The non-linear rescaling has a significant effect ($p < 10^{-16}$) on performance when using the C and J similarity metrics: when using a logarithmic or square root kernel they can reach or, in some cases, overtake G and I, though overall G and I still prove to be better choices. Kernels can also improve performance of the best performing similarity metrics, however the differences are smaller and less consistent. G and I perform very similarly in most cases, with a slight edge to G.

From previous work, e.g., [21], [16], it is clear that word proximity is an important feature when estimating semantic similarities for the affective model. Restricting the search engine so that it registers co-occurrence when w_i and w_j occur within a small distance (NEAR queries), rather than when they co-occur within a document at any distance (AND queries) provides a noticeable performance boost. Next we investigate the optimal co-occurrence distance. For this experiment we estimated similarities using the (best-performing) G similarity metric on the 116 m corpus. Results are reported on the ANEW-CV task for various distance requirements: accepting co-occurrence if the term distance is up to n or alternatively if the term distance is exactly n. The results are shown in Fig. 3(a) as a function of the number of seeds. As expected close proximity is an important feature, with best performance of the "equal to" experiment achieved for distance 2, while for the "up to" experiment there is virtually no performance gain at distances over 5. There is no performance drop when moving to larger distances, however this is an artifact of the snippet corpus[11].

Corpus size and type also significantly affect similarity estimation and model performance. In Fig. 3(b), we report the correlation performance on the ANEW-CV task using the G metric estimated on each of the three available corpora (web, 14 m and 116 m). NEAR queries are used to obtain the co-occurrence statistics for the web corpus, while co-occurrence at the snippet level is computed for the 14 m and 116 m corpora. Performance for similarities estimated on the large corpus (116 m) are significantly better ($p < 10^{-3}$ over 300 seeds) than for the small corpus (14 m). The 116 m and web corpora have similar performance for a few hundred seeds (around 300 seeds), yet the 116 m corpus achieves better performance for fewer seeds and higher top performance. These results further validate the use of a corpus as a substitute for the elusive[12] web NEAR queries.

Next we investigate the performance of context based similarity metrics as a function[13] of context window length K

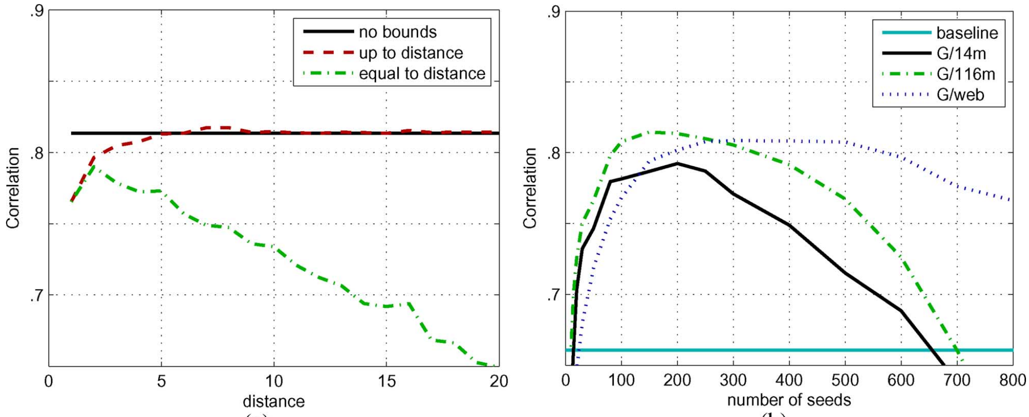
[9] Given that the corpus is composed of independent sentences, instead of full documents, the co-occurrence statistics are very similar to the result of a NEAR query (since we will only get a hit if the two terms co-occur within the same sentence).

[10] Seed selection is performed here using the heuristic of maximum absolute valence score and zero mean valence over all seed words, as detailed in Section IV-C.

[11] Moving beyond sentence boundaries, e.g., defining co-occurrence at the document level, significantly reduces the performance of semantic similarity based affective models, as shown in [21], [16].

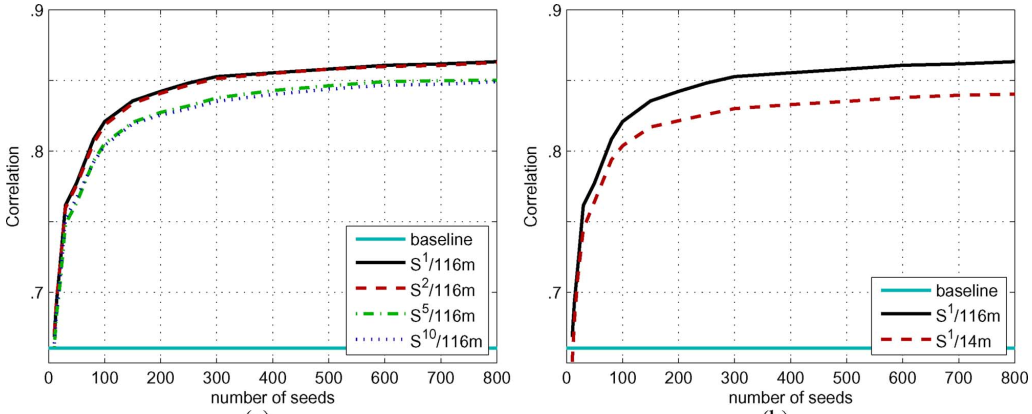
[12] As discussed in Section IV-B the Yahoo! NEAR querying functionality was an undocumented feature of the engine, that has been recently removed.

[13] Different lexical weighting schemes for context vectors have been also investigated (not reported here). As expected, the binary weighting scheme performed best for affect classification, as was the case also for semantic similarity estimation tasks in [30], [32].



(a) (b)

Fig. 3. Performance of the affective lexicon creation algorithm using co-occurrence based similarities. Correlation for the ANEW-CV experiment using: (a) the 116 m corpus and different window sizes at 150 seeds and (b) corpora of different sizes.



(a) (b)

Fig. 4. Performance of the affective lexicon creation algorithm using context-based similarities. Correlation for the ANEW-CV experiment using: (a) the 116 m corpus and different window sizes and (b) a window size of 1 and corpora of different sizes.

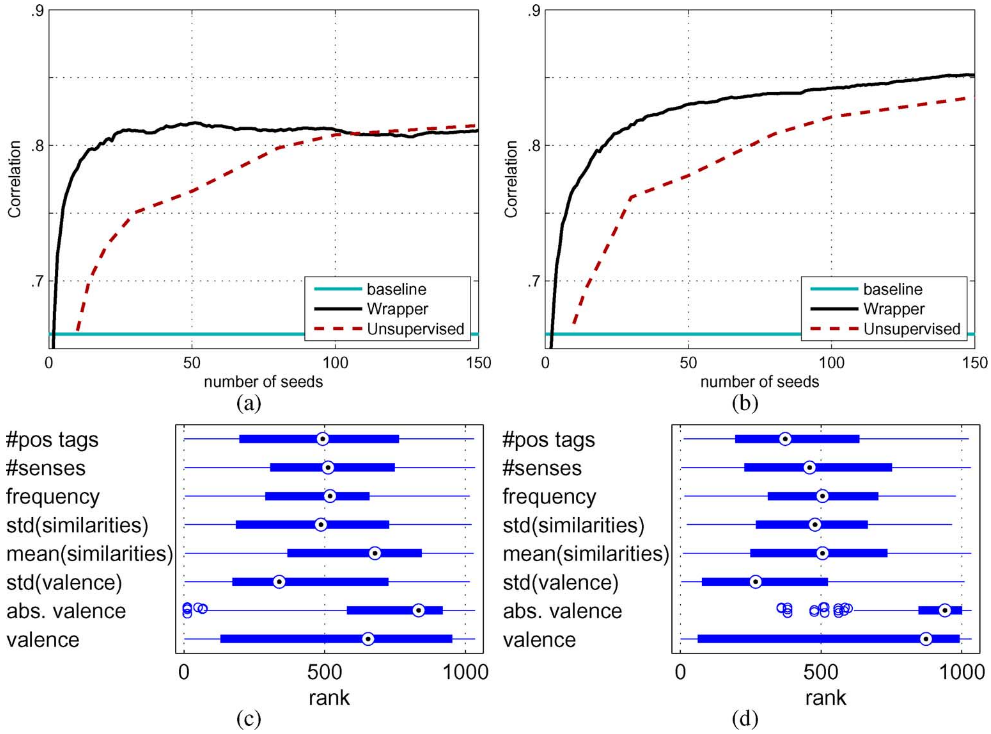
and corpus size (14 m, 116 m). Correlation performance on the ANEW-CV task is shown in Fig. 4(a) for context based metrics with window lengths of $K = 1, 2, 5, 10$. The best performance is consistently obtained for small window sizes of $K = 1$, 2. This is consistent with the results in [30], [32], where $K = 1$ provided the best performance for a word-level semantic similarity task. Correlation performance when context based similarities are evaluated on the 14 m or 116 m corpus is shown in Fig. 4(b) as a function of number of seed words. Estimating context vectors on the larger corpus significantly ($p < 10^{-3}$ over 100 seeds) outperforms the smaller corpus. For a more detailed analysis on why a large task-independent corpus is expected to provide better performance for semantic similarity estimation tasks see [32].

Based on the results reported in this section, henceforth we focus our attention on the Google co-occurrence based semantic similarity metric G and the binary weighted context based semantic similarity metric S^1 with context window $K = 1$. Next, results are reported in terms of correlation and binary classification accuracy for G, S^1 for a variety of word-level and sentence-level affective tasks.

C. Seed Word Selection

Seed words act as points of reference in affective space, relative to which all other words are rated. As such, their selection from a set of candidates is an important step of the rating creation process. Next we try to answer the question of what are the qualitative features of a "good" seed word or good set of seed words. For this purpose, we used a supervised feature selection method in the form of a wrapper and evaluated the automatically selected seed word sets against a range of potentially relevant factors: (i) number of possible part-of-speech tags, (ii) number of possible senses, (iii) seed word frequency of occurrence, (iv) mean and standard deviation of the semantic similarities, (v) standard deviation of valence, (vi) absolute value of valence and, (vii) the valence rating of the seed word. The number of part-of-speech tags and word senses was estimated from WordNet. The mean and standard deviation of semantic similarity scores were estimated between each seed word and all words in the ANEW dataset using the G or S^1 semantic similarity metric. The valence, absolute valence and standard deviation (where standard deviation was computed over all human annotations of each word) of valence were taken from the ANEW dataset.

The experiment was conducted as a modification on the ANEW-CV experiment. For each of the 10 folds, we performed an internal 10-fold cross-validation experiment (splitting the train set only into 10-folds, an approach referred to as double-loop cross-validation) and used the performance of different seed sets in the internal loop to select a seed set for that fold of the external loop. The seed set search strategy was forward, best-first: starting from an empty set we generated bigger sets by adding one feature at a time (the one that improves previous performance most) and there were no



(a) (b) (c) (d)

Fig. 5. Performance of the affective lexicon creation algorithm using different seed selection algorithms and analysis of the wrapper selected seeds for the ANEW-CV experiment using: (a) the G similarity metric, (b) the S^1 similarity metric. The corresponding rank distributions of the top 50 seeds per fold selected by a wrapper when using: (c) the G similarity metric, (d) the S^1 similarity metric.

substitutions or deletions. The criterion of seed selection was the mean square error in the internal experiment. We ran the process up to $N = 150$ seeds, creating 10 ordered seed sets of length 150, one for each fold of the external loop and evaluated the final performance on the external loop experiment.

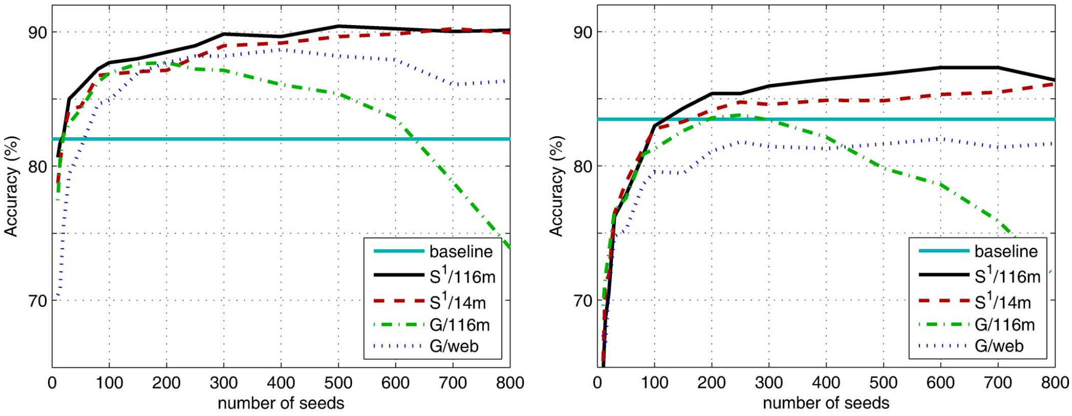
Correlation performance on the ANEW dataset is shown in Fig. 5(a) and (b) when using the G and S^1 similarity metrics, respectively, over the 116 m corpus. As a comparison, we provide the performance attained when using our unsupervised selection method based on absolute valence and seed set balance, as detailed in Section IV-C. There is a clear benefit to using a wrapper: performance is significantly ($p < 10^{-16}$ under 50 seeds) better when using a small number of seed words and the model reaches optimal performance requiring fewer seed words. However, the performance benefit dissipates fast (at 150–200 seed words) and, while a wrapper will reach optimal performance faster, that optimal performance is not significantly higher than that achieved by a model using our unsupervised selection method, especially for the G metric.

To identify features that make a good seed, we looked at the rank distributions of the selected seed words across the various factors, shown in Fig. 5(c) and (d) when using the G and S^1 similarity metrics, respectively. To make the results clearer we used only the top 50 seeds selected for each fold, for a total of 500 samples. Box plots range from the 25% to the 75% percentile of each distribution, while the dot in the box indicates the distribution median. In both cases, valence is the most relevant factor that defines a good set of seed words: the selected seed words have very high absolute valence ratings, a very narrow range of possible affective interpretations (low standard deviation of valence). Also the seed sets are close to balanced (high absolute valence and high set valence variance).

For all the experiments that follow we use the unsupervised seed selection method, since: i) the absolute valence heuristic is validated by the results in Fig. 5(c), (d), and ii) the performance gap between the unsupervised selection method and the double loop cross-validation method is small when over 100 seeds are used. However, note that if using a very small set of seed words is a priority a supervised seed selection algorithm can achieve good performance with a very small number of 20–40 seeds. Note also that correlation performance of the supervised seed selection algorithm is significantly ($p < 10^{-8}$) higher than the baseline method of [20] (solid blue line in Fig. 5(a),(b)) for the same number of (14) seeds.

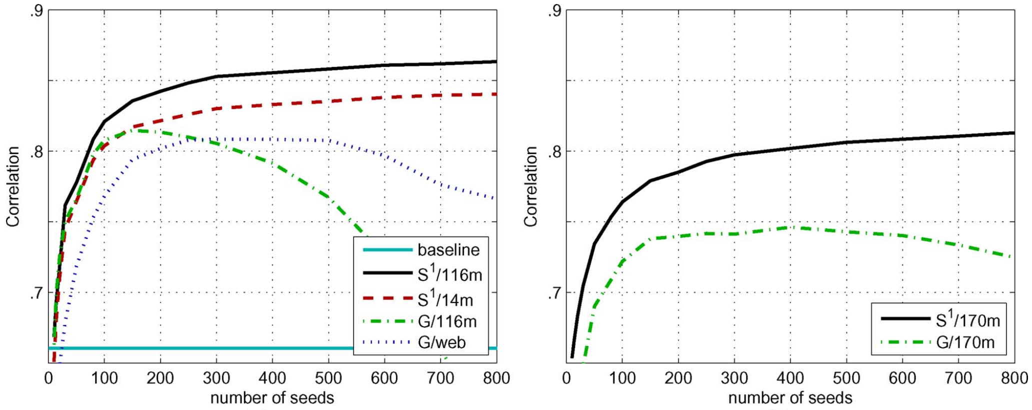
D. Word Affective Ratings

In this section, we report results using an unsupervised seed selection method, the 116 m corpus and the best performing similarity metrics G, S^1 with a linear kernel to evaluate the overall performance of the method on a variety of word level affective tasks. In Fig. 6, 2-class classification accuracy is shown for the binary word polarity detection ANEW-CV (a) and GINQ-PD (b) tasks. In Fig. 7, correlation performance is shown for the continuous polarity rating estimation ANEW-CV (a) and BAWLR-CV (b) tasks. Results are shown for the similarity metrics G (estimated on web and 116 m corpus) and S^1 (estimated on the 14 m and 116 m corpus), as well as the baseline performance of the method described in [20]. For the German task (BAWLR-CV), results for G, S^1 metrics estimated on the 170 m corpus are reported. For the ANEW-CV experiment, correlation with ground truth up to 0.87 and binary classification accuracy up to 91% is achieved using the context based similarity S^1 estimated on the large 116 K corpus. For the GINQ-PD experiment, the best performance achieved is classification accuracy of 87.3% for the S^1 metric estimated on the 116 K corpus. Comparable results for this experiment



(a) (b)

Fig. 6. Accuracy of the affective lexicon creation algorithm: (a) ANEW-CV experiment, (b) GINQ-PD experiment.



(a) (b)

Fig. 7. Correlation of the affective lexicon creation algorithm: (a) ANEW-CV experiment, (b) BAWLR-CV experiment.

found in literature, include: 82.8% [20], 81.9% [47] and 82.1% [24]). In the BAWLR-CV experiment the model reaches 0.82 correlation with the ground truth.

Of note is the good performance and robustness of context-based metrics across all experiments; they perform clearly better and provide a model that is stable to the seed selection process. In fact, the model continues to improve (in performance) even when adding sub-optimal seeds, not exhibiting the large performance drop of co-occurrence based metric models for large number of seeds. Also important is the ability of the method to perform well when applied to a different language (German) for the BAWLR-CV experiment. Though in absolute terms performance in BALWR-CV is lower than the comparable English experiment of ANEW-CV, performance is still good, particularly considering that the proposed model and similarity estimation process is language-agnostic[14].

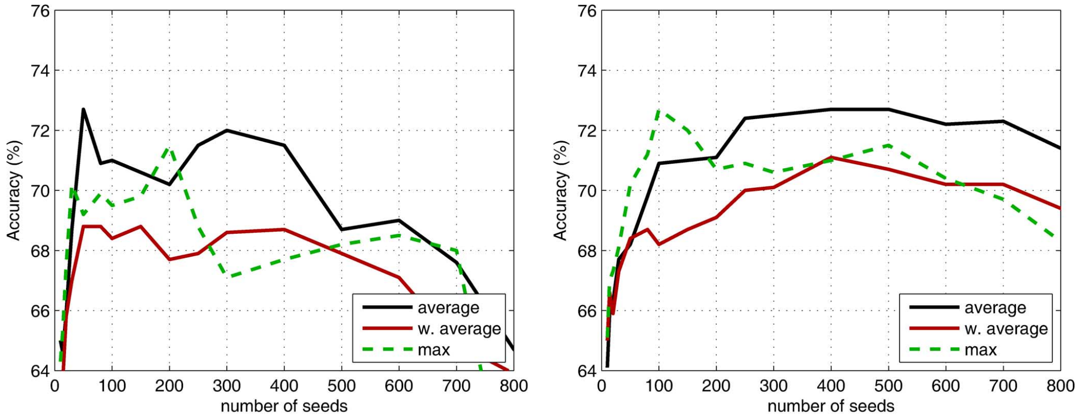
E. Sentence Affective Ratings

For the sentence level affective rating task, we started from (a subset of) the seed words of the ANEW dataset and performed lexicon expansion for all unigrams and bigrams in our sentence corpus. Specifically, G and S^1 semantic similarities were estimated between all unigrams and bigrams in the sentence corpus, and the ANEW seed words. Similarity metrics were estimated on the 116 m corpus. The affective model was then used to create ratings for all unigram and bigrams in the sentence corpus. The affective ratings were then combined using one of the (linear, weighted, max) fusion methods described in Section III-C. Unigram and bigram affective ratings were fused using one of the methods defined in Section III-D, i.e., interpolation, back-off, weighted interpolation. The Least Squares Estimation (LSE) algorithm was used to estimate the unigram and bigram weights[15] on held-out data (SemEval development set). The various sentence level affective models were then evaluated on the sentence corpus (SemEval test set).

1) Baseline Performance: In order to establish a baseline, we used the fusion schemes defined in Section III-C, using only unigram terms. Sentence level classification accuracy as a function of the number of seeds is shown in Fig. 8 for the G and S^1 metrics estimated on the 116 m corpus. Performance peaks at about 72% for the G metric and 72.5% for the S^1 metric, an improvement over previous results reported in [16]. The improvement of the word rating algorithm and the addition of supervised training to the sentence model provide a fairly minimal improvement in performance. The simple numeric average performs better throughout our experiments and benefits, particularly in terms

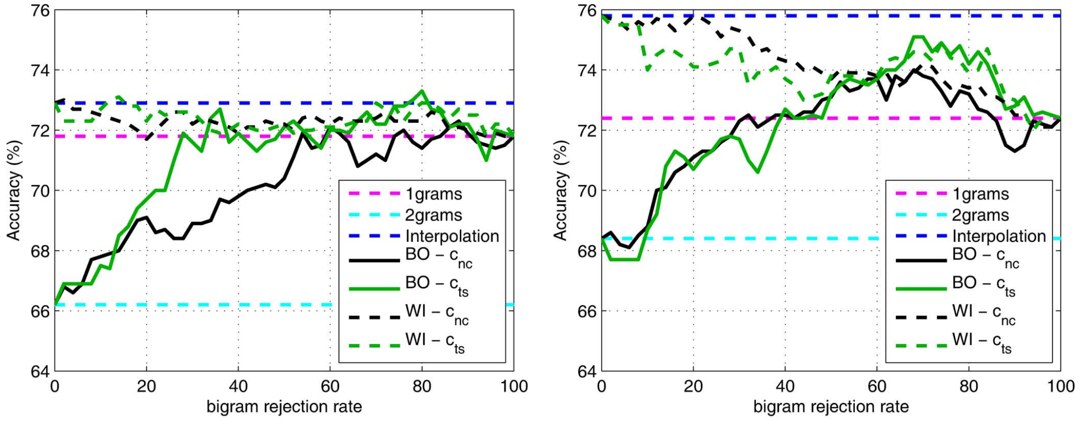
[14] Although, we have not yet performed a detailed evaluation of semantic similarity metrics for the German language, preliminary experiments indicate that context based semantic similarity metrics perform worse for morphologically rich languages probably due to the larger number of word forms in these languages.

[15] Note that only the n-gram fusion weights were estimated on the sentence development dataset. The affective model seed weights were estimated on the ANEW dataset.



(a) (b)

Fig. 8. Binary classification accuracy of the sentence rating algorithm as a function of the number of seed words, when using only unigram terms and: (a) the G similarity metric, (b) the S^1 similarity metric.



(a) (b)

Fig. 9. Binary classification accuracy of the sentence rating algorithm as a function of the bigram selection threshold (backoff rate) for the SemEval'07-Task14 dataset: (a) the G similarity metric and 300 seeds, (b) the S^1 similarity metric and 600 seeds.

of stability, from the supervised training. Sentence ratings exhibit very similar performance dynamics to word-level ratings, e.g., optimal performance occurs for a similar number of seed words.

2) Fusion of n*-Gram Models:* Creating sentence ratings using the higher order models described in Section III-D poses a specific challenge: we need to select which terms to use, from a pool of unigrams and bigrams. To do so, we used the criteria described in Section III-D, i.e., interpolation, back-off and weighted interpolation. To investigate the performance for various combinations of unigrams and bigrams, we selected two specific word models (using the G similarity metric and 300 seeds, and using the S^1 similarity metric and 600 seeds) and used different term selection criteria during the sentence rating creation process. Sentence level classification accuracy as a function of bigram rejection rate (back-off rate) is shown in Fig. 9 for the G (a) and S^1 (b) metrics. The figures can be read as follows: we start at the bottom left, with the models using only bigram terms. Then we move to the right, by replacing bigrams with unigrams (back-off) according to the selection criterion, until we reach the right edge, where the models are using only unigram terms (baseline). From that point we move back towards the upper-left corner, by keeping all unigram terms and adding increasingly more bigrams (weighted interpolation), again based on the selection criterion, until we reach the left edge, where the model is using all unigrams and bigrams (interpolation).

Performance when using only bigrams (dotted cyan line) is noticeably lower than when using only unigrams (dotted purple line), probably due to the lack of bigram seeds to bootstrap the affective model. Despite this shortcoming, combining bigram with unigrams significantly ($p < 10^{-3}$ at 80% bigram rejection in both cases) improves the performance of the affective model over the unigram baseline. The performance gain is noticeably higher when using the context[16] based semantic similarity S^1, as shown in Fig. 9(b).

In Fig. 9, classification accuracy is shown only for the two best term selection criteria: c_{ts} and c_{nc} (to improve readability). The two criteria detect terms in very different ways, c_{ts} is often used for term-extraction or compound detection (i.e., non-compositional semantic constructs), while c_{nc} estimates the degree of affective non-compositionality. The semantic criterion provides the absolute best performance when using a back-off model, while there is no clear winner when using the weighted interpolation model. Focusing on the back-off model performance for S^1 in Fig. 9(b), we observe a significant ($p < 10^{-6}$ at 70% bigram rejection) improvement over the unigram baseline: accuracy improves from 72.4% when using only bigrams to around 75% at a back-off rate around 0.7. Weighted interpolation performs worse than the back-off

[16] The performance gap between G and S^1 is also large for the bigram only experiment, 66% vs. 68.5%. As mentioned in Section III-B, there are similarity metric scaling issues when creating bigram ratings that are more pronounced when using co-occurrence based similarities.

model, up until it converges to the interpolation model (bigram rejection rate under 10%). None of the proposed term selection criteria perform better than the (simple) interpolation baseline. Interpolation is the best performing model reaching an accuracy of 75.9%, a small improvement over the back-off model (at 75%).

Overall, the inclusion of bigrams leads to significantly improved performance over the unigram only models, with accuracy reaching 75.9%. Comparable results in literature are 62% [48], 66% [49], 71% [14] and 72.8% (using cross-validation) [50]. The interpolation model also achieved a correlation to the ground truth of 0.61, compared to 0.5 achieved by the best system in [9].

VI. Conclusions

We proposed a method of creating sentence affective ratings based on the combination of partial affective ratings of word n-grams. At the core of this method is an affective lexicon expansion algorithm capable of creating continuous n-gram affective ratings based on a set of manually labeled seed words and semantic similarity ratings calculated over web data. This algorithm achieves state-of-the-art results in lexical affective tasks and is generic enough to work in languages other than English, achieving high performance in creating ratings for German words. Most importantly it does not require any linguistic resources other than the affective ratings of a few hundred words in each language. Sentence level ratings were obtained from n-gram ratings using linear and non-linear fusion methods. Interpolation and back-off models were proposed for combining unigram and bigram affective ratings. Overall, a simple linear equation containing the weighted ratings of all terms, both unigram and bigram, proved to be the best performing solution achieving state-of-the-art performance in the SemEval'07-Task14.

Future work should include further refinement of the lexicon creation model specifically targeted at the creation of more accurate higher-order n-gram ratings. Incorporating morphosyntactic information into the model is also important especially for morphologically rich languages. The current word/sentence models can be used to create chunk ratings, e.g., for noun phrases or compound nouns, reducing the required complexity of syntactic rules; a simple syntactic model can then be used to model non-linear interaction between these chunks.

References

[1] L. Lloyd, D. Kechagias, and S. Skiena, "Lydia: A system for large-scale news analysis," in *Proc. SPIRE*, 2005, vol. 3772, Lecture Notes in Computer Science, pp. 161–166.

[2] K. Balog, G. Mishne, and M. de Rijke, "Why are they excited? Identifying and explaining spikes in blog mood levels," in *Proc. EACL*, 2006, pp. 207–210.

[3] J. Wiebe and R. Mihalcea, "Word sense and subjectivity," in *Proc. COLING/ACL*, 2006, pp. 1065–1072.

[4] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proc. SIGKDD*, 2004, KDD '04, pp. 168–177.

[5] C. M. Lee and S. Narayanan, "Toward detecting emotions in spoken dialogs," *IEEE Trans. Speech Audio Process.*, vol. 13, no. 2, pp. 293–303, May 2005.

[6] C. M. Lee, S. Narayanan, and R. Pieraccini, "Combining acoustic and language information for emotion recognition," in *Proc. ICSLP*, 2002, pp. 873–876.

[7] J. Ang, R. Dhillon, A. Krupski, E. Shriberg, and A. Stolcke, "Prosody-based automatic detection of annoyance and frustration in human-computer dialog," in *Proc. ICSLP*, 2002, pp. 2037–2040.

[8] A. Purandare and D. J. Litman, "Humor: Prosody analysis and automatic recognition for f*r*i*e*n*d*s*," in *Proc. EMNLP*, 2006, pp. 208–215.

[9] C. Strapparava and R. Mihalcea, "Semeval-2007 task 14: Affective text," in *Proc. SemEval*, 2007, pp. 70–74.

[10] P. Stone, D. Dunphy, M. Smith, and D. Ogilvie, *The General Inquirer: A Computer Approach to Content Analysis*. Cambridge, MA, USA: MIT Press, 1966.

[11] M. Bradley and P. Lang, "Affective norms for English words (ANEW): Stimuli, instruction manual and affective ratings Center for Research in Psychophysiol. Univ. of Florida, , 1999, Tech. Rep. C-1.

[12] A. Esuli and F. Sebastiani, "Sentiwordnet: A publicly available lexical resource for opinion mining," in *Proc. LREC*, 2006, pp. 417–422.

[13] C. Strapparava and A. Valitutti, "WordNet-Affect: An affective extension of WordNet," in *Proc. LREC*, 2004, vol. 4, pp. 1083–1086.

[14] K. Moilanen, S. Pulman, and Y. Zhang, "Packed feelings and ordered sentiments: Sentiment parsing with quasi-compositional polarity sequencing and compression," in *Proc. WASSA Workshop ECAI*, 2010, pp. 36–43.

[15] I. Sag, T. Baldwin, F. Bond, A. Copestake, and D. Flickinger, "Multiword expressions: A pain in the neck for NLP," in *Comput. Linguist. Intell. Text Process.*, 2002, vol. 2276, Lecture Notes in Computer Science, pp. 189–206.

[16] N. Malandrakis, A. Potamianos, E. Iosif, and S. Narayanan, "Kernel models for affective lexicon creation," in *Proc. Interspeech*, 2011, pp. 2977–2980.

[17] V. Hatzivassiloglou and K. McKeown, "Predicting the semantic orientation of adjectives," in *Proc. ACL*, 1997, pp. 174–181.

[18] P. Turney and M. L. Littman, Unsupervised learning of semantic orientation from a hundred-billion-word corpus National Research Council of Canada, 2002, Tech. Rep. ERC-1094 (NRC 44929).

[19] C. Strapparava, A. Valitutti, and O. Stock, "The affective weight of lexicon," in *Proc. LREC*, 2006, pp. 423–426.

[20] P. Turney and M. L. Littman, "Measuring praise and criticism: Inference of semantic orientation from association," *ACM Trans. Inf. Syst.*, vol. 21, pp. 315–346, 2003.

[21] M. Taboada, C. Anthony, and K. Voll, "Methods for creating semantic orientation dictionaries," in *Proc. LREC*, 2006, pp. 427–432.

[22] A. Esuli and F. Sebastiani, "Determining the semantic orientation of terms through gloss classification," in *Proc. CIKM*, 2005, pp. 617–624.

[23] A. Andreevskaia and S. Bergler, "Semantic tag extraction from WordNet glosses," in *Proc. LREC*, 2006, pp. 413–416.

[24] A. Hassan and D. Radev, "Identifying text polarity using random walks," in *Proc. ACL*, 2010, pp. 395–403.

[25] A. Budanitsky and G. Hirst, "Evaluating WordNet-based measures of semantic distance," *Comput. Linguist.*, vol. 32, pp. 13–47, 2006.

[26] A. Pargellis, E. Fosler-Lussier, C.-H. Lee, A. Potamianos, and A. Tsai, "Auto-induced semantic classes," *Speech Commun.*, vol. 43, pp. 183–203, 2004.

[27] D. Bollegala, Y. Matsuo, and M. Ishizuka, "Measuring semantic similarity between words using web search engines," in *Proc. Int. Conf. World Wide Web*, 2007, pp. 757–766.

[28] M. Baroni and A. Lenci, "Distributional memory: A general framework for corpus-based semantics," *Comput. Linguist.*, vol. 36, no. 4, pp. 673–721, 2010.

[29] E. Agirre, E. Alfonseca, K. Hall, J. Kravalova, M. Pasca, and A. Soroa, "A study on similarity and relatedness using distributional and Wordnet-based approaches," in *Proc. NAACL*, 2009, pp. 19–27.

[30] E. Iosif and A. Potamianos, "Unsupervised semantic similarity computation between terms using web documents," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 11, pp. 1637–1647, Nov. 2010.

[31] S. Pado and M. Lapata, "Dependency-based construction of semantic space models," *Comput. Linguist.*, vol. 33, no. 2, pp. 161–199, 2007.

[32] E. Iosif and A. Potamianos, "Similarity computation using semantic networks created from Web-harvested data," *Nat. Lang. Eng.*, no. 8, pp. 1–31, 2013, submitted for publication.

[33] F.-R. Chaumartin, "UPAR7: A knowledge-based system for headline sentiment tagging," in *Proc. SemEval*, 2007, pp. 422–425.

[34] A. Andreevskaia and S. Bergler, "CLaC and CLaC-NB: Knowledge-based and corpus-based approaches to sentiment tagging," in *Proc. SemEval*, 2007, pp. 117–120.

[35] L. Polanyi and A. Zaenen, "Contextual valence shifters," in *Computing Attitude and Affect in Text: Theory and Applications*. New York, NY, USA: Springer-Verlag, 2006, pp. 1–10.

[36] P. M. Vitáanyi, "Universal similarity," in *Proc. Inf. Theory Workshop Coding Complex.*, 2005, pp. 238–243.

[37] R. L. Cilibrasi and P. M. Vitáanyi, "The Google similarity distance," *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 3, pp. 370–383, Mar. 2007.

[38] J. Gracia, R. Trillo, M. Espinoza, and E. Mena, "Querying the web: A multiontology disambiguation method," in *Proc. Int. Conf. Web Eng.*, 2006, pp. 241–248.

[39] H. Rubenstein and J. B. Goodenough, “Contextual correlates of synonymy,” *Commun. ACM*, vol. 8, no. 10, pp. 627–633, 1965.

[40] E. Iosif, A. Tegos, A. Pangos, E. Fosler-Lussier, and A. Potamianos, “Combining statistical similarity measures for automatic induction of semantic classes,” in *Proc. IEEE/ACL Workshop Spoken Lang. Technol.*, 2006, pp. 86–89.

[41] B. Schuller and T. Knaup, “Learning and knowledge-based sentiment analysis in movie review key excerpts,” in *Toward Autonomous, Adaptive, and Context-Aware Multimodal Interfaces. Theoretical and Practical Issues*, 2011, vol. 6456, Lecture Notes in Computer Science, pp. 448–472.

[42] F. J. Pelletier, “The principle of semantic compositionality,” *Topoi*, vol. 13, pp. 11–24, 1994.

[43] N. Malandrakis, A. Potamianos, and S. Narayanan, *Continuous Models of Affect from Text Using* n*-Grams*, 2013, to be published.

[44] “Apache Lucene,” [Online]. Available: http://www.lucene.apache.org/

[45] “Gnu Aspell,” [Online]. Available: http://www.aspell.net

[46] H. Schmid, “Probabilistic part-of-speech tagging using decision trees,” in *Proc. Int. Conf. New Methods in Lang. Process.*, 1994, vol. 12, pp. 44–49.

[47] H. Takamura, T. Inui, and M. Okumura, “Extracting semantic orientations of words using spin model,” in *Proc. ACL*, 2005, pp. 133–140.

[48] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede, “Lexicon-based methods for sentiment analysis,” *Comput. Linguist.*, vol. 1, pp. 1–41, 2010.

[49] K. Moilanen and S. Pulman, “Sentiment composition,” in *Proc. RANLP*, 2007, pp. 378–382.

[50] J. Carrillo de Albornoz, L. Plaza, and P. Gerváas, “A hybrid approach to emotional sentence polarity and intensity classification,” in *Proc. CoNLL*, 2010, pp. 153–161.

Nikolaos Malandrakis (StM’12) received the Diploma and the MS degrees from the Dept. of ECE, Tech. Univ. of Crete, in 2007 and 2012 respectively. Since 2012, he is a research assistant and a Ph.D. candidate at the Dept. of EE, Univ. of Southern California. His current research interests include natural language processing, spoken dialogue systems and emotion detection.

Alexandros Potamianos (M’92, SM’10) received the Dipl. in Electrical & Computer Engineering from the Natl. Tech. Univ. of Athens, Greece in 1990. He received the MS and Ph.D. degrees in Engineering Sciences from Harvard University, USA in 1991 and 1995, respectively. From 1995 to 1999 he was a Senior Technical Staff Member at the Speech and Image Processing Lab, AT&T Shannon Labs, Florham Park, NJ. From 1999 to 2002 he was a Technical Staff Member and Technical Supervisor at the Multimedia Communications Lab at Bell Labs, Lucent Technologies, Murray Hill, NJ. From 1999 to 2001 he was an adjunct Assistant Professor at the Dept. of EE, Columbia University, NY. In 2003, he joined the ECE Dept., Tech. Univ. of Crete, Greece as an associate professor. His current research interests include speech processing, analysis, synthesis and recognition, dialog and multimodal systems, nonlinear signal processing, natural language understanding, data mining, artificial intelligence and multimodal child-computer interaction. Prof. Potamianos has authored or co-authored over 100 papers in professional journals and conferences, and hold four U.S. patents. He is the co-author of the paper “Creating conversational interfaces for children” that received a 2005 IEEE Signal Processing Society Best Paper Award. He is a member of the IEEE Signal Processing Society since 1992 and has served terms with the IEEE SLT and MM committees.

Elias Iosif (StM’08) received the Diploma and the MS degrees from the ECE Dept., Tech. Univ. of Crete, in 2005 and 2007 respectively. Since 2007, he is a research assistant and a Ph.D. candidate at the ECE Dept. Tech. Univ. of Crete. His current research interests include natural language processing and semantic web. He is a member of the Cyprus Scientific and Technical Chamber since 2005.

Shrikanth (Shri) Narayanan (StM’88-M’95-SM’02-F’09) is Andrew J. Viterbi Professor of Engineering at the University of Southern California (USC), and holds appointments as Professor of Electrical Engineering, Computer Science, Linguistics and Psychology and as the founding director of the Ming Hsieh Institute. Prior to USC he was with AT&T Bell Labs and AT&T Research from 1995–2000. At USC he directs the Signal Analysis and Interpretation Laboratory (SAIL). His research focuses on human-centered information processing and communication technologies with a special emphasis on behavioral signal processing and informatics. [http://sail.usc.edu]

Prof. Narayanan is a Fellow of the Acoustical Society of America and the American Association for the Advancement of Science (AAAS) and a member of Tau Beta Pi, Phi Kappa Phi, and Eta Kappa Nu. He is also an Editor for the Computer Speech and Language Journal and an Associate Editor for the IEEE TRANSACTIONS ON AFFECTIVE COMPUTING, APSIPA Transactions on Signal and Information Processing and the Journal of the Acoustical Society of America. He was also previously an Associate Editor of the IEEE TRANSACTIONS OF SPEECH AND AUDIO PROCESSING (2000–2004), IEEE Signal Processing Magazine (2005–2008) and the IEEE TRANSACTIONS ON MULTIMEDIA. He is a recipient of a number of honors including Best Paper awards from the IEEE Signal Processing Society in 2005 (with A. Potamianos) and in 2009 (with C. M. Lee) and selection as an IEEE Signal Processing Society Distinguished Lecturer for 2010–2011. Papers co-authored with his students have won awards at Interspeech 2012 Speaker Trait Challenge, Interspeech 2011 Speaker State Challenge, InterSpeech 2010, InterSpeech 2009-Emotion Challenge, IEEE DCOSS 2009, IEEE MMSP 2007, IEEE MMSP 2006, ICASSP 2005 and ICSLP 2002. He has published over 500 papers and has fourteen granted U.S. patents.