Unsupervised Emotion Detection from Text using Semantic and Syntactic Relations

Ameeta Agrawal
Department of Computer Science and Engineering
York University, Toronto, Canada
ameeta@cse.yorku.ca

Aijun An
Department of Computer Science and Engineering
York University, Toronto, Canada
aan@cse.yorku.ca

***Abstract*—Emotion detection from text is a relatively new classification task. This paper proposes a novel unsupervised context-based approach to detecting emotion from text at the sentence level. The proposed methodology does not depend on any existing manually crafted affect lexicons such as WordNet-Affect, thereby rendering our model flexible enough to classify sentences beyond Ekman's model of six basic emotions. Our method computes an emotion vector for each potential affect-bearing word based on the semantic relatedness between words and various emotion concepts. The scores are then fine tuned using the syntactic dependencies within the sentence structure. Extensive evaluation on various data sets shows that our framework is a more generic and practical solution to the emotion classification problem and yields significantly more accurate results than recent unsupervised approaches.**

***Keywords*-affective computing, emotion detection;**

I. Introduction

An emotion is a particular feeling that characterizes a state of mind, such as joy, anger, love, fear and so on. Automatic emotion detection from text has attracted growing attention due to its potentially useful applications. For examples, psychologists can better assist their patients by analyzing their session transcripts for any subtle emotions; reliable emotion detection can help develop powerful human-computer interaction devices; and deep emotional analysis of public data such as tweets and blogs could reveal interesting insights into human nature and behavior.

Many current approaches to emotion detection are based on supervised learning methods, in which a large set of annotated data (where text has been labeled with emotions) is needed to train the model. Although the supervised learning based methods can achieve good results, the availability of large annotated data sets is very low and a model trained on one domain does not translate well to another.

There are some methods that do not use supervised learning. However, most of these methods use manually designed dictionaries of emotion keywords. A problem with such an affect lexicon-based method is that the number of emotion categories is fixed and limited in the dictionary. Another problem is that if a sentence expresses emotion using words that do not appear in the dictionary, then it would be considered to be unemotional. For example, the sentence "Izzy got lots of new toys for her first birthday" conveys quite a happy feeling despite not containing any obvious happy keywords such as joy, glad, etc. Affect lexicon-dependent techniques may fail to detect emotions from such sentences.

There are also methods that rely on linguistic rules, but designing such rules is not a trivial task. Moreover, most of these rules have not been made publicly available. In addition, most current emotion detection methods look at individual words without considering the context a word is in. However, a word can invoke different emotions in different contexts.

We propose a novel unsupervised context-based emotion detection method that does not rely on any affect dictionaries or annotated training data. Therefore, the approach is not restricted to a fixed number of emotion categories. We start with a small set of representative words which are used to compute an emotion vector of an affect bearing word by calculating the semantic relatedness score between this word and an emotion concept. To fine tune the emotion vectors, the context of the word is considered using three types of syntactic dependencies. Extensive evaluation of our framework shows promising results.

The rest of the paper is organized as follows. In the next section we present a literature survey of textual emotion detection. Section 3 describes the details of our proposed algorithm. An extensive set of experiments that evaluate the performance of our approach is presented in Section 4. Finally, we conclude the paper and discuss some future avenues of research work.

II. Related Work

This section outlines some lexical resources that researchers have compiled over the years to support affective computing and a variety of recently proposed methodologies.

A. Lexical Resources

One of the first such resources was a list of 1,336 adjectives manually labeled [1]. WordNet-Affect was introduced as a hierarchy of affective domain labels [2]. The subjectivity lexicon developed by [3] is comprised of over 8,000 words. Motivated by the assumption that different senses of the same term may have different opinion-related properties, [4] developed SentiWordNet, a lexicon based on WordNet.

978-0-7695-4880-7/12 $26.00 © 2012 IEEE
DOI 10.1109/WI-IAT.2012.170

An automatically generated lexicon called SentiFul database was introduced in [5].

B. Emotion Detection Approaches

Emotion recognition approaches can be broadly classified into keyword-based, linguistic rules-based and machine learning techniques. We further distinguish them based on whether they employ any affect lexicons.

1) Keyword-based Approaches using Affect Lexicons: Keyword-based approaches are applied at the basic word level [6]. Such a simple model cannot cope with cases where affect is expressed by interrelated words.

2) Linguistic Rules-based Approaches: Computational linguists use various rules to define a language structure.

- Rule-based approaches with affect lexicons: The ESNA system [7] was developed to classify emotions in news headlines. Chaumartin [8] manually added seed words to emotion lists and created a few rules in their system UPAR7, which identifies what is being said about the main subject and boosts its emotion rating by exploiting dependency graphs. The effect of conjuncts [9] was studied using rules over syntax trees and lexical resources such as General Inquirer and WordNet. One of the most recent rule-based approaches [10] can recognize nine emotions. Most of these approaches do an excellent task of defining rules that decipher complex language structures. However, designing and modifying rules is far from a trivial task. Similarly, approaches using affect lexicons suffer from the inflexibility of catering to emotions other than those already listed.
- Rule-based approaches without affect lexicons: As an alternative to using affect lexicons, [11] proposed an approach for understanding the underlying semantics of language. Another interesting approach is to recognize emotions from text rich in metaphorical data [12]. Although such methods have the flexibility of using any set of emotions and are thus more practical, the rules are specific to the representation of the source from which knowledge is extracted.

3) Machine Learning Approaches: To overcome the limitations faced by rule-based methods, researchers devised some statistical machine learning techniques which can be subdivided into supervised and unsupervised techniques.

- Supervised machine learning with affect lexicons: One of the earliest supervised machine learning methods was employed by Alm [13], where they used a hierarchical sequential model along with SentiWordNet list for fine-grained emotion classification. Blog sentences have been classified using Support Vector Machines (SVM) in [14]. Although supervised learning performs well, it has the distinct disadvantage that large annotated data sets are required for training the classifiers and classifiers trained on one domain generally do not perform so well on another.
- Supervised machine learning without affect lexicons: A comparison among three machine learning algorithms on a movie review data set concluded that SVM performs the best [15]. The same problem was also attempted using the delta *tf-idf* function in [16].
- Unsupervised machine learning with affect lexicons: An evaluation of two unsupervised techniques using WordNet-Affect exploited a vector space model and a number of dimensionality reduction methods [17]. News headlines have been classified using simple heuristics and more refined algorithms (e.g., similarity in a latent semantic space) [18].
- Unsupervised machine learning without affect lexicons: Some inspiring work done in this area includes 'LSA single word' which measures similarity between text and each emotion and the 'LSA emotion synset' approach which uses WordNet synsets [18]. Our approach shares a similar intuition as that of the 'LSA emotion synset' method, albeit with some notable differences as we use Pointwise Mutual Information (PMI) to compute the semantic relatedness which is further enriched using context-dependency rules. Although [19] use PMI as well to gather statistics from three search engines, they compare an entire phrase to just one emotion word due to long online processing times, whereas we compare each relevant word to a set of representative words for each emotion and take into account its context.

III. Methodology

Emotion detection is modeled as a classification problem where one or more nominal labels are assigned to a sentence from a pool of target emotion labels.

A. Overview of the Framework

Let s be a sentence and ω_s an emotion label. Let e be a set of m possible emotion categories (excluding neutral) where $e = \{e_1, e_2, \ldots, e_m\}$. The objective is to label s with the best possible emotion label ω_s, where $\omega_s \in \{e_1, e_2, \ldots, e_m, \text{neutral}\}$.

Our emotion recognition framework, shown in Fig. 1, includes four main components: preprocessing, semantic, syntactic and sentence analysis. The preprocessing task consists of sentence parsing, parts-of-speech tagging and syntactic dependency parsing. This enables us to extract the relevant affect-bearing words and the syntactic dependencies between them. The next module performs word-level analysis by computing an emotion vector for the affect-bearing words by calculating their semantic relatedness to emotion concepts. Then, the syntactic module performs phrase-level analysis by using context information to adjust the emotion vectors computed in the previous step. Finally, the sentence analysis module aggregates the emotion vectors of all the relevant words to compute the emotion label of the sentence.

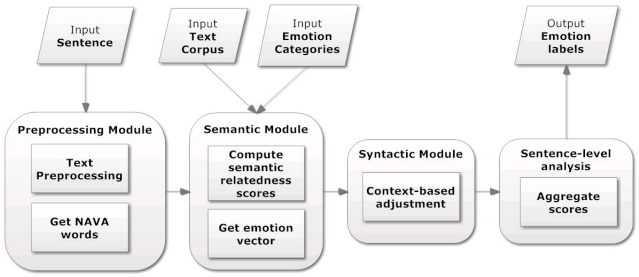

Figure 1. Overview of the emotion detection framework.

B. Extracting Affect Words

Some words express affect more apparently than others. But consider the sentence, "Izzy got lots of new toys for her birthday", again. Here, the words 'new', 'toys' and 'birthday' together convey happiness although it may not seem so when these words are looked at individually. Let us call such affect-bearing words as NAVA (Noun Adjective Verb Adverb) words. We begin by tagging the input sentence and extracting a set of NAVA words from the sentence. For example, for sentence "It feels sad", the words 'feels' and 'sad', tagged as a verb and an adjective respectively, are extracted. Traditionally, these NAVA words are looked up in a lexical resource to gauge their emotional affinity and by combining the emotions of all the words, the overall sentence emotion is derived. But consider another sentence: "The performers were greeted with joyless cheer". The NAVA words extracted are 'performers', 'greeted', 'joyless' and 'cheer'. Lexical resource such as WordNet-Affect understandably lists 'cheer' under the emotion category *joy*. However, in this particular sentence, the emotion tendency of 'cheer' is being influenced by the word 'joyless' turning the emotion value of the phrase 'joyless cheer' to resemble more like *sadness* than *joy*. This is essentially a case of a word conveying different emotions depending on the context it is used in. Conventionally, a keyword-based approach, would consider the words 'joyless' and 'cheer' to be *sad* and *happy* respectively and cancel out their effect, resulting in a possibly neutral sentence. But by using context, 'joyless' can influence the emotion vector of 'cheer', thus resulting in the label to be *sad*. To explore this idea of influencing and dependent words in a sentence, and adjust the emotional vectors accordingly, below we propose to exploit the syntactic dependencies to capture some of the context.

C. Considering Context using Syntactic Dependencies

Words are embedded in a larger structure such as a sentence and it seems natural to use surrounding emotion expressions of words to help inform the classification process, modeling a phenomenon that extends beyond the current bag-of-words approach.

A syntactic dependency is represented as $d\left(w1^{\downarrow}, w2^{\uparrow}\right)$ [20], where the predicate d denotes a syntactic grammatical relation such as nominal subject, negation, adjectival modifier and so on. The arrows $\downarrow$ and $\uparrow$ represent the *modified* and *modifier* positions respectively and $w1$ is the dependent word while $w2$ is the influencing word. We focus on three types of typed dependencies, namely, *adjectival complement*, *adjectival modifier* and *negation modifier*. An adjectival complement of a verb is an adjectival phrase which functions like an object of the verb. For example, the adjectival complement dependency from "She looks very beautiful" is *acomp(looks, beautiful),* where 'beautiful' is the adjectival complement of the verb 'looks'. Mapping this to the binary dependency notation, 'looks' becomes the dependent word and 'beautiful', the influencer. An adjectival modifier is any adjectival phrase that modifies the meaning of the noun phrase. For example, the adjectival modifier in "Sam eats red meat" is *amod(meat, red)*, where 'meat' is dependent, and 'red', the influencing word. A negation modifier is the relation between a negation word and the word it modifies. For example, in "She is not happy today", the adverb 'not' is the influencing word, whereas 'happy' the dependent. Currently we use only these three dependencies as both the words in these relationships belong to the NAVA word set which depict interesting inter-word relationships.

D. Representing Emotion as a Vector

Formally, the emotion of a NAVA word can be defined as a vector whose elements each represent the strength of the affinity of the word for an emotion category. For example, if Ekman's emotion model [21] is used, the emotion of a word is represented as a six-valued vector and each value corresponds to one of the six emotions: *happiness, sadness, anger, fear, surprise*, and *disgust*. Traditionally, the emotion vector is calculated by directly matching it against an affect dictionary. One of the shortcomings of this approach is that it cannot detect emotions if the sentence does not contain any obvious emotional keywords. Consider a sentence such as "That is nonsense", which clearly sends out an angry vibe but unless the word "nonsense" exists in the affect lexicon, it would be difficult for a system to identify the emotion of this sentence. To this effect, we propose to use semantic relatedness to compute the emotion vector.

E. Semantic Relatedness between Two Words

Adjectives with same polarity tend to appear together [1]. We propose to extend this idea further to assume that the affect words (adjectives, nouns, verbs and adverbs) that frequently co-occur together have the same emotional tendency. If two words co-occur more frequently, they tend to be semantically related. There are various models for measuring semantic relatedness and although they use different algorithms, they are all fundamentally based on the principle that a word's meaning can be induced by observing its statistical usage across a large sample of language.

Pointwise Mutual Information (PMI) is a simple information-theoretic measure of semantic relatedness that measures the similarity between two terms by using the probability of co-occurrence [22]. Mathematically, the PMI between two words x and y is calculated as follows:

$$\text{PMI}\,(x,y) = \frac{\text{co-occurrence}\,(x,y)}{\text{occurrence}\,(x)\,\text{occurrence}\,(y)} \qquad (1)$$

where occurrence (x) is the number of times that x appears in a corpus, and co-occurrence (x,y) is the number of times that x and y co-occur within a specified window[1] in the corpus. The corpus can be domain-dependent or general depending on the task at hand.

Being a measure of the degree of statistical dependence between two words, the purpose of PMI is to determine how closely two words are related. The motivation for using PMI instead of other measures of semantic relatedness stems from the statistical results found in the study [24] which found that PMI, which is a scalable and incremental method greatly benefits from training on large corpus of data and can outperform a commonly used version of LSA. For five out of six tests, the model built on Wikipedia using PMI was the second highest performing measure, outperformed solely by the model built using WordNet similarity vector measure. PMI was the highest performing measure on the remaining test. We choose to use PMI instead of WordNet similarity vector measure for one important distinction - WordNet measure is based on hand-coded intelligence and is limited to the words in the WordNet lexicon. Therefore, as impressive as WordNet's performance is, from a practical standpoint PMI provides a faster and more scalable measure.

F. Calculating Emotion Vector of a NAVA word

Let $w = \{w_1, w_2, \ldots, w_n\}$ be a set of n NAVA words of a sentence s, where $w \subset s$. Let $\alpha = \{\alpha_1, \alpha_2, \ldots, \alpha_c\}$ be a set of c influencing words in s. Let $\beta = \{\beta_1, \beta_2, \ldots, \beta_d\}$ be a set of d dependent words in s. Clearly, $\alpha \subset w$ and $\beta \subset w$. Let $e = \{e_1, e_2, \ldots, e_m\}$ be the set of m emotion concepts that a sentence can be classified into. For example, if we choose to classify a sentence using Ekman's model of six emotions, then e = {happiness, sadness, anger, fear, surprise, disgust}. The two subsections describe how to compute and adjust the emotion vectors.

1) Computing Emotion Vector of a Word without Context Information: A simple solution to derive the emotion vector σ_{w_i} for a NAVA word w_i is to use the PMI score between w_i and the word representing an emotion concept. However, since an emotion concept can often be expressed through various words (e.g. 'glad' or 'joy' for 'happiness'), we propose to use a few words [2] rather than just one generic word representing the entire emotion category. Table I shows examples of some representative words.

Table I
SAMPLE REPRESENTATIVE WORD SET

Emotion	Representative words
happiness	happy, glad, joy, good, love
sadness	sad, sorrow, hurt, cry, bad
anger	angry, irritate, stupid, annoy, frustrate
fear	fear, afraid, frighten, scare, terrify
surprise	surprise, amazing, astonish, incredible, wonder
disgust	disgust, dislike, hate, sick, ill

The idea is that if a word belongs to an emotion category, it will be closely related to most of the representative words that comprise the emotion concept instead of a random off-chance association with a single word. The representative words in Table I are the most commonly used synonyms taken from a generic thesaurus. Our experiments show that different common synonyms produced similar results. Thus, we will use these representative words in our method to compare with other emotion detection methods. The results using other representative words will not be included in this paper due to the space limitation.

The PMI scores between w_i and each representative word of an emotion category are used to compute the PMI score of w_i and the category. Let K_j be a set of r representative words for emotion concept e_j. The semantic relatedness score between an affect word w_i and an emotion category e_j is calculated as shown in (2),

$$\text{PMI}\,(w_i, e_j) = \sqrt[r]{\prod_{g=1}^{r} \text{PMI}\left(w_i, K_j^g\right)} \ . \qquad (2)$$

where K_j^g is the gth word in K_j. Geometric mean was chosen over arithmetic average as it indicates the central tendency of a set of elements. Using (2), the emotion vector σ_{w_i} for word w_i is represented as follows,

$$\sigma_{w_i} = < \text{PMI}\,(w_i, e_1), \text{PMI}\,(w_i, e_2), \ldots, \text{PMI}\,(w_i, e_m) > \ . \qquad (3)$$

2) Adjusting Emotion Vector of a Word using Context Information: Depending on the type of syntactic dependencies identified in Sect. III-C, we fine-tune the emotion vector of the dependent word. For the dependent word in an *adjectival complement* or *adjectival modifier* relationship, we adjust the emotion scores of the dependent word using the scores of its influencing word. Let β_q be a dependent word and α_p be its influencing word. The emotion vector of β_q is adjusted as follows:

$$\sigma'_{\beta_q} = \frac{\sigma_{\beta_q} + \sigma_{\alpha_p}}{2} \ . \qquad (4)$$

[1]For our experiments, we use a window size of 16 words as previous findings report that counting co - occurrences within small windows of text produces better results than larger contexts [23].

[2]Note that using representative words is different from consulting an affect dictionary in the sense that we only need a very small, fixed number of representative words for each emotion concept.

If a dependent word is part of a negation relation, such as "She is not sad", where 'sad' is negatively modified by 'not', then the dependent word's score is set to zero. This way, the word 'sad' becomes *neutral* and no longer contributes to the overall emotion of the sentence. The reason for not reverting the emotion to its counterpart is that 'not sad' does not mean 'happy'. Also, not every emotion has its direct inverse.

G. Calculating Emotion Vector of a Sentence

To sum it, the emotion vector of a sentence can be computed by aggregating the emotion vectors of all the affect words and averaging it as shown in (5),

$$\sigma_s = \frac{\sum_{i=1}^{n} \sigma_{w_i}}{n} \tag{5}$$

where n is the total number of affect words. After obtaining the emotion vector $\sigma_s =< s_1, s_2, \ldots, s_m >$ of a sentence, if the highest score is above a certain threshold t, the sentence is labeled with that emotion. Otherwise, it is classified as neutral. The final emotion label ω_s is computed as in (6),

$$\omega_s = \begin{cases} e_k & \text{if } \max\limits_{i=1,\ldots,m}(s_i) = s_k \text{ and } s_k \geq t \\ \text{neutral} & \text{otherwise} \end{cases}. \tag{6}$$

IV. Evaluation and Results

In this section, we report and discuss the results of the evaluation of our algorithm UnSED (UNsupervised Semantic Emotion Detection) on three standard data sets. The first question we would like to answer is, how much effect does the text corpus have on the semantic relatedness scores, and ultimately on the accuracy of the emotion detection. We compare three different corpora:

1) Wikipedia data[3]
2) Gutenberg corpus[4], a collection of over 36,000 ebooks (it intuitively seems to be more 'emotional' than the objective data on Wikipedia)
3) Wiki-Guten, created by combining the two aforementioned data sets

The next question is whether stemming will improve the accuracy. It may generalize the words too much or since some words share one stem, they are more likely to invoke similar emotions and it may be advantageous to exploit this latent relationship. Thirdly, can the underlying syntactic dependency structure provide some context and therefore, improve the overall accuracy. We compare two versions of UnSED, one being the context-based where the syntactic dependencies are taken into account and the other context-free version which excludes the use of such dependencies. Finally, how does UnSED fare when compared to other recently proposed methods?

[3]http://download.wikimedia.org/enwiki/latest/enwiki-latest-pages-articles.xml.bz2

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

A. Evaluating the Effect of Stemming

We test the effect of stemming on emotion detection on a subset of Alm's data set (to be described in Sect. IV-B) using stemmed and *unstemmed* Wikipedia and Gutenberg corpora. Table II shows the accuracies of emotion detection.

Table II
Stemming accuracy

	Unstemmed	Stemmed
Wikipedia	52.20 %	58.08 %
Gutenberg	54.76 %	55.34 %

It is observed that stemming the text does improve the accuracy of the emotion detection process. This may be because the roots of words closely relating to a particular emotion concept get grouped together and this latent clustering enables more accurate frequency counts. Since stemming is beneficial to the overall detection process, stemmed text is used in the following experiments.

B. Evaluation on the Alm Data Set

Emotions are particularly significant elements in the literary genre of fairy tales and in this experiment, we work with Alm gold standard data set[5] which comprises of 1,207 sentences annotated with five emotions taken from 176 fairy tale stories. To directly compare the results of context-based and context-free variations of UnSED with recent related work, we replicate two different test sets – one set contains five classes of emotions as well as neutral as used by Alm [25] and the other is identical to that used by [17] containing a subset of the original data set (only four emotions).

1) Alm Six Emotions Classification: On Alm's six-emotion data set, we compare our method with a keyword baseline and Alm's unsupervised lextag method [25]. The keyword baseline works as follows. For each NAVA word, if it appears in the affect lexicon (i.e., WordNet-Affect), it is tagged with the emotion category under which it is listed. The sentence label is derived by choosing the most frequent emotion in the sentence. If there is a tie, one of the highest emotions is randomly selected. If none of the NAVA words is found in the lexicon, the sentence is labeled as *neutral*. Alm's lextag method uses a special word list and employs a straightforward heuristic. The results, as shown in Table III, where our algorithm is denoted as UnSED, are reported in terms of accuracy so as to keep it comparable to that reported in [25]. As shown in the table, our approaches perform better than the keyword baseline and Alm's lextag method, and the context-based approach is slightly better than the context-free one.

2) Alm Four Emotions Classification: On Alm's four-emotion data set, in Table IV, we list the F-score values of 6 versions of our unsupervised method (without any affect

[5]Affect data: http://lrc.cornell.edu/swedish/dataset/affectdata/index.html

Table III
RESULTS ALM DATA SET SIX EMOTIONS

Algorithm	Overall Accuracy on Six Emotions
Keyword baseline	45 %
Alm's unsupervised lextag	54-55 %
UnSED Context-free Wikipedia	56.31 %
UnSED Context-based Wikipedia	57.25 %

dictionary) along with the keyword baseline and 4 other unsupervised methods (with affect dictionary) [17]. The 4 unsupervised methods include a vector space model with dimensionality reduction variants (LSA, PLSA and NMF) and a dimensional model (DIM). The F-score on a class c is defined as $\frac{2\times precision\times recall}{precision+recall}$, where $precision$ is the number of sentences correctly labeled as belonging to the class c divided by the total number of sentences labeled as belonging to c, and $recall$ is the number of sentences correctly labeled as belonging to class c divided by the total number of sentences that actually belong to c.

Table IV
F-SCORE RESULTS ALM DATA SET FOUR EMOTIONS

Algorithm	Happy	Sad	Ang-Dis	Fear	Avg.
Keyword baseline	0.593	0.417	0.247	0.265	0.380
LSA	0.727	0.642	0.510	0.640	0.629
PLSA	0.436	0.370	0.313	0	0.279
NMF	0.781	0.760	0.650	0.741	0.733
DIM	0.789	0.240	0.392	0.255	0.419
Context-free Wiki	0.730	0.539	0.548	0.579	0.599
Context-free Guten	0.718	0.549	0.366	0.525	0.540
Context-free W-G	0.733	0.621	0.548	0.571	0.618
Context-based Wiki	0.732	0.533	0.577	0.582	0.606
Context-based Guten	0.722	0.544	0.376	0.533	0.544
Context-based W-G	0.737	0.622	0.549	0.572	0.620

The average F-scores in Table IV show that our methods are better than the keyword baseline, PLSA and DIM methods, and comparable to the LSA method. NMF is much better than others on this data set (but later we will show it is much worse on another data set). Although all the methods compared on this data set are unsuperivsed approaches, ours have an additional advantage of not using any affect lexicons. As we will notice in the next section, not using any affect lexicons enables our approach to work with any emotions as well as any number of categories. The results also indicate that the context-based methods are better than context-free ones.

C. Evaluations on the ISEAR Data Set

The next data set, which contains 7,666 sentences labeled with seven emotions, is the ISEAR text[6] that was developed by asking 3,000 participants from different cultural backgrounds about their emotional experiences.

[6]http://www.affective-sciences.org/system/files/page/2636/ISEAR.zip

1) ISEAR Four Emotions Classification: To be able to compare our results with other unsupervised approaches discussed in [17], we only work with four emotion categories in this task. Table V lists the F-scores of a keyword baseline, 4 unsupervised methods plus 3 versions of our method on the ISEAR four-class data set, where the proposed context-based approach yields the highest F-score values for all the four emotions. On average, our method is significanly better than all the other methods. It is also noticed that on this data set NMF (which was the best on Alm's 4-emotion data) performs significantly worse than all the other methods.

Table V
F-SCORE RESULTS ISEAR DATA SET FOUR EMOTIONS

Algorithm	Joy	Sad	Ang-Dis	Fear	Avg.
Keyword baseline	0.371	0.270	0.346	0.328	0.328
LSA	0.103	0.106	0.631	0.071	0.227
PLSA	0.340	0.282	0.456	0	0.269
NMF	0.010	0.017	0.579	0.056	0.165
DIM	0.515	0.337	0.286	0.351	0.372
Context-based Wiki	0.564	0.408	0.628	0.592	0.548
Context-based Guten	0.574	0.253	0.582	0.536	0.486
Context-based W-G	0.542	0.296	0.668	0.574	0.520

2) ISEAR Seven Emotions Classification: Techniques that employ WordNet-Affect are restricted to Ekman's six emotion classification. However, the ISEAR data set has been annotated with seven emotions, of which the emotions *shame* and *guilt* are not found in WordNet-Affect. To the best of our knowledge, no other emotion detection results have been reported on the ISEAR seven categories. Our F-score results of full seven class categorization are presented in Table VI.

For four out of seven categories and overall too, the UnSED context-based approach based on the Wikipedia corpus results in the best performance. The *fear* and *joy* categories have good performance, but the F-score values on *guilt* are among the lowest. A look at the data set reveals sentences such as "While having an argument with my daughter, I got angry and over-excited" and "Falling in love with a close friend" labeled as *guilt*. These sentences border on the fuzzy boundary of emotions and we believe that deeper syntactic and semantic analyses are required to decipher the underlying feelings.

D. Evaluation on Aman Blog Data Set

This rich emotional blog data set was kindly provided by the authors of [14]. We test on the gold standard set which includes 1,890 sentences annotated with six emotions as well as neutral. Since there is no unsupervised, affect lexicon-independent work reporting results on this data set as far as we know, we present our unsupervised results in Table VII alongside those from supervised approaches taken from [14] and [26] and a keyword baseline.

Supervised SVM was applied in [14] on two algorithms – one using unigrams (Aman's supervised 1 in Table VII) and the other better result obtained by combining unigrams,

Table VI
F-SCORE RESULTS ISEAR DATA SET SEVEN EMOTIONS

UnSED Context-based	Joy	Sadness	Anger	Fear	Disgust	Shame	Guilt	Avg.
Wikipedia	0.514	0.396	0.413	0.517	0.430	0.400	0.338	0.430
Gutenberg	0.500	0.248	0.415	0.439	0.432	0.397	0.374	0.401
Wiki-Guten	0.500	0.290	0.414	0.483	0.470	0.400	0.329	0.412

Table VII
F-SCORE RESULTS AMAN DATA SET

Algorithm	Happy	Sad	Anger	Fear	Surprise	Disgust	Neutral	Avg.
Keyword baseline	0.519	0.331	0.348	0.244	0.218	0.145	0.510	0.330
Aman's supervised 1	0.740	0.405	0.457	0.629	0.479	0.571	0.431	0.530
Aman's supervised 2	0.751	0.493	0.522	0.645	0.522	0.566	0.605	0.586
Ghazi's supervised	0.690	0.460	0.430	0.450	0.380	0.310	0.84	0.508
Context-free Wiki	0.748	0.487	0.465	0.497	0.396	0.408	0.691	0.527
Context-free Guten	0.742	0.502	0.428	0.542	0.42	0.431	0.717	0.540
Context-free WikiGuten	0.698	0.403	0.433	0.645	0.278	0.434	0.453	0.478
Context-based Wiki	0.751	0.498	0.454	0.503	0.425	0.420	0.697	0.535
Context-based Guten	0.745	0.513	0.433	0.531	0.425	0.450	0.722	0.546
Context-based WikiGuten	0.707	0.427	0.440	0.649	0.286	0.448	0.652	0.516

Roget's Thesaurus and WordNet-Affect (Aman's supervised 2 in Table VII). SVM was also applied in [26] that experimented with a hierarchical approach, Roget's Thesaurus and WordNet-Affect. It should be noted that unlike other approaches, we do not employ any affect lexicons.

E. Summary of the Results

Our proposed approach retrieves semantic relatedness scores from different text corpora. Now, let us see how these text corpora fared when compared to each other in Fig. 2.

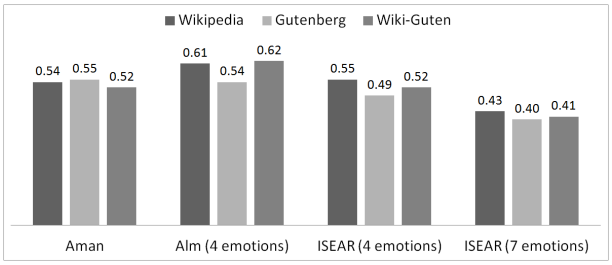

Figure 2. Average F-score values across three text corpora

The Wikipedia corpus resulted in the higest average for two out of four tasks and stood second in the other two tasks. This could be attributed to the fact that Wikipedia contains more structured data. On the other hand, the Gutenberg corpus falls in the last place in three out of four tasks while being the best in one task, where it does only slightly better than Wikipedia. This goes against our initial postulation that Gutenberg, which contains supposedly more 'emotional' text, would be a better choice. From the results, it can be concluded that semantic relatedness scores derived from Wikipedia perform relatively better. Moreover, the largest corpus (Wiki-Guten) did not result in the best performance.

Different methods perform differently on different data sets. One of the first observations that can be made from Table VIII is that for all the approaches, the F-score values on the Alm data set are higher than that of ISEAR. This could be because fairy tales tend to have more emotional words. Secondly, although NMF shines on the Alm data set, it has the weakest results on ISEAR. Such discrepancy implies that it is effective for certain types of sentences only. The average F-score values in Table VIII show that the proposed context-sensitive method seems to be a more suitable choice in a general emotion classification task.

Table VIII
COMPARING AVERAGE F-SCORES OF UNSUPERVISED APPROACHES

Algorithm	Alm	ISEAR	Average
LSA	0.629	0.227	0.428
PLSA	0.279	0.269	0.274
NMF	0.733	0.165	0.449
DIM	0.419	0.372	0.396
UnSED Context-based Wikipedia	0.606	0.548	0.577
UnSED Context-based Gutenberg	0.544	0.486	0.503
UnSED Context-based Wiki-Guten	0.620	0.520	0.570

V. CONCLUSION

In this paper we proposed a context-sensitive unsupervised approach of detecting emotions from text. Our methodology requires neither an annotated data set, nor any detailed affect lexicon. The results of evaluations show that our technique yields more accurate results than other recent unsupervised approaches and comparable results to those of some supervised methods. One of the weaknesses of our approach is that the semantic relatedness scores depend on the text corpus from which they are derived. From the empirical results, we observed that the Wikipedia corpus is better than the other two corpora and that the context-based approach consistently outperformed the context-free approach which supports the claim that it is useful to look

at words within their context. In the future, we would like to derive the semantic relatedness scores from multiple measures and test the use of other syntactic dependencies.

ACKNOWLEDGMENTS

We would like to thank Dr. Sara Diamond for inspiring our research. This project is funded in part by the Centre for Information Visualization and Data Driven Design (CIV/DDD) established by the Ontario Research Fund.

REFERENCES

[1] V. Hatzivassiloglou and K. R. McKeown, "Predicting the semantic orientation of adjectives," in *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, 1997.

[2] C. Strapparava and A. Valitutti, "Wordnet-affect: an affective extension of wordnet," in *Proceedings of the 4th International Conference on Language Resources and Evaluation*, 2004.

[3] T. Wilson, J. Wiebe, and P. Hoffmann, "Recognizing contextual polarity in phrase-level sentiment analysis," in *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, 2005, pp. 347–354.

[4] A. Esuli and F. Sebastiani, "Sentiwordnet: A publicly available lexical resource for opinion mining," in *Proceedings of the 5th Conference on Language Resources and Evaluation*, 2006, pp. 417–422.

[5] A. Neviarouskaya, H. Prendinger, and M. Ishizuka, "Sentiful: Generating a reliable lexicon for sentiment analysis," in *Affective Computing and Intelligent Interaction and Workshops*, 2009.

[6] J. Olveres, M. Billinghurst, J. Savage, and A. Holden, "Intelligent, expressive avatars," in *Proceedings of the First Workshop on Embodied Conversational Characters*, 1998.

[7] S. Al Masum, H. Prendinger, and M. Ishizuka, "Emotion sensitive news agent: An approach towards user centric emotion sensing from the news," in *Web Intelligence, IEEE/WIC/ACM International Conference on*, nov. 2007, pp. 614 –620.

[8] F.-R. Chaumartin, "Upar7: a knowledge-based system for headline sentiment tagging," in *Proceedings of the 4th International Workshop on Semantic Evaluations*, 2007, pp. 422–425.

[9] A. Meena and T. V. Prabhakar, "Sentence level sentiment analysis in the presence of conjuncts using linguistic analysis," in *Proceedings of the 29th European Conference on IR Research*, 2007, pp. 573–580.

[10] A. Neviarouskaya, H. Prendinger, and M. Ishizuka, "Recognition of affect, judgment, and appreciation in text," in *Proceedings of the 23rd International Conference on Computational Linguistics*, ser. COLING '10, 2010, pp. 806–814.

[11] H. Liu, H. Lieberman, and T. Selker, "A model of textual affect sensing using real-world knowledge," in *Proceedings of the 8th International Conference on Intelligent User Interfaces*, 2003, pp. 125–132.

[12] Y. Neuman, G. Kedma, Y. Cohen, and O. Nave, "Using web-intelligence for excavating the emerging meaning of target-concepts," in *Proceedings of the International Conference on Web Intelligence and Intelligent Agent Technology*, 2010, pp. 22–25.

[13] C. Alm, "Affect in text and speech," 2008. [Online]. Available: http://cogcomp.cs.illinois.edu/papers/Alm thesis(1).pdf

[14] S. Aman and S. Szpakowicz, "Using roget's thesaurus for fine-grained emotion recognition," in *Proceedings of the Third International Joint Conference on Natural Language Processing*, 2008, pp. 296–302.

[15] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: sentiment classification using machine learning techniques," in *Proceedings of the Conference on Empirical methods in natural language processing*, 2002.

[16] J. Martineau and T. Finin, "Delta tfidf: An improved feature space for sentiment analysis," in *Proceedings of the AAAI Internatonal Conference on Weblogs and Social Media*, 2009.

[17] S. M. Kim, A. Valitutti, and R. A. Calvo, "Evaluation of unsupervised emotion models to textual affect recognition," in *Proceedings of the NAACL HLT Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, 2010, pp. 62–70.

[18] C. Strapparava and R. Mihalcea, "Learning to identify emotions in text," in *Proceedings of the ACM symposium on Applied computing*, 2008, pp. 1556–1560.

[19] Z. Kozareva, B. Navarro, S. Vázquez, and A. Montoyo, "Uazbsa: a headline emotion classification through web information," in *Proceedings of the 4th International Workshop on Semantic Evaluations*, 2007, pp. 334–337.

[20] P. Gamallo, C. Gasperin, A. Agustini, and J. G. P. Lopes, "Syntactic-based methods for measuring word similarity," in *Proceedings of the 4th International Conference on Text, Speech and Dialogue*, 2001, pp. 116–125.

[21] P. Ekman, "Facial expression and emotion," *American Psychologist*, vol. 48, no. 4, pp. 384–392, 1993.

[22] K. W. Church and P. Hanks, "Word association norms, mutual information, and lexicography," *Computational Linguistics*, vol. 16, no. 1, pp. 22–29, 1990.

[23] J. A. Bullinaria and J. P. Levy, "Extracting semantic representations from word co-occurrence statistics," *Behavior Research Methods*, no. 3, pp. 510–526, 2007.

[24] G. Recchia and M. N. Jones, "More data trumps smarter algorithms: Comparing pointwise mutual information with latent semantic analysis," *Behavior Research Methods*, vol. 41, no. 3, p. 647, 2009.

[25] C. Quan and F. Ren, "An exploration of features for recognizing word emotion," in *Proceedings of the 23rd International Conference on Computational Linguistics*, 2010, pp. 922–930.

[26] D. Ghazi, D. Inkpen, and S. Szpakowicz, "Hierarchical versus flat classification of emotions in text," in *Proceedings of the Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, 2010, pp. 140–146.